CG4 COURSEWORK PROJECT

ARE YOU NOT EDUTAINED?

BOURNE GRAMMAR SCHOOL

2012-2013

BEN GOLDSWORTHY

9056

CENTRE NUMBER: 26202

Contents

Analysis

Background

Bourne Grammar School is a selective co-ed secondary school in Bourne, established in 1330. The school is a specialist performing arts school and is renowned for its innovative approach to using technology to enhance teaching. The school serves students aged 11-18.

Lately, the school has been experimenting with using games to engage the attention of students and hopefully help them retain what they have learnt more easily; games like noughts & crosses, with correct answers to subject-specific questions giving the student the chance to choose a square. Currently, the games are done on the classroom boards. The school is looking to digitise the experience, in keeping with their reputation as a technologically-minded school.

The current headmaster of the school has posted an advert for a computer programmer to create a suite of edutainment games that can be installed throughout the school's computer systems and will ideally serve to engage the students in their studies more than the traditional methods. Another benefit is that the program can be used interdepartmentally, which would give it greater scope to be used across the school. The selection of games on offer is up to the programmer, but the advert requests that they be augmented with questions relating to a variety of subject, which will serve as the 'edu-' part, correctly answering which will allow the player to play the game for a brief period (the '-tainment' part).

In a further effort to engage the students in the system by fostering a competitive spirit, the advert requests that the program be capable recording highscores, allowing students to challenge each other. A suggestion is made for some sort of persistent account system; allowing the students to track their progress and giving them sort of record of their achievements. It will also enhance the game aspect of the suite, which in theory would assuage the student's revulsion towards anything educational. The games need to be suitable for the school environment, targeted towards both genders and all the ages present in the school.

I have chosen this undertaking because I feel computerisation can greatly enhance the learning experience of modern-day youths who live in an increasingly computerised world, and because the networking aspect especially would be something useful to learn how to do for the future.

Investigation and Analysis of the Current System

To start my analysis of the current system, I sat in the back of a room for a typical lesson and observed the system in use during the closing ten minutes of the lesson. The teacher drew a noughts & crosses grid on the board and divided the class into two teams. After flipping a coin to decide which team will start, the teacher asked a student on each team in turn questions relevant to what they have been learning. If the student answered correctly, they could choose to place a nought or a cross and it was the other team's go. If the student answered it incorrectly, the question passed over to the opposing team, allowing them to possibly get two goes in a row.

After winning a game, the team earned a point which the teacher recorded. The winning team is displayed in the classroom, which fosters some competition and can lead to rewards for the winner upon the end of the year.

This system has limitation, as I witnessed a few students not participating and losing focus. Following the lesson, I interviewed the teacher:

ME How do you feel the use of games at the end of lessons enhances the learning experience of your students?

MR BROWN I feel that it provides a good gauge of how well the students have taken onboard everything they've learned in the lesson, as well as being a good way of unwinding at the end.

ME Do you feel a computer-based system would be the better option?

MR BROWN Yes. For one, I could leave the students to play the games whilst I could catch up on other work. Another benefit would be that multiple games could easily be played in a short amount of time, as the computer program would be far faster than me drawing the game on the board. I also feel that the students would connect better with a computer-based system as they have been brought up alongside computers.

ME Thank you.

Having analysed the current system, these are the inputs, processes and outputs involved:

Process 1: Noughts & Crosses

Inputs: Teacher asks team question, chosen student answers.

Processes: If the answer is right, the student may place their team's nought/cross. If wrong, the question is passed onto the opposing team. If they are also wrong, the answer to the question is given and the

game moves on. If three of a team's symbols are connected either horizontally, vertically or diagonally, they are the winners.

Outputs: Student knowledge of subject, a board with a completed noughts & crosses board and possibly a winning team.

This process involves students answering a question asked by the teacher, which allows them to place a nought or a cross (depending on their team) depending on whether they answered correctly or not.

Process 2: Connect Four

Inputs: Teacher asks team question, chosen student answers.

Processes: If the answer is right, the student may choose a column to drop a counter in. If wrong, the question is passed onto the opposing team. If they are also wrong, the answer to the question is given and the game moves on. If four of a team's counters are connected either horizontally, vertically or diagonally, they are the winners.

Outputs: Student knowledge of subject, a board with a completed connect four grid and possibly a winning team.

This process involves students answering a question asked by the teacher, which allows them to drop a counter into a column depending on whether they answered correctly or not.

Process 3: Hangman

Inputs: Teacher displays the number of letters in a word of their choosing. Teacher asks student question, chosen student answers

Processes: If the answer is right, the student may choose a letter. If the letter is present in the word, all instances of it are added to the blank space. If the letter is not present, a segment of the hangman is drawn. Completion of the hangman means victory for the teacher. If the student answers incorrectly, the correct answer is given and another student is chosen.

Outputs: Student knowledge of subject, a board with either a completed hangman drawing or a completed word.

This process involves students answering a question asked by the teacher, which allows them to guess at a letter present in a word chosen by the teacher depending on whether they answered correctly or not.

Process 4: Squares

Inputs: Teacher asks team question, chosen student answers.

Processes: If the answer is right, the student may choose two adjacent dots to draw a horizontal or vertical line between, the intention being to form a square without allowing their opponents ¾ square to finish off. If the answer in incorrect, the question is passed onto the opposing time.

Outputs: Student knowledge of subject, a board with a completed squares grid and possibly a winning team.

This process involves students answering a question asked by the teacher, which allows them to place a line depending on whether they answered correctly or not.

Process 5: Scorekeeping

Inputs: Teams win games.

Processes: When a team wins a game, they win a point which is recorded.

Outputs: At the end of the year, the team with the most points is crowned the winner.

This process involves teams earning points by winning games. This leads to persistent competition.

Problem Definition

The current system has limitations. As they are done on classroom boards, the games are limited in scope and quite simple, whereas there is far more room for added complexity in a computer-based system. Also, with the games being performed on the board, students that are hard of seeing might not be able to see them as well as if they had a personal screen in front of them. Another limitation is that some teachers are less enthusiastic about the system, which would be remedied by a computer system saving them the effort. Finally, a computer system allows for more in-depth stat-tracking, far beyond just recording the winning teams.

I intend to create a computer suite of edutainment games for the Bourne Grammar School, which will allow the teachers to get on with other tasks whilst the students enhance their learning with subject- and topic-tailored games. This will increase the efficiency of lessons and give the school a more forward-thinking, up-to-date image.

There are a number of requirements for this system. Regarding accounts, it must track the games played and win/loss rates of each student. There must be a way of creating new accounts and deleting old ones as students move through the school. An extra feature, time permitting, would be the tracking of how well students do in each subject comparatively, which would allow teachers to quickly identify students slipping behind and provide extra assistance.

As for the games on offer, the advert specified at least noughts & crosses, connect four and hangman, but if there is time other games could be added such as squares, or rock, paper, scissors. These games must be combined with questions pertaining to the subject and topic, and there must be a way of selecting these.

As it is targeted towards teenagers, the UI should be simple and easy-to-use, but not overly simplified lest it appear patronising. The questions should be scalable from Year 7-level all the way up to Upper Sixth-level. Some placeholder questions will be present to test the system, but there must be a way of implementing new questions. A simple database that new questions can be entered into will do, but if there is time a more fleshed-out system should be implemented, involving accounts for teachers which can access a question-creation form.

Objectives

As a bare minimum, my system will achieve the following goals:

- Provide an intuitive UI that is simple and attractive
- Provide versions of hangman, connect four and noughts & crosses with subject- and topic-specific questions selectable on the main menu
- Provide a rudimentary method of inputting new questions
- Provide an account system, whereby each student has a unique account that will track wins and losses
- Provide a rudimentary method of creating and deleting accounts

After these bare necessities are nailed down I can, time permitting, add additional refinements such as:

- Providing more games, such as squares or rock, paper, scissors
- Providing a way of saving games and restoring them at a later date
- Providing more detailed student accounts, with information such as which subject they are doing best at
- Provide a more complex and visually appealing system for monitoring student progress, inputting new questions and creating/deleting accounts
- Some form of achievement system
- Support for teams (similar to how clans work in online gaming)

To achieve all this, I will start by performing an analysis on the current system and plan my solution, followed by designing the UI and data flow. After this, I will work on the programming of my system, followed by rigorous testing. Assuming it passes these, documentation will be created and the solution evaluated to determine the success of the project.

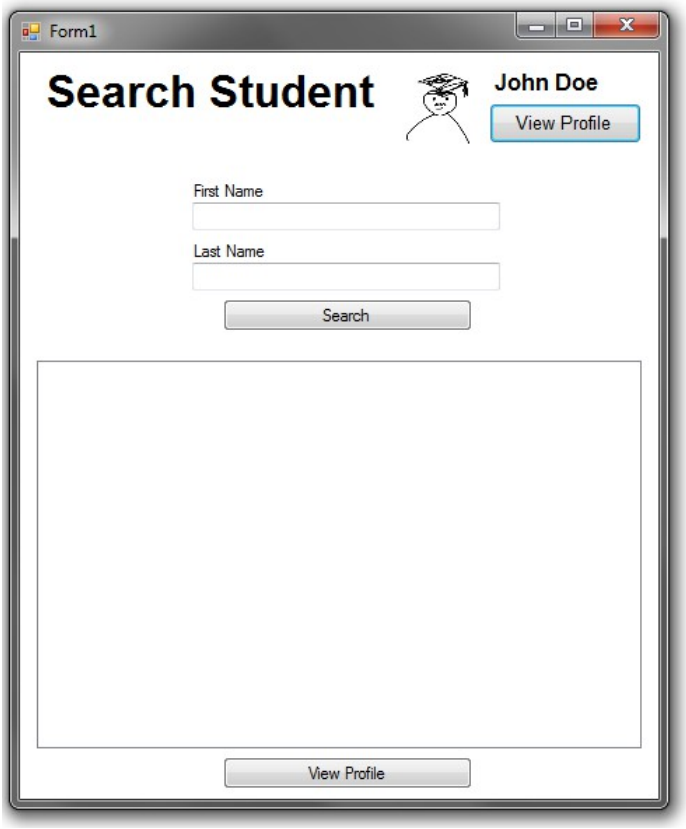
This system is not without its limitations. Without teacher supervision and with the immediate access to computers, students could be distracted and start browsing the internet or emailing friends. Another limitation is that there is less room for a teacher to help out a student who is struggling with a particular question or to see which areas specifically the student needs help in. Another limitation is that the graphics will be very basic, and animation limited if present at all due to not using DirectX or any similar graphics technology. Also, due to the nature of the system, only closed questions are doable, which doesn't best reflect the open question nature of exams. There is of course always the possibility of not having enough time to implement a feature, in which case it would be preferable to excise the feature, rather than include a buggy attempt at it. There is also the possibility that network play will be unimplementable due to the complex nature of the school network, which would make testing impossible.

A final limitation, but perhaps the most major, is that if the computers of the school are unavailable for whatever reason, be it a hardware failure, network out of order or not being able to book the room, then the program will be unusable.

Design

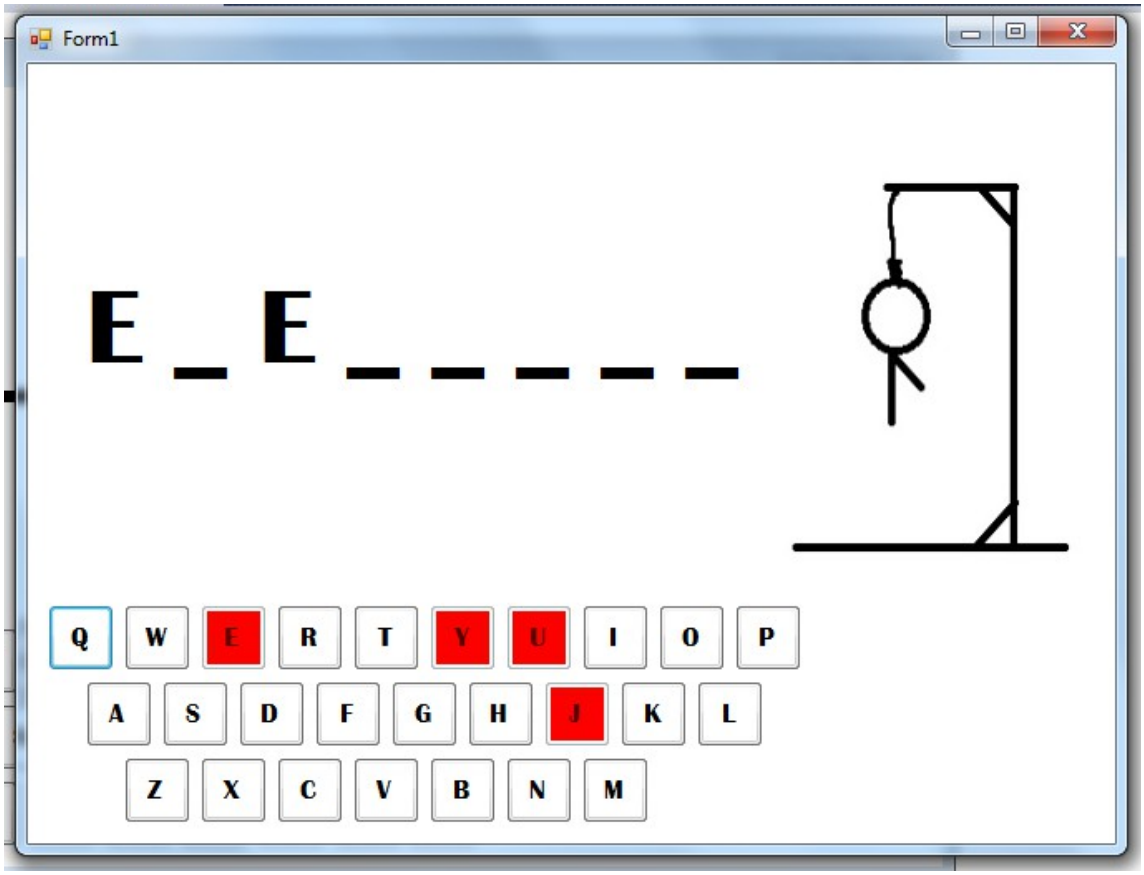
Design Output Content and Format

View Student form



This form displays the usual items along the top and textboxes for entering the desired student's first and last names. Clicking search will expand the form to show the listbox and view profile button, the former of which will be populated with database details of students matching the search criteria and the latter of which will open up the student account form.

Hangman form

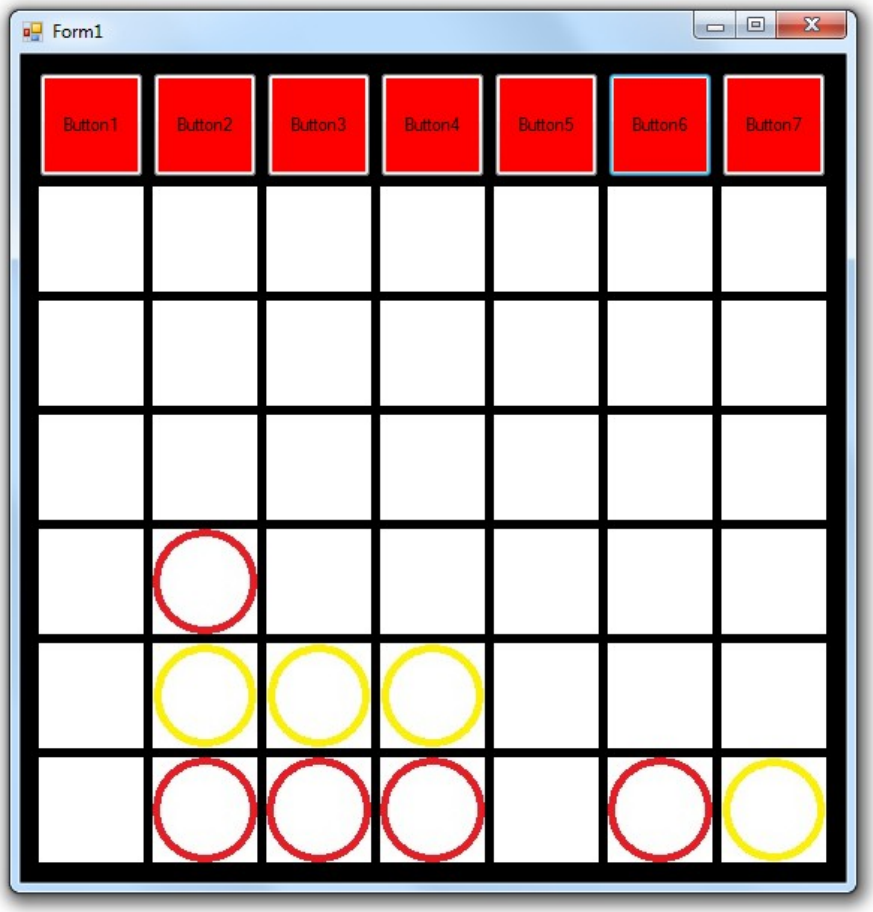


The form will be constructed with the hangman in the top-left, who will be built up piece-by-piece as the player guesses more and more wrong answers. If the hangman is completed, the player will lose.

To the left of the hangman is the word to be guessed. Unknown letters will be represented by underscores with correct guesses replacing these with the appropriate letters. This shall use the Art Deco font, Britannic Bold, in a large 48pt to draw attention immediately and to make it easy to read.

At the bottom of the form is the pool of letters to choose from. I have arranged it in a QWERTY layout so that I can link the keys with the keyboard and have the player press the key for their chosen letter. If I wanted to showboat a little, I could include an option for different keyboard layouts, such as AZERTY or Dvorak. Once a letter has been used, the button will disable and turn red to offer a quick way of seeing which letters have been used.

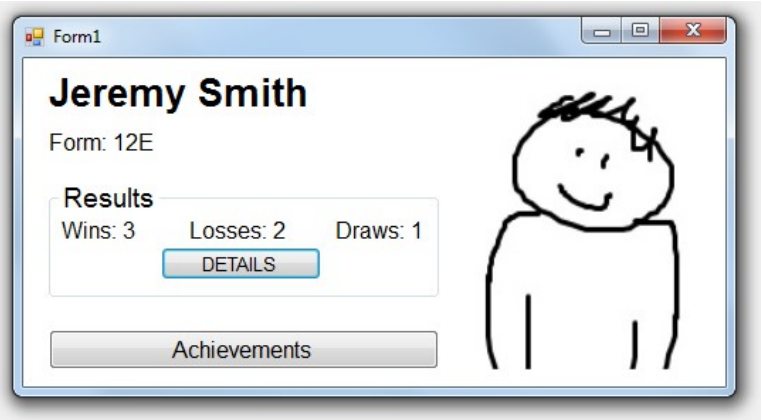
Connect Four form



This form will be constructed with the 7 buttons for dropping counters at the top, above their respective columns in a 6x7 grid. Clicking one will drop a counter (red or yellow depending on the current player) down the column until it reaches either the bottom or a previously-placed counter. If I have time I could add rudimentary animation to this process, showing the counter falling down the row.

Once four counters of the same colour are connected horizontally, vertically or diagonally, or once every square on the board is filled with no winner, the game ends.

Student Account form



This form will be constructed with a picture of the student taking up the right-hand side, taken from the school database. At the top-left will be their name, followed by their form underneath. Then there will be a brief breakdown of their total wins, losses and draws, and a button to go to a more detailed breakdown. These elements will be surrounded by a groupbox to keep everything neat and tidy. Underneath this is a button that will take the user to a screen detailing which achievements the student has unlocked.

Student Details form

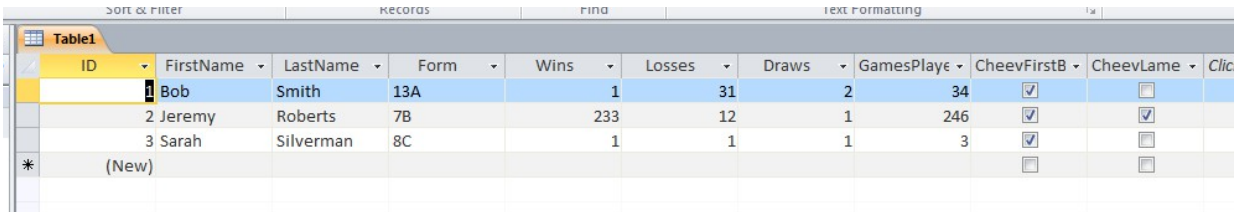
This form will be constructed with the title at the top in Arial 20pt showing the name of the student in question. Underneath this shall be a listbox which shall contain details of who they played, the result, what game they were playing and when it happened.

Student Achievements form

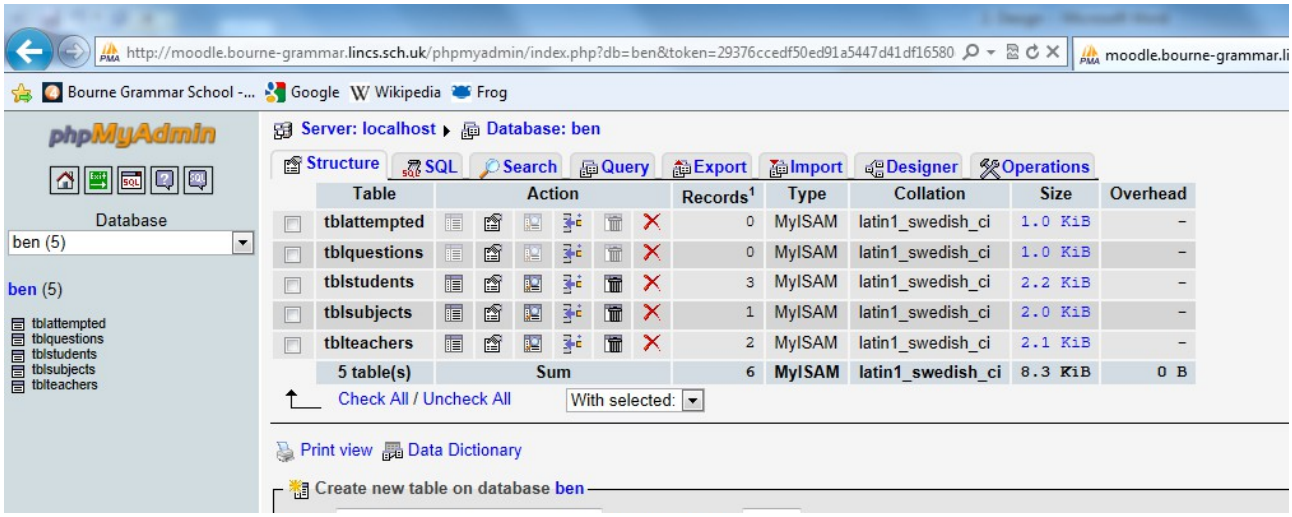
This form will be constructed with a title at the top in Arial 20pt showing the name of the student in question. The rest of the form will be taken up by achievements. Unlocked achievements will display a small picture, their name, the conditions for unlocking them and the date unlocked. The names and pictures will consist mainly of pop-culture references (this example is a references to the Rambo films), terrible puns and similar elements of light-hearted whimsy which will hopefully help the program avoid the stigma of edutainment being universally awful.

Locked achievements will display a padlock and the word “locked” in all-caps, the unattractiveness of which should hypothetically spur the student on to unlock them all. By not displaying the win conditions, it should also take the student longer to unlock them all, lead to curiosity and a desire to uncover the secrets, and foster teamwork in students (e.g. when one student unlocks an achievement, telling other students how to).

Database

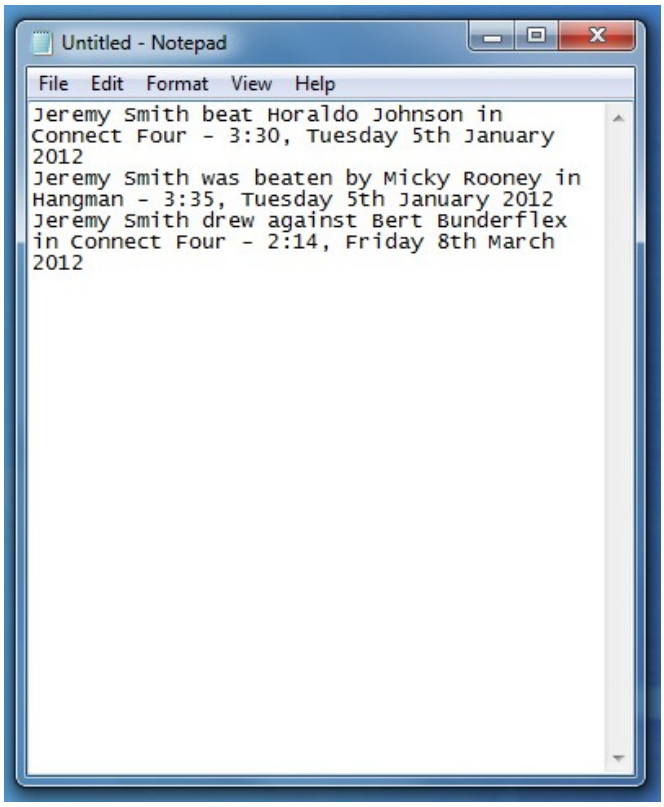


This database will contain four tables; one with details to do with the various students, such as their names, forms, results and which achievements they’ve unlocked; one for the details of teachers, such as name and number of questions they’ve created; one for subjects, with the subject and years that can do it; and one for questions, with details such as the question and answer, the teacher who created it and how many attempts there have been at it.



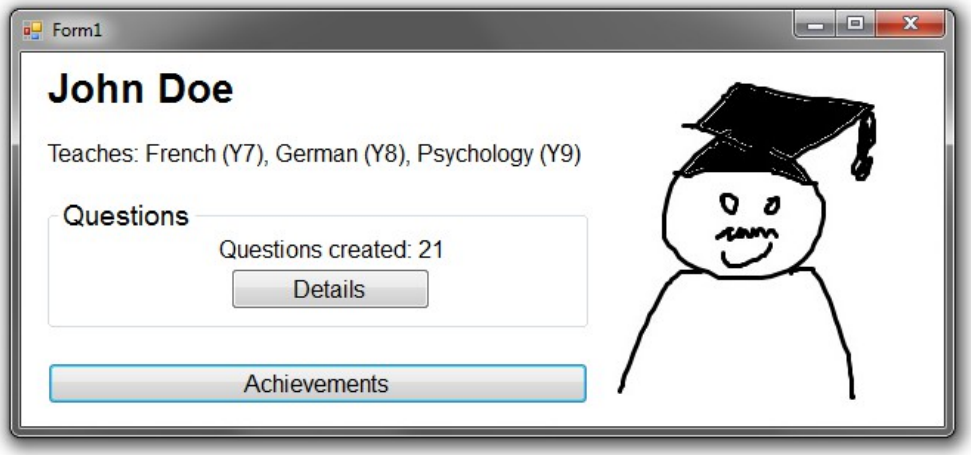
Whilst I am unfamiliar with it at the moment, I could also use mySQL and phpMyAdmin as my database facility. On the one hand, I would need to learn how to use an entirely new program, but on the other, this would allow me to access my database from any computer with an internet connection.

Wins and Losses text file



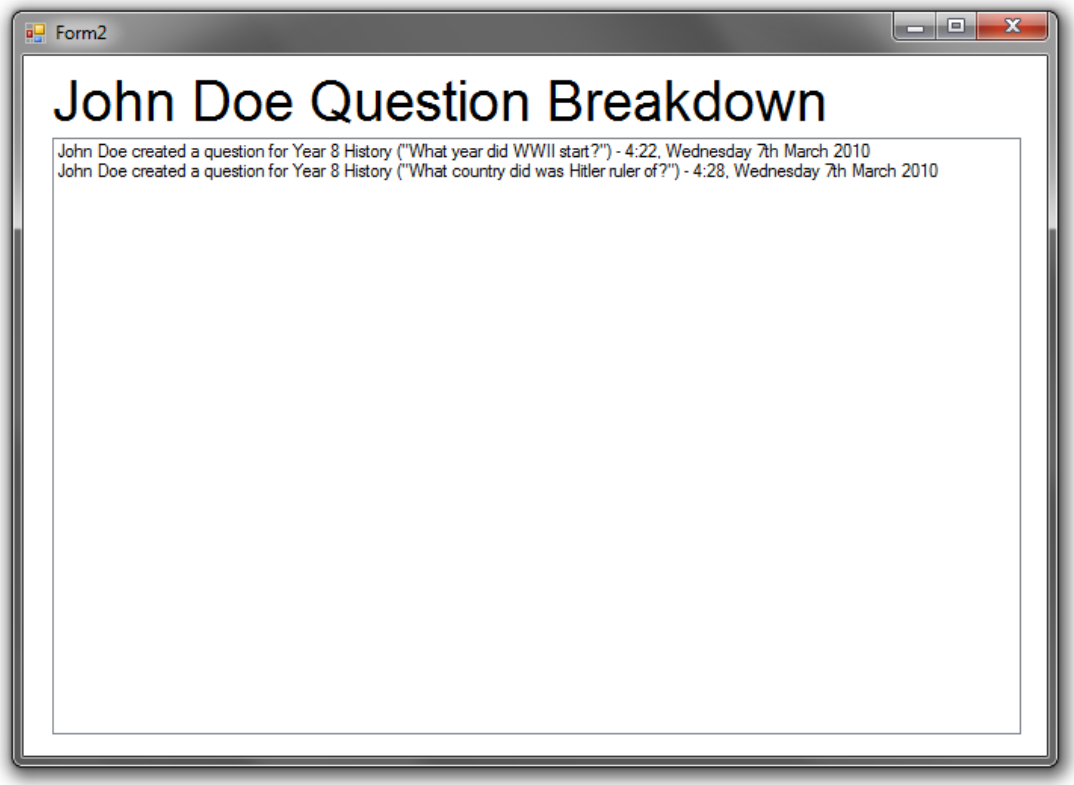
This .txt file will contain the lines referring to previously played games to populate the listbox in the Win Breakdown form.

Teacher Accounts form



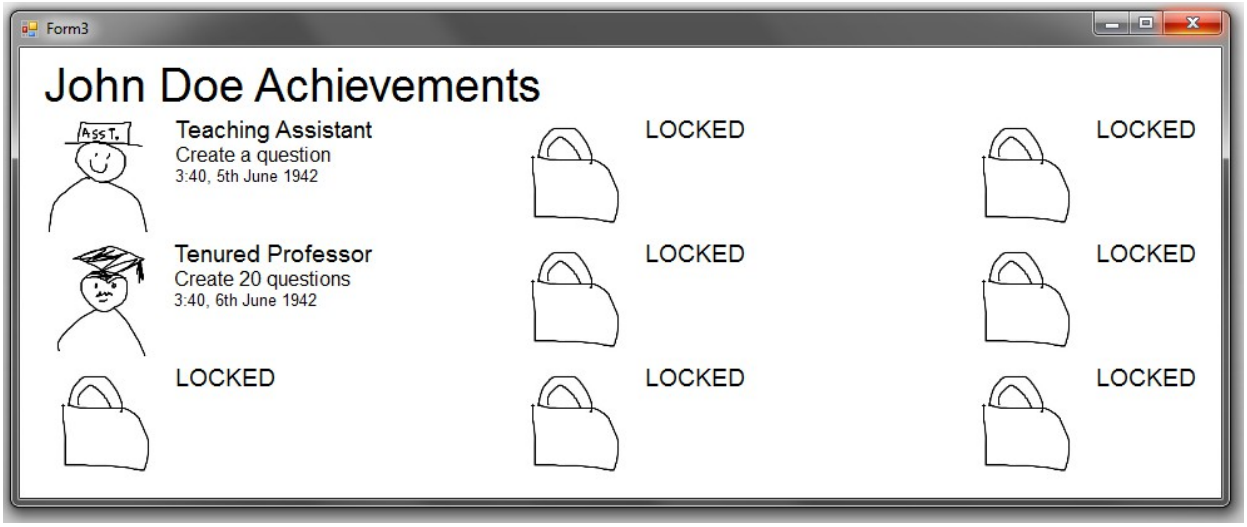
This form will be constructed similarly to the student account form, with a picture of the teacher in the right and the view achievements button along the bottom left. It will display which subject the teacher teaches, as well as what year he teacher them to, as well as the number of questions the teacher has created and a button to see a more detailed dissection of their contributions.

Teacher Details form



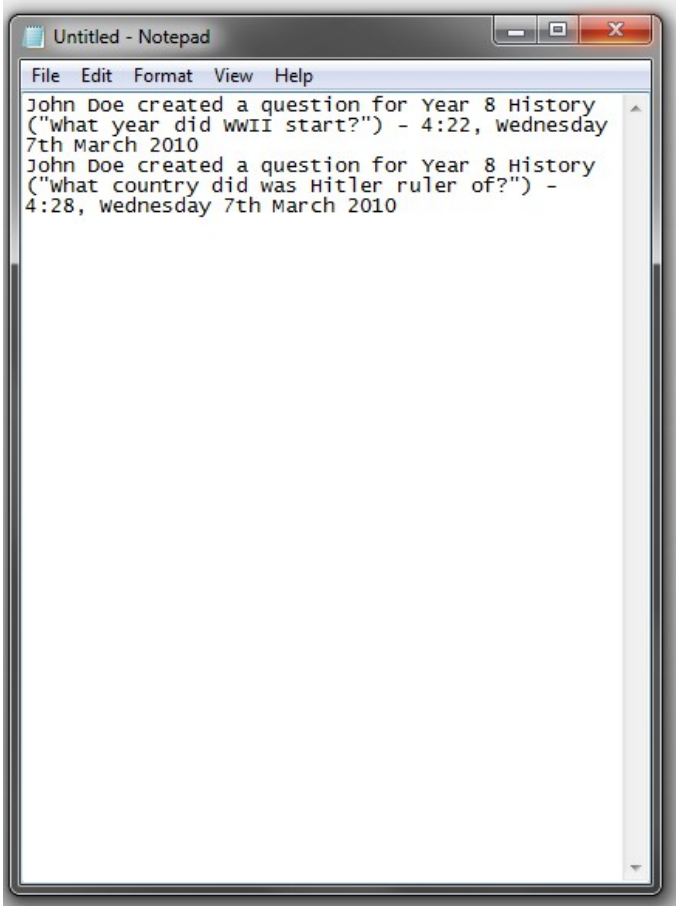
This form is constructed identically to the student details form, except the listbox here displays details of questions created and not games won/lost/drawn.

Teacher Achievements form



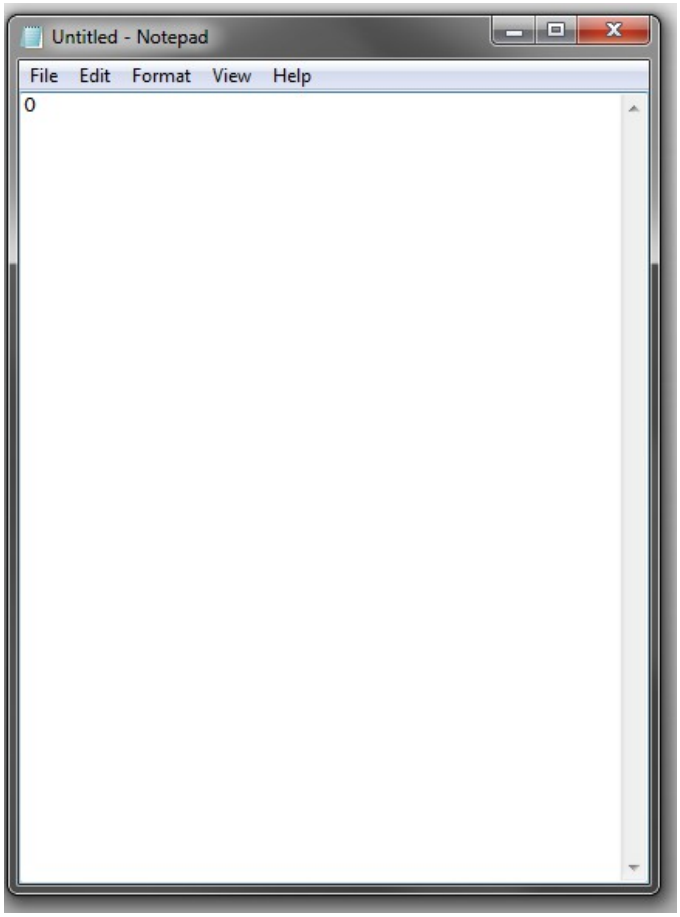
This form is also almost identically to the student equivalent, except obviously with different achievements. Perhaps the unlocking of achievements could be tied to payrises within the school, although such things would be the domain of the school itself to deal with.

Question Creation text file



This .txt file will contain the lines referring to created questions to populate the listbox in the Question Creation Breakdown form.

First Time Setup text file

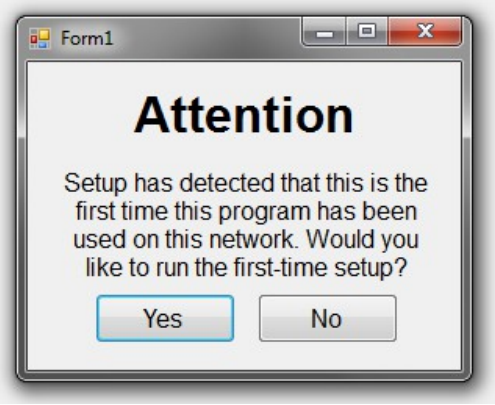


This .txt file simply contains a 1 or a 0 to determine whether this is the first time running the program.

Input Content, Capture and Format

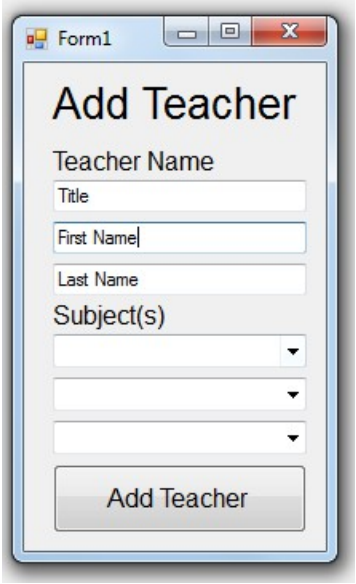
Upon the first loading of the program, the following forms will display:

First Time Setup form



Clicking the yes button leads to the next few forms, clicking no goes to the login screen.

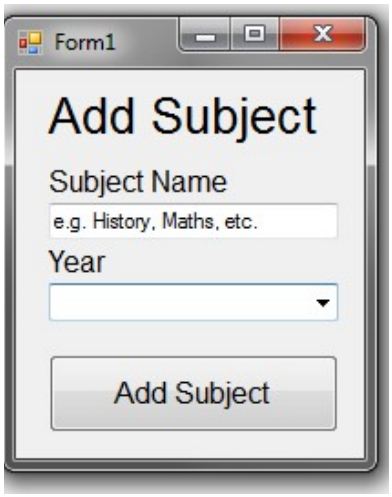
Add Teacher form



As part of the program's first-time setup, teachers, subject and topics can be added. The teacher entry form takes in the teacher in question's title and name via textboxes, and the subject(s) they teach via comboboxes to avoid typos and other mistakes. Not shown is a section for uploading an image of the teacher.

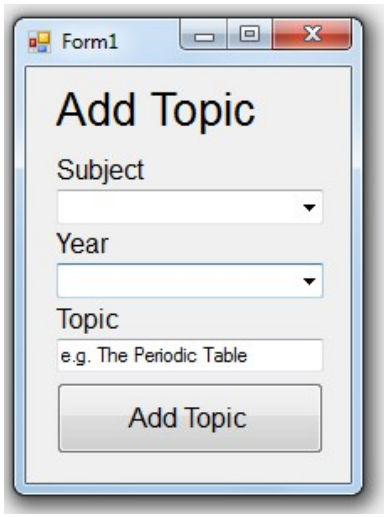
This form, as with all other data entry forms in this section, is intended for use by staff and not students, so the grey control colour is suitable, whereas using it for the student-aimed forms would make the program appear drab and dull.

Add Subject form



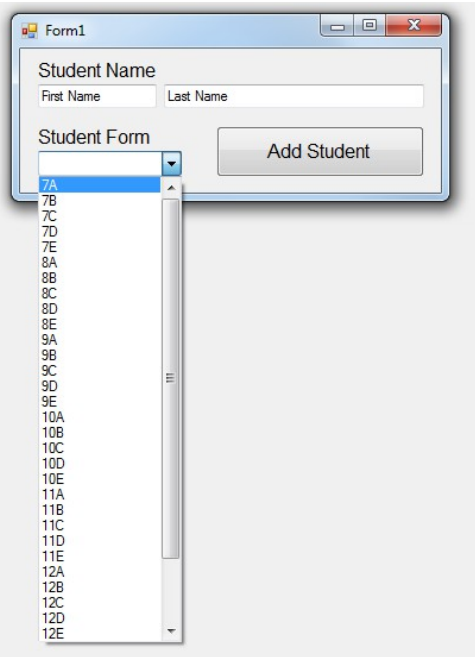
This form takes in subjects via a text box, which could lead to problems if the subjects are misspelt but is the only real option for it, and could be alleviated by some sort of suggested text feature. The relevant year(s) are inputted via combobox, which avoids these problems as there are a limited number of years to choose from.

Add Topic form



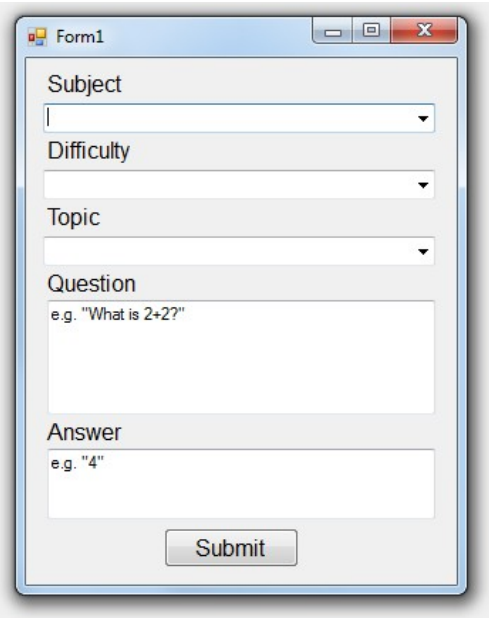
This form uses comboboxes to select the subject, entered in the previous form, and the relevant year, in order to hopefully avoid mistakes (although if the subject was entered incorrectly in the previous form, that error will be carried forward. The topic is inputted via a textbox, which again opens up the possibility for errors but is the only option.

Add Student form



This form will be constructed with two text boxes for entering the student's name (clicking on the text boxes will clear the hint text), and a combobox for the student's form, to avoid any invalid data input. There is also a large button to add the student to the database, which will default their achievements to locked and their gameplay stats to 0. Not shown is a section for uploading an image of the teacher.

Add Question form



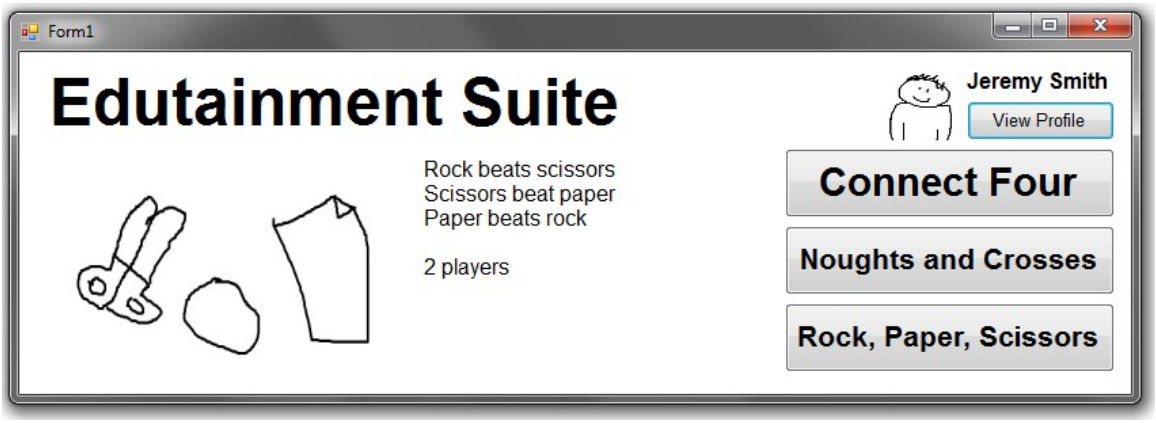
This form will be constructed with three comboboxes and two textboxes. The first combobox will be for the subject the question relates to (e.g. Maths, History, French, etc.). The second will be to specify the difficulty level of the question, from Year 7 to Year 13. The third will display a list of topics tailored to the subject and year chosen previously (e.g. "The British Experience of War – The Crimean War, The Boer War

and World War I" for Year 12 History). In the first textbox, the question will be entered. The answer goes in the second. After clicking the submit button, a message box will pop up, asking the user if they are certain they have typed the question and answer correctly.

Login form

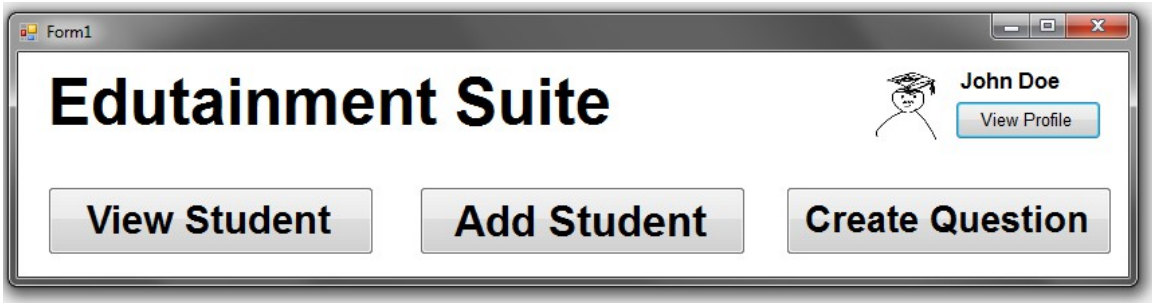
This form is a generic login form, with textboxes for the entry of usernames and passwords and buttons to submit those details for approval. In the finished product, perhaps a graphic more tailored to the suite could be added to the left.

Student Home form



This form displays the currently logged-in user's name and picture in the top-right, along with a button to view their profile. The name of the suite appears in large in the top-left (to be replaced with a somewhat catchier name). To the right are buttons for all of the games available, and moving the mouse over them will display a picture of the game and a brief synopsis, in case a student is somehow unfamiliar with them.

Teacher Home form



This form is similar to the student home form, displaying the program title in the top-left and the logged-in teacher's details in the top-right. Instead of games, however, the teacher has buttons giving them the option to view a student's profile, add a new student or create a new question.

Files and/or Data Structure, Methods of Access

The data for my program shall be stored in a number of locations. A MySQL database will be used to store student details, the advantage being that the database can then be accessed by any computer with an internet connection, teacher details and question details, and text files will be used to store logs of question creating and game wins/losses/draws.

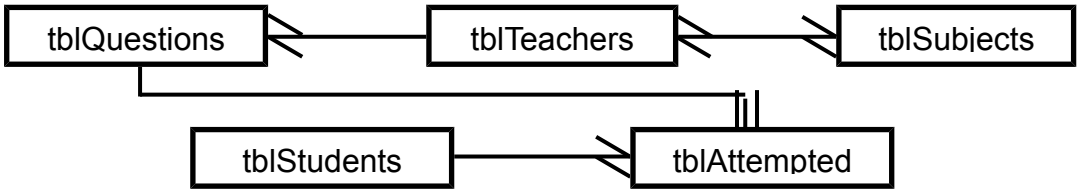
To start with, the database:

Field Name	Data Type	Length	Validation
tblStudents			
StudentID	Primary Key	∞	Only integers accepted
Fname	String	10 characters max.	Only letters accepted
Lname	String	15 characters max.	Only letters accepted
Username	String	6 characters	Only certain format accepted
Password	String	8 characters max.	-
Form	String	3 characters max.	Only certain format accepted
ImageLoc	String	6 characters	Only letters accepted
Year	String	1-2 characters	Only 7-13 accepted
Gamesplayed	Integer	∞	Only integers accepted, Wins + Losses + Draws = Gamesplayed
Wins	Integer	∞	Only integers accepted, Wins + Losses + Draws = Gamesplayed
Losses	Integer	∞	Only integers accepted, Wins + Losses + Draws = Gamesplayed
Draws	Integer	∞	Only integers accepted, Wins + Losses + Draws = Gamesplayed
tblTeachers			

TeacherID	Primary Key	∞	Only integers accepted
	Foreign Key	∞	Only integers accepted, only values present on tblSubjects valid
Fname	String	10 characters max.	Only letters accepted
Lname	String	15 characters max.	Only letters accepted
Username	String	6 characters	Only certain format accepted
Password	String	8 characters max.	-
ImageLoc	String	6 characters	Only letters accepted
QsAdded	Integer	∞	Only integers accepted, can't be more than the number of records in tblQuestions
tblSubjects			
SubjectID	Primary Key	∞	Only integers accepted
Subject	String	10 characters max.	Only letters accepted
Topic1	String	30 characters max.	-
Topic2	String	30 characters max.	-
Topic3	String	30 characters max.	-
Topic4	String	30 characters max.	-
Topic5	String	30 characters max.	-
Topic6	String	30 characters max.	-
tblQuestions			
QuestionID	Primary Key	∞	Only integers accepted
	Foreign Key	∞	Only integers accepted, only values present on tblSubjects valid
	Foreign Key	∞	Only integers accepted, only values present on

			tblTeachers valid
Question	String	255 characters max.	-
Answer	String	255 characters max	-
Year	String	6-7 characters	Only 7-13 accepted
Topic	String	30 characters max.	Only letters accepted
tblAttempted			
AttemptID	Primary Key	∞	Only integers accepted
	Foreign Key	∞	Only integers accepted, only values present on tblSubjects valid
	Foreign Key	∞	Only integers accepted, only values present on tblSubjects valid
When	Date & Time	18 – 25 characters	Only certain formats accepted
Correct	Character	1	Only 1 or 0 accepted

Here is an entity relationship diagram, a visual way of displaying the connections between the tables:



Also, BNF specifications for data validation:

Fname, Lname, Subject, Topic, Password, ImageLoc, Topic1-6, Question, Answer

```
<letter>::=a|b|c|d|e|f|g|h|i|j|k|l|m|n|o|p|q|r|s|t|u|v|w|x|y|z|A|B|C|D|E|F|
G|H|I|J|K|L|M|N|O|P|Q|R|S|T|U|V|W|X|Y|Z|-

<word>::=<letter><word>
```

Student Username

```
<letter>::=a|b|c|d|e|f|g|h|i|j|k|l|m|n|o|p|q|r|s|t|u|v|w|x|y|z

<num>::=1|2|3|4|5|6|7|8|9|0
```

```
<username>::=<num><num><letter><letter><num><num>
```

Teacher Username

```
<letter>::=a|b|c|d|e|f|g|h|i|j|k|l|m|n|o|p|q|r|s|t|u|v|w|x|y|z

<num>::=1|2|3|4|5|6|7|8|9|0

<username>::=st<letter><letter><num><num>
```

Year

```
<year>::=7|8|9|10|11|12|13
```

Correct

```
<value>::=0|1
```

When

```
<num>::=1|2|3|4|5|6|7|8|9|0

<day>::=Monday|Tuesday|Wednesday|Thursday|Friday|Saturday|Sunday

<month>::=January|February|March|April|May|June|July|August|September|
October|November|December

<nums>::=<num><nums>

<year>::=<num><num><num><num>

<date>::=<nums>:<nums>, <day> <nums>th <month> <year>
```

Form

```
<num>::=7|8|9|10|11|12|13

<letter>::=A|B|C|D|E|F|G|H

<form>::=<num><letter>
```

QsAnswered, QsAdded, Wins, Losses, Draws, Gamesplayed

```
<num>::=1|2|3|4|5|6|7|8|9|0

<nums>::=<num>|<num><nums>
```

There will be three .txt files, one for recording wins/losses/draws, one for recording the creation of new questions and one for detecting whether to run the first-time setup or not. The BNF specification for GameDetails.txt is as follows:

```
<letter>::=a|b|c|d|e|f|g|h|i|j|k|l|m|n|o|p|q|r|s|t|u|v|w|x|y|z|A|B|C|D|E|F|
G|H|I|J|K|L|M|N|O|P|Q|R|S|T|U|V|W|X|Y|Z|-

<num>::=1|2|3|4|5|6|7|8|9|0

<day>::=Monday|Tuesday|Wednesday|Thursday|Friday|Saturday|Sunday
```

```
<month>::=January|February|March|April|May|June|July|August|September|
October|November|December

<letters>::=<letter><letters>

<nums>::=<num><nums>

<year>::=<num><num><num><num>

<name>::=<letters> <letters>

<gameresult>::=beat|lost to|drew with

<game>::=Connect Four|Noughts and Crosses|Hangman|etc.

<date>::=<nums>:<nums>, <day> <nums>th <month> <year>

<line>::=<name> <gameresult> <name> in <game> - <date>
```

The BNF specification for QuestionCreation.txt:

```
<letter>::=a|b|c|d|e|f|g|h|i|j|k|l|m|n|o|p|q|r|s|t|u|v|w|x|y|z|A|B|C|D|E|F|
G|H|I|J|K|L|M|N|O|P|Q|R|S|T|U|V|W|X|Y|Z|-

<num>::=1|2|3|4|5|6|7|8|9|0

<day>::=Monday|Tuesday|Wednesday|Thursday|Friday|Saturday|Sunday

<month>::=January|February|March|April|May|June|July|August|September|
October|November|December

<year>::=Year <num>

<letters>::=<letter><letters>

<nums>::=<num><nums>

<year>::=<num><num><num><num>

<name>::=<letters> <letters>

<date>::=<nums>:<nums>, <day> <nums>th <month> <year>

<line>::=<name> created a question for <year> <letters> (“<letters>”) –
<date>
```

And finally, the BNF for FirstTimeSetup.txt:

```
<line>::=1|0
```

As the suite is to be used throughout the school network, the database and .txt documents should be stored on the network itself so that every computer connected can access them. This will also avoid any danger of the files being found and modified client-side, which could lead to cheating and errors.

Validation

Validation is the process of using a computer to ensure that entered data is valid but not necessarily correct. An example of this would be a type check for the entry of a name. For example, “Bob” would be accepted, whereas “Bo2b” would be rejected. “Bob II” would also be accepted, as the numbers are represented by “i”s. However, this does not mean that the data entered is correct. “Bab” would also be accepted, which is why validation needs to be paired up with additional verification to prove most successful.

In the previous section, I elaborated upon almost all of the validation I shall be using in my program. The only unmentioned piece is in the view student form, in which the textboxes for name input will accept only the letters a-z and “-“, for double-barrelled names.

Processing Stages

Module

```
This module defines global variables

Module Globals

      Declare global variables

      FnameStudent as String

      LnameStudent as String

      FnameOppStudent as String

      LnameOppStudent as String

      FnameTeacher as String

      LnameTeacher as String

      ImageStudent as Image

      ImageOppStudent as Image

      ImageTeacher as Image

      Form as String

      StudentID as Integer

      TeacherID as Integer

      GameMode as Integer = 0

      [0 = Network ConnectFour game, 1 = Hotseat ConnectFour game]

End
```

FirstTimeSetupMessageForm

```
FirstTimeSetupMessageForm

      This sub deals with the button for performing the first-time setup
      being clicked

      Click Yes Button

            Close Login form

            Start First Time Setup

            Close form

      End

      This sub deals with the button for not performing the first-time
      setup being clicked
```

```
        Click No Button
                Open FirstTimeSetup text file
                Clear all text
                Add "1"
                Close FirstTimeSetup text file
                Close form
        End
End
```

FirstTimeSetupForm

```
FirstTimeSetupForm
        This sub deals with the Add New Subject button being clicked
        Click Add New Subject button
                Show Add Subject form
        End
        This sub deals with the Add New Topic button being clicked
        Click Add New Topic button
                Show Add Topic form
        End
        This sub deals with the Add New Teacher button being clicked
        Click Add New Teacher button
                Show Add Teacher form
        End
        This sub deals with the Done button being clicked
        Click Done button
                Open FirstTimeSetup text file
                Clear all text
                Add "1"
                Close FirstTimeSetup text file
                Show Login form
                Close form
```

```
        End
End
```

AddSubjectForm

```
AddSubjectForm
        This sub deals with the Add Subject button being clicked
        Click Add Subject button
                Open database
                Select Subjects table
                Place input Subject Name into the Subject field
                Close database
        End
        This sub deals with the Back button being clicked
        Click Back button
                Show First Time Setup form
                Close form
        End
End
```

AddTopicForm

```
AddTopicForm
        This sub deals with the form loading
        Load Add Topic form
                Open database
                Select Subjects table
                Populate Subject Combobox with values in Subject field
                Close database
        End
        This sub deals with the Add Topic button being clicked
        Click Add Topic Button
                Open database
                Select Subjects table
```

```
            Find record that corresponds to selected subject

            Find next empty "Topic(x)" field

            Place input topic into the Topic(x) field

            Close database

      End

      This sub deals with the Back button being clicked

      Click Back button

            Show First Time Setup form

            Close form

      End

End
```

AddTeacherForm

```
AddTeacherForm

      This sub deals with the form loading

      Load Add Teacher form

            Open database

            Select Subjects table

            Populate Subject Comboboxes with values in Subject fields

            Close database

      End

      This sub deals with the Add Teacher button being clicked

      Click Add Teacher button

            Open database

            Select Teachers table

            Place Name into respective fields

            Place Subject(s) in correct field

            Generate Username via conventions and place in Username field

            Save image to folder, rename via conventions, add filename to
            ImageLoc field

            Place defaults in all other fields

            Close database
```

```
        End
        This sub deals with the Back button being clicked
        Click Back button
                Show First Time Setup form
                Close form
        End
End
```

AddStudentForm

```
AddStudentForm
        This sub deals with the Add Student button being clicked
        Click Add Student button
                Open database
                Select Students table
                Place Name, Form and Year into respective fields
                Generate Username via conventions and place in Username field
                Save image to folder, rename via conventions, add filename to
                ImageLoc field
                Place defaults in all other fields
                Close database
        End
        This sub deals with the Back button being clicked
        Click Back button
                Open FirstTimeSetup text file
                If Text = "0" then
                        Show First Time Setup form
                Else
                        Show Teacher Account form
                End
                Close FirstTimeSetup text file
                Close form
        End
```

```
End
```

AddQuestionForm

```
AddQuestionForm
      This sub deals with the form loading
      Load Add Question form
            Open database
            Select Subjects table
            Populate Subject combobox with values in Subject field
            Populate Topic combobox with values in Topic(x) fields
            Add line to relevant QuestionCreation text file
            Close database
      End
      This sub deals with the Submit button being clicked
      Click Submit button
            Open Database
            Select Questions table
            Place SubjectID in SubjectID field
            Place TeacherID TeacherID field
            Place Topic in Topic Field
            Place Question, Answer in Question, Answer fields
            Place defaults in all other fields
      End
      This sub deals with the Back button being clicked
      Click Back button
            Open FirstTimeSetup text file
            If line = "0" then
                  Show First Time Setup form
            Else
                  Show Teacher Account form
            End
```

```
            Close FirstTimeSetup text file
            Close form
      End
End
```

LoginForm

```
LoginForm
      This sub deals with the form loading
      Load Login form
            Open FirstTimeSetup text file
            If text = 0 then
                  Show First Time Setup message
            End
            Close FirstTimeSetup text file
            Play intro music
      End
      This sub deals with the log in button being clicked
      Click Okay button
            Open database
            Select Students and Teachers tables
            Search for username and password
            If Username and Password = valid then
                  If Login Details = student login then
                        StudentID = StudentID in database
                        FnameStudent = student first name
                        LnameStudent = student last name
                        ImageStudent = student image
                        Form = student form
                        Show Student Home form
                        Close form
                  Elseif Login Details = teacher login then
```

```
                        TeacherID = TeacherID in database
                        FnameTeacher = teacher first name
                        LnameTeacher = teacher last name
                        ImageTeacher = teacher image
                        Show Teacher Home form
                        Close form
                  End
            Else
                  Message box = "Invalid login details"
                  Clear Username and Password textboxes
            End
      End
      This sub deals with the Cancel button being clicked
      Click Cancel button
            Close form
      End
End
```

StudentHomeForm

```
StudentHomeForm
      This sub deals with the form loading
      Load Student Home form
            Account Section Sub
      End
      This sub deals with the Connect Four button being moused over
      Mouseover Connect Four button
            GamePicture = Connect Four image
            GameDesc = "Appropriate description"
      End
      This sub deals with the Hangman button being moused over
      Mouseover Hangman button
```

```
            GamePicture = Hangman image
            GameDesc = "Appropriate description"
        End
        This sub deals with the form being moused over
        Mouseover Student Home form
            Clear GamePicture and GameDesc
        End
        This sub deals with the Connect Four button being clicked
        Click Connect Four button
            Show Connect Four Main Menu form
            Close form
        End
        This sub deals with the Hangman button being clicked
        Click Hangman button
            Show Hangman Main Menu form
            Close form
        End
        This sub deals with the View Profile button being clicked
        Click View Profile button
            Show Student Profile
        End
        This subroutine populates the account details section in the top-
        right corner
        Account Section Sub
            ProfilePicture = ImageStudent
            ProfileName = FnameStudent & LnameStudent
        End
End
```

TeacherHomeForm

```
TeacherHomeForm
        This sub deals with the form loading
```

```
        Load Teacher Home form
              Account Section Sub
        End
        This sub deals with the View Student button being clicked
        Click View Student button
              Show View Student form
        End
        This sub deals with the Add Student button being clicked
        Click Add Student button
              Show Add Student form
        End
        This sub deals with the Create Question button being clicked
        Click Create Question button
              Show Create Question form
        End
        This sub deals with the View Profile button being clicked
        Click View Profile button
              Show Teacher Profile form
        End
        This subroutine populates the account details section in the top-
        right corner
        Account Section Sub
              ProfilePicture = ImageTeacher
              ProfileName = FnameTeacher & LnameTeacher
        End
End
```

ViewStudentForm

```
ViewStudentForm
        This sub deals with the form loading
        Load Teacher Home form
              Account Section Sub
```

```
End

This sub deals with the Search button being clicked

Click Search button

      Open database

      Select Students table

      Search for First Name and Last Name

      If found then

            Populate listbox with matching name and year

      Elseif

            Message box = "No such student found"

      End

End

This sub deals with the Listbox being clicked

Click Listbox Element

      StudentID = Selected student's StudentID in database

End

This sub deals with the View Profile button being clicked

Click View Profile button

      Show Student Account form

      Close database

End

This subroutine populates the account details section in the top-
right corner

Account Section Sub

      ProfilePicture = ImageTeacher

      ProfileName = FnameTeacher & LnameTeacher

End

This sub deals with the Back button being clicked

Click Back button

      Show Teacher Home form

      Close form
```

```
	End
End
```

StudentAccountForm

```
StudentAccountForm
	This sub deals with the form loading

	Load Student Account form
		Name Label = FnameStudent & LnameStudent
		Form Label = "Form: " & Form
		StudentPicture = ImageStudent
		Open database
		Find record matching StudentID
		Populate labels in Results Groupbox with results from database
		Close database
	End
	This sub deals with the Details button being clicked
	Click Details button
		Show Student Details form
	End
	This sub deals with the Achievements button being clicked
	Click Achievements button
		Show Student Achievements form
	End
	This sub deals with the Back button being clicked
	Click Back button
		Close form
	End
End
```

TeacherAccountForm

```
TeacherAccountForm
```

```
        This sub deals with the form loading
        Load Teacher Account form
                Name Label = FnameTeacher & LnameTeacher
                TeacherPicture = ImageTeacher
                Open database
                Select Teachers table
                Find record matching TeacherID
                Teaches Label = "Teaches: " & Subjects
                Populate Questions Created Label with data from database
                Close database
        End
        This sub deals with the Details button being clicked
        Click Details button
                Click Teacher Details form
        End
        This sub deals with the Achievements button being clicked
        Click Achievements button
                Click Teacher Achievements form
        End
        This sub deals with the Back button being clicked
        Click Back button
                Close form
        End
End
```

StudentDetailsForm

```
StudentDetailsForm
        This sub deals with the form loading
        Load Student Details form
                Name Label = FnameStudent & LnameStudent & " win breakdown"
                Open relevant GameResults text file
```

```
            Populate listbox with text

            Close GameResults text file

      End

      This sub deals with the Back button being clicked

      Click Back button

            Show Student Account form

            Close form

      End

End
```

TeacherDetailsForm

```
TeacherDetailsForm

      This sub deals with the form loading

      Load Teacher Details form

            Name Label = FnameTeacher & LnameTeacher & " question
            breakdown"

            Open relevant QuestionCreation text file

            Populate listbox with text

            Close QuestionCreation text file

      End

      This sub deals with the Back button being clicked

      Click Back button

            Show Teacher Account form

            Close form

      End

End
```

StudentAchievementsForm

```
StudentAchievementsForm

      This sub deals with the form loading

      Load Student Achievements form

            Name Label = FnameStudent & LnameStudent & " achievements"
```

```
            Achievements Sub

        End

        This subroutine determines which achievements are locked and unlocked
        and populates the form accordingly

        Achievements Sub

            Open database

            Search for all Achievement criteria

            If Achievement Unlocked, change picture and labels to relevant
            unlocked ones

        End

        This sub deals with the Back button being clicked

        Click Back button

            Show Student Account form

            Close form

        End

End
```

TeacherAchievementsForm

```
TeacherAchievementsForm

        This sub deals with the form loading

        Load Teacher Achievements form

            Name Label = FnameTeacher & LnameTeacher & " achievements"

            Achievements Sub

        End

        This subroutine determines which achievements are locked and unlocked
        and populates the form accordingly

        Achievements Sub

            Open database

            Search for all Achievement criteria

            If Achievement Unlocked, change picture and labels to relevant
            unlocked ones

        End

        This sub deals with the Back button being clicked
```

```
	Click Back button
		Show Teacher Account form
		Close form
	End
End
```

ConnectFourMainMenuForm

```
ConnectFourMainMenuForm
	This sub deals with the form loading
	Load Connect Four Main Menu form
		Account Section Sub
	End
	This sub deals with the Find Game button being clicked
	Click Find Game button
		Show Connect Four Game Finder form
		Close form
	End
	This sub deals with the Hotseat Game button being clicked
	Click Hotseat Game button
		Gamemode = 1
		Show Connect Four form
		Close form
	End
	This sub deals with the View Profile button being clicked
	Click View Profile button
		Show Student Account form
		Close database
	End
	This subroutine populates the account details section in the top-
	right corner
	Account Section Sub
		ProfilePicture = ImageTeacher
```

```
            ProfileName = FnameTeacher & LnameTeacher
        End
        This sub deals with the Back button being clicked
        Click Back button
            Show Student Home form
            Close form
        End
End
```

ConnectFourGameFinderForm

```
ConnectFourGameFinderForm
        This sub deals with the form loading
        Load Connect Four Game Finder form
            Populate Lobby Listbox with currently available games
            Account Section Sub
        End
        This sub deals with the Create Game button being clicked
        Click Create Game button
            Create game with name, subject, year and topic specified in
            comboboxes
            Show Connect Four form
            Close form
        End
        This sub deals with the Join Game button being clicked
        Click Join Game button
            Join selected game
            Close form
        End
        This sub deals with the View Profile button being clicked
        Click View Profile button
            Show Student Account form
            Close database
```

```
        End

        This subroutine populates the account details section in the top-
        right corner

        Account Section Sub

              ProfilePicture = ImageTeacher

              ProfileName = FnameTeacher & LnameTeacher

        End

        This sub deals with the Back button being clicked

        Click Back button

              Show Connect Four Main Menu form

              Close form

        End

End
```

ConnectFourForm

```
ConnectFourForm

        Declares a variable to be used later

        HoldStudentID as Integer

        This sub deals with the form loading

        Load Connect Four form

              Account Section Sub

              Assemble grid

              Set starting player

        End

        This sub deals with the Column button being clicked

        Click Column button

              Find next empty spot on column

              Place current player's counter there

              Detect Winner Sub

              Change Player Sub

        End
```

```
This subroutine checks the game grid to determine if the previous
move has caused a win condition to trigger

Detect Winner Sub

      Search for four counters of the same colour in a row
      horizontally, vertically or diagonally

      If four counters in a row = true then

            Declare current player winner

            Open database

            Find record matching current player

            Add a win

            Find record matching current opponent

            Add a loss

            Close database

            Open winning player's GameResults text file

            Add a line

            Close text file

            Open losing player's GameResults text file

            Add a line

            Close text file

            Show reset button

      End

End

This sub deals with the Reset button being clicked

Click Reset button

      Reset grid and player

End

This sub deals with the View Profile button being clicked

Click View Profile button

      Show Student Account form

      Close database

End
```

```
        This sub deals with the View Opp Profile button being clicked
        Click View Opp Profile button
              HoldStudentID = StudentID
              StudentID = OppStudentID
              Show Student Account form
              StudentID = HoldStudentID
        End
        This subroutine populates the account details section in the top-
        right corner
        Account Section Sub
              ProfilePicture = ImageStudent
              ProfileName = FnameStudent & LnameStudent
              OppProfilePicture = ImageOppStudent
              OppProfileName = FnameOppStudent & LnameOppStudent
        End
        This sub deals with the Back button being clicked
        Click Back button
              Show Connect Four Game Finder form
              Close form
        End
        This subroutine changes which player is currently up
        Change Player Sub
              If Gamemode = 0 then
                    Change player to opponent over network
              Elseif Gamemode = 1 then
                    Change player on current system
              End
        End
End
```

HangmanMainMenuForm

```
HangmanMainMenuForm
```

```
        This sub deals with the form loading
        Load Hangman Main Menu form
              Account Section Sub
        End
        This sub deals with the Play button being clicked
        Click Play button
              Show Hangman form
              Close form
        End
        This sub deals with the View Profile button being clicked
        Click View Profile button
              Show Student Account form
              Close database
        End
        This subroutine populates the account details section in the top-
        right corner
        Account Section Sub
              ProfilePicture = ImageTeacher
              ProfileName = FnameTeacher & LnameTeacher
        End
        This sub deals with the Back button being clicked
        Click Back button
              Show Student Home form
              Close form
        End
End
```

HangmanForm

```
HangmanForm
        Declares a variable to be used later
        HoldStudentID as Integer
        This sub deals with the form loading
```

```
Load Hangman form
      Account Section Sub
      Assemble hangman and blank space for word
End
This sub deals with the Letter button being clicked
Click Letter button
      Disable Letter button
      If Letter is present in word then
            Display all instances of Letter
      Elseif
            Display next stage of hangman
            If Hangman = completed then
                  Declare player loser
                  Open database
                  Find record matching current player
                  Add a loss
                  Close database
                  Open winning player's GameResults text file
                  Add a line
                  Close text file
                  Show reset button
            End
      End
      If Word = complete then
            Declare player winner
            Open database
            Find record matching current player
            Add a win
            Close database
            Open winning player's GameResults text file
```

```
                Add a line
                Close text file
                Show reset button
            End
        End
        This sub deals with the Reset button being clicked
        Click Reset Button
            Reset word and hangman
        End
        This sub deals with the View Profile button being clicked
        Click View Profile button
            Show Student Account form
            Close database
        End
        This subroutine populates the account details section in the top-
        right corner
        Account Section Sub
            ProfilePicture = ImageStudent
            ProfileName = FnameStudent & LnameStudent
        End
        This sub deals with the Back button being clicked
        Click Back button
            Show Hangman Main Menu form
            Close form
        End
End
```

Evaluation Criteria

- A new user, who has never experienced the program before and who is at least slightly computer literate, must be able to complete the first-time setup in less than five minutes with no assistance.
- The user must be able to start and both win and lose a game of Hangman, with the appropriate result recorded in his account's details, within 10 minutes.
- The user must be able to start and win, lose and draw a game of Noughts & Crosses in hotseat mode, with the appropriate result recorded in his account's details, within 10 minutes.
- The user must be able to log in as both a student and as a teacher in under a minute.
- The user must be able to view his account details within 30 seconds of the Home form being displayed.
- Both the student and the teacher must have distinct and tailored experiences
    - A teacher must be able to view the details of a specific student within 2 minutes.
    - A student must be able to start a game within 2 minutes
- The program will work both in school and out of it.

Maintenance Documentation

Annotated Listing

mdlPublicVars.vb

```
'Add a reference to COM Microsoft ActiveX Data Objects 6.1
Imports System
Imports System.Data
Imports System.Data.Odbc
Module mdlPublicVars

    '//Connect Four\\
    'Declares the variable used for storing the current player of a game of Connect
Four
    Public C4Player As String
    'Declares the variables used for determining the questions to display in a game of
Connect Four
    Public C4HSubject, C4HDifficulty, C4HTopic As String

    '//Noughts and Crosses\\
    'Declares the variable used for storing the current player of a game of Noughts
and Crosses
    Public NaCPlayer As String
    'Declares the variables used for determining the questions to display in a game of
Noughts and Crosses
    Public NaCSubject, NaCDifficulty, NaCTopic As String

    '//Rock, Paper, Scissors\\
    'Declares the variable used for storing the current player of a game of Rock,
Paper, Scissors
    Public RPSPlayer As String
    'Declares the variables used for determining the questions to display in a game of
Rock, Paper, Scissors
    Public RPSSubject, RPSDifficulty, RPSTopic As String
    'Declares the variables used to store the selected weapons of each player
    Public LoggedInWeapon, OppWeapon As Integer

    '//General\\

    '/General\

    'Declares the variable used for storing the relative path to the program
    Public Path As String
    'Declares the variable used for storing the path to the database
    Public DBPath As String
    'Declares the variable used to connect to mySQL database
    Public DBConn As ADODB.Connection
    'Subroutine runs when called
    Public Sub OpenDB()
        'Declares the variable used for the server address of the database
        Dim Server As String = "moodle.bourne-grammar.lincs.sch.uk"
        'Declares the variable used for storing the database name
        Dim Dtabase As String = "ben"
        'Declares the variable used for storing the username for the database
        Dim UID As String = "ben"
        'Declares the variable used for storing the password for the database
        Dim Pwd As String = "123456"

        'Declares the variable used for connecting to the database
        Dim ConnStr As String

        'Builds the connection string for the database
```

```
        ConnStr = "DRIVER={MySQL ODBC 5.1 Driver};SERVER=" & Server & ";DATABASE=" &
Dtabase & ";UID=" & UID & ";PWD=" & Pwd & ";OPTION=3"
        DBConn = New ADODB.Connection
        DBConn.ConnectionString = ConnStr
        DBConn.CursorLocation = ADODB.CursorLocationEnum.adUseClient

        'Opens the connection
        DBConn.Open()
    End Sub
    'Declares the variable used for storing the GameID for network play
    Public GameID As String

    '/Students\

    'Declares the variables used for storing the details of both players
    Public FnameStudent, LnameStudent, FnameOppStudent, LnameOppStudent, Form,
ImageStudentLoc, ImageOppStudentLoc As String
    'Declares the variable used for storing the StudentID of a player
    Public StudentID As Integer
    'Declares the class used for both players
    Public Class Student
        'Declares the variables used for storing the details of the student
        Public Fname, Lname, Form, Username, C4Player, NaCPlayer As String
        Public StudentID, Wins, Losses, Draws, RPSPlayer As Integer
    End Class
    'Creates two objects of the student class
    Public LoggedInStudent As New Student
    Public OppStudent As New Student
    'Used to populate the Student Account form with the correct data
    Public Viewing As Integer = 1

    '/Teachers\

    'Declares the variables used for storing the details of the teacher
    Public FnameTeacher, LnameTeacher, imageteacherloc As String
    'Declares the variable used for storing the TeacherID
    Public TeacherID As Integer
    'Declares the class used for the teacher
    Public Class Teacher
        'Declares the variables used for storing the details of the teacher
        Public Fname, Lname, Username As String
        Public TeacherID As Integer
    End Class
    'Creates an object of the teacher class
    Public LoggedInTeacher As New Teacher

End Module
```

frmSplash.vb

```
Public Class frmSplash
    'Declares the variable used for counting down the loading
    Dim Count As Integer = 0

    'Subroutine runs when the form loads
    Private Sub frmSplash_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        'Enables the timer used to simulate a loading
        tmrLoading.Enabled = True
     End Sub
```

```
    'Subroutine runs when timer is enabled
    Private Sub tmrLoading_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles tmrLoading.Tick
        'Runs the random number generator subroutine
        Randomize()

        'Declares the variable used for determining how big a step to take
        Dim Tick As Integer

        'If there is more than 10 counts remaining in the timer...
        If progLoading.Value < 990 Then
            'Sets the step size as a random number between 0-9
            Tick = CInt(Int(10 * Rnd()))
            'If there are only 10 counts remaining...
        Else
            'Sets the step size as a random number within the bounds of the timer
            Tick = CInt(Int((1000 - progLoading.Value) * Rnd()) + 1)
        End If

        'On each tick of the timer, 'Count' is increased by the step size
        Count = Count + Tick
        'Along with this, the progress bar increments by the step size
        progLoading.Value = progLoading.Value + Tick

        'If the timer has reached its limit...
        If Count = 1000 Then
            'Disable the timer
            tmrLoading.Enabled = False
            'Show the login form
            frmLogin.Show()
        End If
    End Sub

End Class
```

frmFTSMsg.vb

```
Public Class frmFTSMsg

    'Subroutine runs when the yes button is clicked
    Private Sub btnYes_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnYes.Click
        'Closes the login form
        frmLogin.Close()
        'Opens the first-time setup form
        frmFirstTimeSetup.Show()
        'Closes this form
        Me.Close()
    End Sub

    'Subroutine runs when the no button is clicked
    Private Sub btnNo_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnNo.Click
        'Declares the variable used for writing to the text file
        Dim writer As System.IO.StreamWriter
        'Gets the filepath to the first-time setup text file
        writer = My.Computer.FileSystem.OpenTextFileWriter("G:\Computing Group\Y10
2009-10\Ben's\Edutainment Suite\Connect4\bin\Debug\FTS.txt", False)
        'Amends the text file to indicate that the first-time setup has been run
        writer.WriteLine("1")
        'Saves the log file
```

```
        writer.Close()
        'Closes this form
        Me.Close()
    End Sub

End Class_
```

frmFirstTimeSetup.vb

```
Public Class frmFirstTimeSetup

    'Subroutine runs when the add subject button is clicked
    Private Sub btnAddSubject_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnAddSubject.Click
        'Opens the add subject form
        frmAddSubject.Show()
    End Sub

    'Subroutine runs when the add topic button is clicked
    Private Sub btnAddTopic_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnAddTopic.Click
        'Opens the add topic form
        frmAddTopic.Show()
    End Sub

    'Subroutine runs when the add teacher button is clicked
    Private Sub btnAddTeacher_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnAddTeacher.Click
        'Opens the add teacher form
        frmAddTeacher.Show()
    End Sub

    'Subroutine runs when the add student button is clicked
    Private Sub btnAddStudent_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnAddStudent.Click
        'Opens the add student form
        frmAddStudent.Show()
    End Sub

    'Subroutine runs when the done button is clicked
    Private Sub btnDone_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnDone.Click
        'Declares the variable used for writing to the text file
        Dim writer As System.IO.StreamWriter
        'Gets the filepath to the first-time setup text file
        writer = My.Computer.FileSystem.OpenTextFileWriter("G:\Computing Group\Y10
2009-10\Ben's\Edutainment Suite\Connect4\bin\Debug\FTS.txt", False)
        'Amends the text file to indicate that the first-time setup has been run
        writer.WriteLine("1")
        'Saves the log file
        writer.Close()

        'Opens the login form
        frmLogin.Show()
        'Closes this form
        Me.Close()
    End Sub

End Class
```

frmAddSubject.vb

```
Public Class frmAddSubject

    'Subroutine runs when the form loads
    Private Sub frmAddSubject_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        'Establishes the connection to the database
        OpenDB()
    End Sub

    'Subroutine runs when the add subject button is clicked
    Private Sub btnAddSubject_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnAddSubject.Click
        'Declares the variable used for adding a new record into the database
        Dim RSNewRec As New ADODB.Recordset
        If RSNewRec.State = ADODB.ObjectStateEnum.adStateOpen Then RSNewRec.Close()
        'Declares the variable used for detecting invalid data entry
        Dim Errors As String = ""

        'Assembles an error report if any invalid data entry detected
        If txtSubjectName.Text = "" Then
            Errors = Errors & "No subject name input" & vbCrLf
        End If

        'If no invalid data entry is detected...
        If Errors = "" Then
            'Opens a recordset
            RSNewRec.Open("SELECT * FROM tblsubjects", DBConn, ,
ADODB.LockTypeEnum.adLockOptimistic)
            'Adds a new record and fills it with information before closing
            With RSNewRec
                .AddNew()
                .Fields("Subject").Value = txtSubjectName.Text
                RSNewRec.Update()
                RSNewRec.Close()
            End With

            'Creates the topics text file for the subject
            Dim file As System.IO.FileStream
            file = System.IO.File.Create("G:\Computing Group\Y10 2009-
10\Ben's\Edutainment Suite\Connect4\bin\Debug\Topics\" & txtSubjectName.Text &
"Topics.txt")

            'If any invalid data entry is detected...
        Else
            'Displays a message box with any detected invalid data entry
            MsgBox("Invalid input:" & vbCrLf & vbCrLf & Errors)
        End If
    End Sub

    'Subroutine runs when the subject name text box is clicked
    Private Sub txtSubjectName_Click(ByVal sender As Object, ByVal e As
System.EventArgs) Handles txtSubjectName.Click
        'Blanks out the text box
        txtSubjectName.Text = ""
    End Sub

End Class
```

frmAddTopic.vb

```
Public Class frmAddTopic
```

```
    'Subroutine runs when the form loads
    Private Sub frmAddTopic_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        'Establishes the connection to the database
        OpenDB()

        'Runs the form population subroutine
        Populate()
    End Sub

    'Subroutine runs when the add topic button is clicked
    Private Sub btnAddTopic_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnAddTopic.Click
        'Runs the form population subroutine
        CheckValid()
    End Sub

    'Subroutine runs when called in the form load sub
    Sub Populate()
        'Declares the variable used for reading data from the database
        Dim RSStaff As New ADODB.Recordset
        If RSStaff.State = ADODB.ObjectStateEnum.adStateOpen Then RSStaff.Close()

        'Builds SQL query to execute
        RSStaff.Open("SELECT * FROM tblsubjects", DBConn, ,
ADODB.LockTypeEnum.adLockOptimistic)

        'Whilst the end of the database hasn't been reached...
        While Not RSStaff.EOF
            'Populates the subject combobox with data
            cmboSubject.Items.Add(RSStaff.Fields("Subject").Value)
            'Moves onto the next record
            RSStaff.MoveNext()
        End While

        'Closes the recordset
        RSStaff.Close()
    End Sub

    'Subroutine runs when called in the add topic button sub
    Sub CheckValid()
        'Declares the variable used for getting the result of the message box
        Dim Result As MsgBoxResult
        'Declares the variable used for detecting invalid data entry
        Dim Errors As String = ""

        'Assembles an error report if any invalid data entry detected
        If cmboSubject.Text = "" Then
            Errors = Errors & "No subject selected" & vbCrLf
        End If
        If cmboYear.Text = "" Then
            Errors = Errors & "No year selected" & vbCrLf
        End If
        If txtTopic.Text = "" Then
            Errors = Errors & "No topic input" & vbCrLf
        End If

        'If no invalid data entry is detected...
        If Errors = "" Then
            'Displays a validation message box before saving the data to the database
```

```
            Result = MsgBox("Are you sure all these details are correct? Remember,
spelling is vital." & vbCrLf & vbCrLf & "Details:" & vbCrLf & cmboSubject.SelectedItem
& " (" & cmboYear.SelectedItem & ")" & vbCrLf & txtTopic.Text, MsgBoxStyle.YesNo)
            'If the data is approved by the user...
            If Result = MsgBoxResult.Yes Then
                'Declares the variable used for writing to the text file
                Dim writer As System.IO.StreamWriter
                'Gets the filepath to the selected subject's topic text file, creating
it if it doesn't exist
                writer = My.Computer.FileSystem.OpenTextFileWriter("G:\Computing
Group\Y10 2009-10\Ben's\Edutainment Suite\Connect4\bin\Debug\Topics\" &
cmboSubject.Text & "Topics.txt", True)
                'Adds the new topic to the selected subject's topics text file
                writer.WriteLine(txtTopic.Text & " (" & cmboYear.Text & ")")
                'Saves the text file
                writer.Close()
            End If
            'If any invalid data entry is detected...
        Else
            'Displays a message box with any detected invalid data entry
            MsgBox("Invalid input:" & vbCrLf & vbCrLf & Errors)
        End If
    End Sub

End Class
```

frmAddTeacher.vb

```
Public Class frmAddTeacher

    'Declares the variable used for generating the username of the new student
    Dim Uname As String
    'Declares the variable used for getting the image of the new student
    Dim Filepath As String

    'Subroutine runs when the form loads
    Private Sub frmAddTeacher_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        'Establishes the connection to the database
        OpenDB()

        'Indicates that no picture has been selected
        picPic.Tag = 0
    End Sub

    'Subroutine runs when the teacher first name text box is clicked
    Private Sub txtTeacherFirstName_Click(ByVal sender As Object, ByVal e As
System.EventArgs) Handles txtTeacherFirstName.Click
        'Blanks out the text box
        txtTeacherFirstName.Text = ""
    End Sub

    'Subroutine runs when the teacher last name text box is clicked
    Private Sub txtTeacherLastName_Click(ByVal sender As Object, ByVal e As
System.EventArgs) Handles txtTeacherLastName.Click
        'Blanks out the text box
        txtTeacherLastName.Text = ""
    End Sub

    'Subroutine runs when the add teacher button is clicked
```

```
    Private Sub btnAddTeacher_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnAddTeacher.Click
        'Declares the variable used for getting the result of the message box
        Dim Result As MsgBoxResult
        'Declares the variable used for adding a new record into the database
        Dim RSNewRec As New ADODB.Recordset
        If RSNewRec.State = ADODB.ObjectStateEnum.adStateOpen Then RSNewRec.Close()
        'Declares the variable used for detecting invalid data entry
        Dim Errors As String = ""

        'Assembles an error report if any invalid data entry detected
        If txtTeacherFirstName.Text = "" Then
            Errors = Errors & "No first name input" & vbCrLf
        End If
        If txtTeacherLastName.Text = "" Then
            Errors = Errors & "No last name input" & vbCrLf
        End If
        If picPic.Tag = 0 Then
            Errors = Errors & "No picture input" & vbCrLf
        End If

        'If no invalid data entry is detected...
        If Errors = "" Then
            'Runs the username generation subroutine
            Username()
            'Displays a validation message box before saving the data to the database
            Result = MsgBox("Are you sure all these details are correct? Remember,
spelling is vital." & vbCrLf & vbCrLf & "Details:" & vbCrLf & txtTeacherFirstName.Text
& " " & txtTeacherLastName.Text, MsgBoxStyle.YesNo)
            'If the data is approved by the user...
            If Result = MsgBoxResult.Yes Then
                'Opens a recordset
                RSNewRec.Open("SELECT * FROM tblteachers", DBConn, ,
ADODB.LockTypeEnum.adLockOptimistic)
                'Adds a new record and fills it with information before closing
                With RSNewRec
                    .AddNew()
                    .Fields("Fname").Value = txtTeacherFirstName.Text
                    .Fields("Lname").Value = txtTeacherLastName.Text
                    .Fields("Username").Value = Uname
                    .Fields("Password").Value = "password"
                    RSNewRec.Update()
                    RSNewRec.Close()
                End With

                'Creates the question creation text file for the teacher
                Dim file As System.IO.FileStream
                file = System.IO.File.Create("G:\Computing Group\Y10 2009-
10\Ben's\Edutainment Suite\Connect4\bin\Debug\QuestionCreation\" & Uname & ".txt")

                'Copies the image to the folder for student images and renames it to
the new student's username
                My.Computer.FileSystem.CopyFile(Filepath, "G:\Computing Group\Y10
2009-10\Ben's\Edutainment Suite\Connect4\bin\Debug\images\teachers\" & Uname & ".jpg",
FileIO.UIOption.AllDialogs, FileIO.UICancelOption.DoNothing)
            End If
            'If any invalid data entry is detected...
        Else
            'Displays a message box with any detected invalid data entry
            MsgBox("Invalid input:" & vbCrLf & vbCrLf & Errors)
        End If
    End Sub
```

```
    'Subroutine runs when called in the add teacher button sub
    Sub Username()
        'Declares the variable used for reading data from the database
        Dim RSStaff As New ADODB.Recordset
        If RSStaff.State = ADODB.ObjectStateEnum.adStateOpen Then RSStaff.Close()

        'Declares the variable used to get the first letter of the new teacher's first
name
        Dim L1 As String = Mid(txtTeacherFirstName.Text, 1, 1)
        'Declares the variable used to get the first letter of the new teacher's last
name
        Dim L2 As String = Mid(txtTeacherLastName.Text, 1, 1)

        'Declares the variable used to get the numbers at the end of the new teacher's
username
        Dim Numbers As Integer = 11

        'Assembles the beginning and middle of the username
        Uname = "st" & L1 & L2

        'Opens a recordset
        RSStaff.Open("SELECT * FROM tblteachers WHERE Username='" & Uname & Numbers &
"'", DBConn, , ADODB.LockTypeEnum.adLockOptimistic)
        'If any records with the same username are detected...
        While RSStaff.RecordCount > 0
            'Closes the recordset
            RSStaff.Close()
            'Increases the numbers on the end of the username
            Numbers = Numbers + 1
            'Opens a new recordset to check if the new username isn't taken
            RSStaff.Open("SELECT * FROM tblteachers WHERE Username='" & Uname &
Numbers & "'", DBConn, , ADODB.LockTypeEnum.adLockOptimistic)
        End While
        'Assembles the username
        Uname = Uname & Numbers
    End Sub

    'Subroutine runs when the browse pic button is clicked
    Private Sub btnBrowsePic_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnBrowsePic.Click
        'Sets the filter of the browse window to only allow image files
        diaPic.Filter = "Image Files|*.jpg;*.gif;*.bmp;*.png;*.jpeg|All Files|*.*"
        'Sets the starting directory of the browse window to the C: drive
        diaPic.InitialDirectory = "C:\"
        'Sets the filter index of the browse window
        diaPic.FilterIndex = 1
        'Sets the title of the browse window
        diaPic.Title = "Open File"
        'If okay button of browse window is clicked...
        If (diaPic.ShowDialog() = Windows.Forms.DialogResult.OK) Then
            'Sets filepath to image
            Filepath = diaPic.FileName
            'Sets displayed image to selected image file
            picPic.Image = Image.FromFile(Filepath)
            'Sets picture box tag to indicate that a picture has been selected
            picPic.Tag = 1
        End If
    End Sub

End Class
```

frmAddStudent.vb

```
Public Class frmAddStudent

    'Declares the variable used for generating the username of the new student
    Dim Uname As String
    'Declares the variable used for getting the image of the new student
    Dim Filepath As String

    'Subroutine runs when the form loads
    Private Sub frmAddStudent_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        'Establishes the connection to the database
        OpenDB()

        'Indicates that no picture has been selected
        picPic.Tag = 0
    End Sub

    'Subroutine runs when the student first name text box is clicked
    Private Sub txtStudentFirstName_Click(ByVal sender As Object, ByVal e As
System.EventArgs) Handles txtStudentFirstName.Click
        'Blanks out the text box
        txtStudentFirstName.Text = ""
    End Sub

    'Subroutine runs when the student last name text box is clicked
    Private Sub txtStudentLastName_Click(ByVal sender As Object, ByVal e As
System.EventArgs) Handles txtStudentLastName.Click
        'Blanks out the text box
        txtStudentLastName.Text = ""
    End Sub

    'Subroutine runs when the add student button is clicked
    Private Sub btnAddStudent_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnAddStudent.Click
        'Declares the variable used for getting the result of the message box
        Dim Result As MsgBoxResult
        'Declares the variable used for adding a new record into the database
        Dim RSNewRec As New ADODB.Recordset
        If RSNewRec.State = ADODB.ObjectStateEnum.adStateOpen Then RSNewRec.Close()
        'Declares the variable used for detecting invalid data entry
        Dim Errors As String = ""

        'Assembles an error report if any invalid data entry detected
        If txtStudentFirstName.Text = "" Then
            Errors = Errors & "No first name input" & vbCrLf
        End If
        If txtStudentLastName.Text = "" Then
            Errors = Errors & "No last name input" & vbCrLf
        End If
        If cmboYear.Text = "" Then
            Errors = Errors & "No year selected" & vbCrLf
        End If
        If cmboForm.Text = "" Then
            Errors = Errors & "No form selected" & vbCrLf
        End If
        If picPic.Tag = 0 Then
            Errors = Errors & "No picture input" & vbCrLf
        End If
```

```
        'If no invalid data entry is detected...
        If Errors = "" Then
            'Runs the username generation subroutine
            Username()
            'Displays a validation message box before saving the data to the database
            Result = MsgBox("Are you sure all these details are correct? Remember,
spelling is vital." & vbCrLf & vbCrLf & "Details:" & vbCrLf & txtStudentFirstName.Text
& " " & txtStudentLastName.Text & vbCrLf & cmboYear.SelectedItem &
cmboForm.SelectedItem, MsgBoxStyle.YesNo)
            'If the data is approved by the user...
            If Result = MsgBoxResult.Yes Then
                'Builds SQL query to execute
                RSNewRec.Open("SELECT * FROM tblstudents", DBConn, ,
ADODB.LockTypeEnum.adLockOptimistic)
                'Adds a new record and fills it with information before closing
                With RSNewRec
                    .AddNew()
                    .Fields("Fname").Value = txtStudentFirstName.Text
                    .Fields("Lname").Value = txtStudentLastName.Text
                    .Fields("Username").Value = Uname
                    .Fields("Password").Value = "password"
                    .Fields("FormNum").Value = cmboYear.SelectedItem
                    .Fields("FormLetter").Value = cmboForm.SelectedItem
                    RSNewRec.Update()
                    RSNewRec.Close()
                End With

                'Creates the wins and losses text file for the student
                Dim file As System.IO.FileStream
                file = System.IO.File.Create("G:\Computing Group\Y10 2009-
10\Ben's\Edutainment Suite\Connect4\bin\Debug\WinsLosses\" & Uname & ".txt")

                'Copies the image to the folder for student images and renames it to
the new student's username
                My.Computer.FileSystem.CopyFile(Filepath, "G:\Computing Group\Y10
2009-10\Ben's\Edutainment Suite\Connect4\bin\Debug\images\students\" & Uname & ".jpg",
FileIO.UIOption.AllDialogs, FileIO.UICancelOption.DoNothing)
            End If
            'If any invalid data entry is detected...
        Else
            'Displays a message box with any detected invalid data entry
            MsgBox("Invalid input:" & vbCrLf & vbCrLf & Errors)
        End If
    End Sub

    'Subroutine runs when called in the add student button sub
    Sub Username()
        'Declares the variable used for reading data from the database
        Dim RSStaff As New ADODB.Recordset
        If RSStaff.State = ADODB.ObjectStateEnum.adStateOpen Then RSStaff.Close()

        'Declares the variable used to get the first letter of the new student's first
name
        Dim L1 As String = Mid(txtStudentFirstName.Text, 1, 1)
        'Declares the variable used to get the first letter of the new student's last
name
        Dim L2 As String = Mid(txtStudentLastName.Text, 1, 1)

        'Declares the variable used to get the numbers at the end of the new student's
username
        Dim Numbers As Integer = 11
```

```
        'Assembles the beginning and middle of the username
        Uname = "95" & L1 & L2

        'Builds SQL query to execute
        RSStaff.Open("SELECT * FROM tblstudents WHERE Username='" & Uname & Numbers &
"'", DBConn, , ADODB.LockTypeEnum.adLockOptimistic)
        'If any records with the same username are detected...
        While RSStaff.RecordCount > 0
            'Closes the recordset
            RSStaff.Close()
            'Increases the numbers on the end of the username
            Numbers = Numbers + 1
            'Opens a new recordset to check if the new username isn't taken
            RSStaff.Open("SELECT * FROM tblstudents WHERE Username='" & Uname &
Numbers & "'", DBConn, , ADODB.LockTypeEnum.adLockOptimistic)
        End While
        'Assembles the username
        Uname = Uname & Numbers
    End Sub

    'Subroutine runs when the browse pic button is clicked
    Private Sub btnBrowsePic_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnBrowsePic.Click
        'Sets the filter of the browse window to only allow image files
        diaPic.Filter = "Image Files|*.jpg;*.gif;*.bmp;*.png;*.jpeg|All Files|*.*"
        'Sets the starting directory of the browse window to the C: drive
        diaPic.InitialDirectory = "C:\"
        'Sets the filter index of the browse window
        diaPic.FilterIndex = 1
        'Sets the title of the browse window
        diaPic.Title = "Open File"
        'If okay button of browse window is clicked...
        If (diaPic.ShowDialog() = Windows.Forms.DialogResult.OK) Then
            'Sets filepath to image
            Filepath = diaPic.FileName
            'Sets displayed image to selected image file
            picPic.Image = Image.FromFile(Filepath)
            'Sets picture box tag to indicate that a picture has been selected
            picPic.Tag = 1
        End If
    End Sub

End Class
```

frmLogin.vb

```
Public Class frmLogin

    'Declares the variables used for logging in
    Dim EnteredUsername, EnteredPassword As String

    'Subroutine runs when the form loads
    Private Sub frmLogin_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        'Closes the splash screen form
        frmSplash.Close()

        'Sets 'Path' to the the filepath leading to the bin/debug/ folder of the
program, allowing portability.
```

```
        Path =
System.IO.Path.GetDirectoryName(System.Reflection.Assembly.GetExecutingAssembly().GetN
ame().CodeBase)

        'Runs the first-time setup detection subroutine
        DetectFTS()
    End Sub

    'Subroutine runs when the OK button is clicked
    Private Sub btnOK_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnOK.Click
        'Runs the login subroutine
        Login()
    End Sub

    'Subroutine runs when the cancel button is clicked
    Private Sub btnCancel_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnCancel.Click
        'Closes the program
        Me.Close()
    End Sub

    'Subroutine runs when called in the form load sub
    Sub DetectFTS()
        'Declares the variable used to determine whether first-time setup has been run
        Dim FTS As String
        'Declares the variable used to read the first-time setup text file
        Dim Reader As System.IO.StreamReader

        'Gets the path to the FTS.txt text file
        Reader = My.Computer.FileSystem.OpenTextFileReader("G:\Computing Group\Y10
2009-10\Ben's\Edutainment Suite\Connect4\bin\Debug\FTS.txt")

        'Reads what is in the FTS.txt text file
        FTS = Reader.ReadToEnd
        'If the text file consists of 0 then the first-time setup has not been run
before, so...
        If FTS = "0" Then
            'Opens the first-time setup message form
            frmFTSMsg.Show()
        End If
        'Closes the reader
        Reader.Close()
    End Sub

    'Subroutine runs when called in the OK button click sub
    Sub Login()
        'Establishes the connection to the database
        OpenDB()

        'Sets the EnteredUsername variable to the entered username
        EnteredUsername = txtUsername.Text
        'Sets the EnteredPassword variable to the entered password
        EnteredPassword = txtPassword.Text

        'Declares the variable used for reading data from the database
        Dim RSStaff As New ADODB.Recordset
        If RSStaff.State = ADODB.ObjectStateEnum.adStateOpen Then RSStaff.Close()

        'Builds SQL query to execute
```

```
        RSStaff.Open("SELECT * FROM tblstudents WHERE Username='" & EnteredUsername &
"' AND Password='" & EnteredPassword & "'", DBConn, ,
ADODB.LockTypeEnum.adLockOptimistic)

        'If records are found within the students table...
        If RSStaff.RecordCount > 0 Then
            'Fills the various properties of the LoggedInStudent object with their
respective values from the database
            With LoggedInStudent
                .Fname = RSStaff.Fields("Fname").Value
                .Lname = RSStaff.Fields("Lname").Value
                .Form = RSStaff.Fields("FormNum").Value &
RSStaff.Fields("FormLetter").Value
                .Wins = RSStaff.Fields("Wins").Value
                .Losses = RSStaff.Fields("Losses").Value
                .Draws = RSStaff.Fields("Draws").Value
                .Username = RSStaff.Fields("Username").Value
                .StudentID = RSStaff.Fields("StudentID").Value
            End With

            'Opens the student home form
            frmStudentHome.Show()
            'Closes this form
            Me.Close()
            'If no records are found within the students table...
        Else
            'Closes the recordset
            RSStaff.Close()
            'Builds SQL query to execute
            RSStaff.Open("SELECT * FROM tblteachers WHERE Username='" &
EnteredUsername & "' AND Password='" & EnteredPassword & "'", DBConn, ,
ADODB.LockTypeEnum.adLockOptimistic)

            'If records are found within the teachers table...
            If RSStaff.RecordCount > 0 Then
                'Fills the various properties of the LoggedInTeacher object with their
respective values from the database
                With LoggedInTeacher
                    .Fname = RSStaff.Fields("Fname").Value
                    .Lname = RSStaff.Fields("Lname").Value
                    .Username = RSStaff.Fields("Username").Value
                    .TeacherID = RSStaff.Fields("TeacherID").Value
                End With

                'Opens the teacher home form
                frmTeacherHome.Show()
                'Closes this form
                Me.Close()
                'If no records are found in either table...
            Else
                'Displays an error message
                MsgBox("Invalid: Incorrect username or password.")
                'Blanks out the username textbox
                txtUsername.Text = ""
                'Blanks out the password textbox
                txtPassword.Text = ""
            End If
        End If
    End Sub

End Class_
```

frmStudentHome.vb

```
Public Class frmStudentHome

    'Subroutine runs when the form loads
    Private Sub frmStudentHome_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        'Runs the account section population subroutine
        AccountSection()
    End Sub

    'Subroutine runs when called in the form load sub
    Sub AccountSection()
        'Places the logged-in student's name onto the form
        lblStudentName.Text = LoggedInStudent.Fname & " " & LoggedInStudent.Lname
        'Places the logged-in student's picture onto the form
        picStudent.ImageLocation = "G:\Computing Group\Y10 2009-10\Ben's\Edutainment
Suite\Connect4\bin\Debug\images\students\" & LoggedInStudent.Username & ".jpg"
    End Sub

    'Subroutine runs when the view profile button is clicked
    Private Sub btnViewProfile_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnViewProfile.Click
        'Sets the viewed profile to that of the logged-in student
        Viewing = 1

        'Opens the teacher account form
        frmStudentAccount.Show()
    End Sub


    'Subroutine runs when the Connect Four button is clicked
    Private Sub btnConnectFour_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnConnectFour.Click
        'Opens the Connect Four menu form
        frmConnect4Menu.Show()
        'Closes this form
        Me.Close()
    End Sub

    'Subroutine runs when the Noughts and Crosses button is clicked
    Private Sub btnNoughtsandCrosses_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnNoughtsandCrosses.Click
        'Opens the Noughts and Crosses menu form
        frmNoughtsandCrossesMenu.Show()
        'Closes this form
        Me.Close()
    End Sub

    'Subroutine runs when the Rock, Paper, Scissors button is clicked
    Private Sub btnRPS_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnRPS.Click
        'Opens the Rock, Paper, Scissors menu form
        frmRockPaperScissorsMenu.Show()
        'Closes this form
        Me.Close()
    End Sub

    'Subroutine runs when Connect Four button is moused over
    Private Sub btnConnectFour_MouseMove(ByVal sender As Object, ByVal e As
System.Windows.Forms.MouseEventArgs) Handles btnConnectFour.MouseMove
```

```
        'Changes the game description
        lblGameDesc.Text = "Strike from the skies with your mighty red or yellow
tokens, and slay the foul xenos with your glorious row of 4!" & vbCrLf & vbCrLf & "2
players"
        'Changes the game image
        picGameImg.Image = My.Resources.nac
    End Sub

    'Subroutine runs when Noughts and Crosses button is moused over
    Private Sub btnNoughtsandCrosses_MouseMove(ByVal sender As Object, ByVal e As
System.Windows.Forms.MouseEventArgs) Handles btnNoughtsandCrosses.MouseMove
        'Changes the game description
        lblGameDesc.Text = "On the barren fields of battle, strike the enemy where he
is most vulnerable by forming a line of three consecutive counters!" & vbCrLf & vbCrLf
& "2 players"
        'Changes the game image
        picGameImg.Image = My.Resources.ox
    End Sub

    'Subroutine runs when Rock Paper Scissors button is moused over
    Private Sub btnRPS_MouseMove(ByVal sender As Object, ByVal e As
System.Windows.Forms.MouseEventArgs) Handles btnRPS.MouseMove
        'Changes the game description
        lblGameDesc.Text = "It's a veritable battle royale out there, show no mercy as
you lead your chosen item of stationary or geological formation to victory and glory!"
& vbCrLf & vbCrLf & "2 players"
        'Changes the game image
        picGameImg.Image = My.Resources.rps
    End Sub

    'Subroutine runs when form is moused over
    Private Sub frmStudentHome_MouseMove(ByVal sender As Object, ByVal e As
System.Windows.Forms.MouseEventArgs) Handles Me.MouseMove
        'Changes the game description back to the default
        lblGameDesc.Text = "Hello and welcome to the official Bourne Grammar School
edutainment suite!" & vbCrLf & vbCrLf & "Pick a game or check out your account"
        'Changes the game image
        picGameImg.Image = My.Resources.edutained
    End Sub

    'Subroutine runs when game description label is moused over
    Private Sub lblGameDesc_MouseMove(ByVal sender As Object, ByVal e As
System.Windows.Forms.MouseEventArgs) Handles lblGameDesc.MouseMove
        'Changes the game description back to the default
        lblGameDesc.Text = "Hello and welcome to the official Bourne Grammar School
edutainment suite!" & vbCrLf & vbCrLf & "Pick a game or check out your account"
        'Changes the game image
        picGameImg.Image = My.Resources.edutained
    End Sub
End Class
```

frmStudentAccount.vb

```
Public Class frmStudentAccount

    'Subroutine runs when the form loads
    Private Sub frmAccount_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        'Runs the account section population subroutine
        AccountSection()
    End Sub
```

```
    'Subroutine runs when the form loads
    Sub AccountSection()

        'If the account to be viewed is the logged-in or searched-for student's...
        If Viewing = 1 Then
            'Places the logged-in student's name onto the form
            lblStudentName.Text = LoggedInStudent.Fname & " " & LoggedInStudent.Lname
            'Places the logged-in student's form onto the form
            lblForm.Text = LoggedInStudent.Form
            'Places the logged-in student's picture onto the form
            picStudent.ImageLocation = "G:\Computing Group\Y10 2009-
10\Ben's\Edutainment Suite\Connect4\bin\Debug\images\students\" &
LoggedInStudent.Username & ".jpg"
            'Places the logged-in student's wins onto the form
            lblWins.Text = "Wins: " & LoggedInStudent.Wins
            'Places the logged-in student's losses onto the form
            lblLosses.Text = "Losses: " & LoggedInStudent.Losses
            'Places the logged-in student's draws onto the form
            lblDraws.Text = "Draws: " & LoggedInStudent.Draws
            'However, if it is the opponent student's...
        Else
            'Places the opponent student's name onto the form
            lblStudentName.Text = OppStudent.Fname & " " & OppStudent.Lname
            'Places the opponent student's form onto the form
            lblForm.Text = OppStudent.Form
            'Places the opponent student's picture onto the form
            picStudent.ImageLocation = "G:\Computing Group\Y10 2009-
10\Ben's\Edutainment Suite\Connect4\bin\Debug\images\students\" & OppStudent.Username
& ".jpg"
            'Places the opponent student's wins onto the form
            lblWins.Text = "Wins: " & OppStudent.Wins
            'Places the opponent student's losses onto the form
            lblLosses.Text = "Losses: " & OppStudent.Losses
            'Places the opponent student's draws onto the form
            lblDraws.Text = "Draws: " & OppStudent.Draws
        End If
    End Sub

    'Subroutine runs when the view breakdown button is clicked
    Private Sub btnBreakdown_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnBreakdown.Click
        'Opens the student breakdown form
        frmStudentBreakdown.Show()
    End Sub

    'Subroutine runs when the view achievments button is clicked
    Private Sub btnAchievements_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnAchievements.Click
        'Opens the student achievments form
        frmStudentAchievements.Show()
    End Sub

End Class_
```

frmStudentBreakdown.vb

```
Public Class frmStudentBreakdown

    'Subroutine runs when the form loads
```

```
    Private Sub frmStudentBreakdown_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        'Declares the variable used to read the student breakdown text file
        Dim Reader As System.IO.StreamReader

        'If the account to be viewed is the logged-in or searched-for student's...
        If Viewing = 1 Then
            'Places the logged-in student's name onto the form
            lblBreakdown.Text = LoggedInStudent.Fname & " " & LoggedInStudent.Lname &
" Win Breakdown"

            'Gets the path to the logged-in student's breakdown text file
            Reader = My.Computer.FileSystem.OpenTextFileReader("G:\Computing Group\Y10
2009-10\Ben's\Edutainment Suite\Connect4\bin\Debug\WinsLosses\" &
LoggedInStudent.Username & ".txt")
            'However, if it is the opponent student's...
        Else
            'Places the opponent student's name onto the form
            lblBreakdown.Text = OppStudent.Fname & " " & OppStudent.Lname & " Win
Breakdown"

            'Gets the path to the opponent student's breakdown text file
            Reader = My.Computer.FileSystem.OpenTextFileReader("G:\Computing Group\Y10
2009-10\Ben's\Edutainment Suite\Connect4\bin\Debug\WinsLosses\" & OppStudent.Username
& ".txt")
        End If

        'Whilst the end of the text file hasn't been reached...
        While Not Reader.EndOfStream
            'Adds a line to the breakdown listbox
            lstBreakdown.Items.Add(Reader.ReadLine)
        End While

        'Closes the recordset
        Reader.Close()
    End Sub

End Class_
```

frmStudentAchievements.vb

```
Public Class frmStudentAchievements

    'Subroutine runs when the form loads
    Private Sub frmStudentAchievements_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        'Establishes the connection to the database
        OpenDB()

        'If the account to be viewed is the logged-in or searched-for student's...
        If Viewing = 1 Then
            'Places the logged-in student's name onto the form
            lblAchievements.Text = LoggedInStudent.Fname & " " & LoggedInStudent.Lname
& " Achievements"
            'However, if it is the opponent student's...
        Else
            'Places the opponent student's name onto the form
            lblAchievements.Text = OppStudent.Fname & " " & OppStudent.Lname & "
Achievements"
        End If
```

```
        'Runs the achievements population subroutine
        Achievements()
    End Sub

    'Subroutine runs when the form runs
    Sub Achievements()
        'Runs the achievment subroutines
        FirstBlood()
    End Sub

    'Subroutines run when the achievement subroutine calls them
    Sub FirstBlood()
        'Declares the variable used for reading data from the database
        Dim RSStaff As New ADODB.Recordset
        If RSStaff.State = ADODB.ObjectStateEnum.adStateOpen Then RSStaff.Close()

        'If the account to be viewed is the logged-in or searched-for student's...
        If Viewing = 1 Then
            'Builds SQL query to execute
            RSStaff.Open("SELECT * FROM `tblstudents` WHERE `username`='" &
LoggedInStudent.Username & "' AND `wins` >0", DBConn, ,
ADODB.LockTypeEnum.adLockOptimistic)
            'However, if it is the opponent student's...
        Else
            'Builds SQL query to execute
            RSStaff.Open("SELECT * FROM `tblstudents` WHERE `username`='" &
OppStudent.Username & "' AND `wins` >0", DBConn, ,
ADODB.LockTypeEnum.adLockOptimistic)
        End If

        'If any records are found...
        If RSStaff.RecordCount > 0 Then
            'Unlocks the achievement
            lblFirstBlood.Text = "First Blood"
            lblFirstBloodDeets.Text = "Win your first game"
            picFirstBlood.Image = My.Resources.firstblood
        End If

        'Closes the recordset
        RSStaff.Close()
    End Sub

End Class
```

frmConnect4Menu.vb

```
Public Class frmConnect4Menu

    'Subroutine runs when the hotseat game button is clicked
    Private Sub btnHotseat_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnHotseat.Click
        'Shows the Connect Four hotseat login form
        frmConnect4HotseatLogin.Show()
        'Closes this form
        Me.Close()
    End Sub

    'Subroutine runs when the network game button is clicked
    Private Sub btnNetwork_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnNetwork.Click
        'Shows the Connect Four network lobby form
```

```
        frmConnect4NetworkLobby.Show()
        'Closes this form
        Me.Close()
    End Sub

    'Subroutine runs when the back button is clicked
    Private Sub btnBack_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnBack.Click
        'Shows the student home form
        frmStudentHome.Show()
        'Closes this form
        Me.Close()
    End Sub

End Class
```

frmConnect4HotseatLogin.vb

```
Public Class frmConnect4HotseatLogin

    'Declares the variables used to log in
    Dim EnteredUsername, EnteredPassword As String

    'Subroutine runs when the form loads
    Private Sub frmConnect4HotseatLogin_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        'Runs the OpenDB subroutine
        OpenDB()
    End Sub

    'Subroutine runs when the okay button is clicked
    Private Sub btnOK_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnOK.Click
        'Runs the Login subroutine
        Login()
    End Sub

    'Subroutine runs when called in btnOK_Click subroutine
    Sub Login()
        'Sets the EnteredUsername and EnteredPassword variables to the entered
username and password
        EnteredUsername = txtUsername.Text
        EnteredPassword = txtPassword.Text

        If EnteredUsername <> LoggedInStudent.Username Then
            'Declares the Recordset used to view records in the database
            Dim RSStaff As New ADODB.Recordset
            If RSStaff.State = ADODB.ObjectStateEnum.adStateOpen Then RSStaff.Close()

            'Builds SQL query to execute
            RSStaff.Open("SELECT * FROM tblstudents WHERE Username='" &
EnteredUsername & "' AND Password='" & EnteredPassword & "'", DBConn, ,
ADODB.LockTypeEnum.adLockOptimistic)

            'If results are found...
            If RSStaff.RecordCount > 0 Then
                'Fills the various properties of the OppStudent object with their
respective values from the database
                OppStudent.Fname = RSStaff.Fields("Fname").Value
                OppStudent.Lname = RSStaff.Fields("Lname").Value
```

```
                OppStudent.Form = RSStaff.Fields("FormNum").Value &
RSStaff.Fields("FormLetter").Value
                OppStudent.Wins = RSStaff.Fields("Wins").Value
                OppStudent.Losses = RSStaff.Fields("Losses").Value
                OppStudent.Draws = RSStaff.Fields("Draws").Value
                OppStudent.Username = RSStaff.Fields("Username").Value
                OppStudent.StudentID = RSStaff.Fields("StudentID").Value

                'Opens the Connect Four subject selection form
                frmConnect4HotseatSubject.Show()
                'Closes this form
                Me.Close()
                'However if no results are found...
            Else
                'If the login details were invalid an error message will appear and
the username and password textboxes will be blanked out
                MsgBox("Invalid: Incorrect username or password.")
                txtUsername.Text = ""
                txtPassword.Text = ""
            End If
        Else
            'If the login details where the same as those of the currently logged-in
student, an error message will appear and the username and password textboxes will be
blanked out
            MsgBox("Invalid: That's you.")
            txtUsername.Text = ""
            txtPassword.Text = ""
        End If
    End Sub

    'Subroutine runs when the cancel button is clicked
    Private Sub btnCancel_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnCancel.Click
        'Opens the Connect Four menu form
        frmConnect4Menu.Show()
        'Closes this form
        Me.Close()
    End Sub

End Class
```

frmConnect4HotseatSubject.vb

```
Public Class frmConnect4HotseatSubject
    'Declares the variable used to store the chosen SubjectID
    Dim SubjectID As Integer

    'Subroutine runs on form load
    Private Sub frmConnect4HotseatSubject_Load(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles MyBase.Load
        'Runs the OpenDB subroutine
        OpenDB()

        'Runs the Populate subroutine
        Populate()
    End Sub

    'Subroutine runs then the okay button is clicked
    Private Sub btnOK_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnOK.Click
        'If everything has been selected...
```

```
        If (cmboDifficulty.SelectedItem <> "") And (cmboSubject.SelectedItem <> "")
And (cmboTopic.SelectedItem <> "") Then

            'Sets the chosen subject, difficulty and topic variables
            C4HSubject = SubjectID
            C4HDifficulty = cmboDifficulty.SelectedItem
            C4HTopic = cmboTopic.SelectedItem

            'Shows the Connect Four hotseat player selection form
            frmConnect4HotseatPlayerSelect.Show()
            'Closes this form
            Me.Close()
        Else
            'Otherwise display an error message
            MsgBox("Incorrect values")
        End If
    End Sub

    'Subroutine runs when called at form load
    Sub Populate()
        'Declares the Recordset used to view records in the database
        Dim RSStaff As New ADODB.Recordset
        If RSStaff.State = ADODB.ObjectStateEnum.adStateOpen Then RSStaff.Close()

        'Builds SQL query to execute
        RSStaff.Open("SELECT * FROM tblsubjects", DBConn, ,
ADODB.LockTypeEnum.adLockOptimistic)

        'Whilst not at the end of the data...
        While Not RSStaff.EOF
            'Add the subject to the combobox
            cmboSubject.Items.Add(RSStaff.Fields("Subject").Value)
            'Move on to the next record
            RSStaff.MoveNext()
        End While

        'Closes the recordset
        RSStaff.Close()
    End Sub

    'Subroutine runs when the selected item of the subject combobox is changed
    Private Sub cmboSubject_SelectedIndexChanged(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles cmboSubject.SelectedIndexChanged
        'Declares the StreamReader used to read the topic text file
        Dim Reader As System.IO.StreamReader

        'Enables the topic combobox and clears it of any data
        cmboTopic.Enabled = True
        cmboTopic.Items.Clear()

        'Sets the path to where the file is
        Reader = My.Computer.FileSystem.OpenTextFileReader("G:\Computing Group\Y10
2009-10\Ben's\Edutainment Suite\Connect4\bin\Debug\Topics\" & cmboSubject.SelectedItem
& "Topics.txt")
        'Whilst not at the end of the text file...
        While Not Reader.EndOfStream
            'Add the topic to the combobox
            cmboTopic.Items.Add(Reader.ReadLine)
        End While

        'Closes the streamreader
        Reader.Close()
```

```
        'Declares the Recordset used to view records in the database
        Dim RSStaff As New ADODB.Recordset
        If RSStaff.State = ADODB.ObjectStateEnum.adStateOpen Then RSStaff.Close()

        'Builds SQL query to execute
        RSStaff.Open("SELECT * FROM tblsubjects WHERE Subject='" &
cmboSubject.SelectedItem & "'", DBConn, , ADODB.LockTypeEnum.adLockOptimistic)
        'Sets SubjectID to that of the selected subject
        SubjectID = RSStaff.Fields("SubjectID").Value
        'Closes the recordset
        RSStaff.Close()
    End Sub

    'Subroutine runs when the cancel button is clicked
    Private Sub btnCancel_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnCancel.Click
        'Shows the Connect Four hotseat login form
        frmConnect4HotseatLogin.Show()
        'Closes this form
        Me.Close()
    End Sub

End Class
```

frmConnect4HotseatPlayerSelect.vb

```
Public Class frmConnect4HotseatPlayerSelect

    'Subroutine runs on form load
    Private Sub frmConnect4HotseatPlayerSelect_Load(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles MyBase.Load
        'Populates the player name labels with data
        lblLoggedInStudent.Text = LoggedInStudent.Fname & " " & LoggedInStudent.Lname
        lblOppStudent.Text = OppStudent.Fname & " " & OppStudent.Lname
    End Sub

    'Subroutines run when the counter selection buttons are clicked
    Private Sub btnOSred_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnOSred.Click
        'Sets the opponent student's colour to red
        OppStudent.C4Player = "Red"
        'Updates the appearance of the form
        picOSred.Image = My.Resources.red
        'Disables changing the colour and the logged-in student also picking red
        btnOSred.Enabled = False
        btnOSyellow.Enabled = False
        btnLISred.Enabled = False
        With picRed
            .Image = Nothing
            .Tag = "None"
        End With
        'Runs the CheckBoth subroutine
        CheckBoth()
    End Sub
    Private Sub btnOSyellow_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnOSyellow.Click
        OppStudent.C4Player = "Yellow"
        picOSyellow.Image = My.Resources.yellow
        btnOSyellow.Enabled = False
        btnOSred.Enabled = False
```

```
        btnLISyellow.Enabled = False
        With picYellow
            .Image = Nothing
            .Tag = "None"
        End With
        CheckBoth()
    End Sub
    Private Sub btnLISred_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnLISred.Click
        LoggedInStudent.C4Player = "Red"
        picLISred.Image = My.Resources.red
        btnLISred.Enabled = False
        btnLISyellow.Enabled = False
        btnOSred.Enabled = False
        With picRed
            .Image = Nothing
            .Tag = "None"
        End With
        CheckBoth()
    End Sub
    Private Sub btnLISyellow_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnLISyellow.Click
        LoggedInStudent.C4Player = "Yellow"
        picLISyellow.Image = My.Resources.yellow
        btnLISyellow.Enabled = False
        btnLISred.Enabled = False
        btnOSyellow.Enabled = False
        With picYellow
            .Image = Nothing
            .Tag = "None"
        End With
        CheckBoth()
    End Sub

    'Subroutine runs when called in the counter selection button click subroutines
    Sub CheckBoth()
        'If both players have chosen...
        If picYellow.Tag = "None" And picRed.Tag = "None" Then
            'Enables the button to leave the form
            btnContinue.Enabled = True
        End If
    End Sub

    'Subroutine runs when the continue button is clicked
    Private Sub btnContinue_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnContinue.Click
        'Shows the Connect Four hotseat form
        frmConnect4Hotseat.Show()
        'Closes this form
        Me.Close()
    End Sub

    'Subroutine runs when the cancel button is clicked
    Private Sub btnCancel_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnCancel.Click
        'Shows the Connect Four hotseat question selection form
        frmConnect4HotseatSubject.Show()
        'Closes this form
        Me.Close()
    End Sub

End Class
```

frmConnect4Hotseat.vb

```
Public Class frmConnect4Hotseat

    'Declares the variable used to determine the horizontal coordinate of the square
being used
    Dim x As Integer = 1
    'Declares the variable used to determine the vertical coordinate of the square
being used
    Dim y As Integer = 1
    'Declares the array containing the coordinates of all the pictureboxes that make
up the grid
    Dim Group(7, 6) As PictureBox
    'Declares the array containing the coordinates of the squares of the grid and what
condition they are currently in
    '0 = Empty
    '1 = Red
    '2=  Yellow
    '3 = Terminator
    Dim theGrid(7, 7) As Integer
    'Declares the array containing the number of the buttons for dropping counters
    Dim Buttons(7) As Button

    'Declares the class used for both players
    Public Class C4Play
        'Declares the variables used for storing the details of the Connect Four
player
        Public Colour, Username, Fname, lname As String
    End Class
    'Declares the class used for both players' scores
    Public Class Score
        'Declares the variable used for storing score of the player
        Public ScoreNum As Integer
        'This subroutine runs when a player wins a game
        Public Sub Increase()
            'Increases the score by 1
            ScoreNum = ScoreNum + 1
        End Sub
    End Class
    'Creates two objects of the Score class, one for each player
    Dim RedScore As New Score
    Dim YellowScore As New Score
    'Creates two objects of the C4Play class, one for each player
    Dim RedPlayer As New C4Play
    Dim YellowPlayer As New C4Play

    'Declares the variable used to store the primary key of the question record in the
database
    Dim QuestionID As Integer
    'Declares the y-coords for moving the current player label
    Dim StudentCurrLocationY As Integer = 35
    Dim OppStudentCurrLocationY As Integer = 140
    'Declares the variable used for detecting wins along the y-axis
    Dim Why As Integer
    'Declares the variable used to keep track of how many reds in a row there are
    Dim RedAddUp As Integer = 0
    'Declares the variable used to keep track of how many yellow in a row there are
    Dim YellowAddUp As Integer = 0
    'Declares the variable used to determine if a question was answered correctly
    Dim QCorrect As Boolean = False
```

```
    'Declares the variables used to store whether there is a winner or the game is a
draw
    Dim Won As Boolean = False
    Dim Draw As Boolean = False

    'Subroutine runs on form load
    Private Sub frmConnect4Hotseat_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        'Runs the OpenDB subroutine
        OpenDB()

        'If the logged-in student is red...
        If LoggedInStudent.C4Player = "Red" Then
            'Populate the RedPlayer object with the logged-in student's details
            With RedPlayer
                .Colour = "Red"
                .Username = LoggedInStudent.Username
                .Fname = LoggedInStudent.Fname
                .lname = LoggedInStudent.Lname
            End With

            'Populate the YellowPlayer object with the opponent student's details
            With YellowPlayer
                .Colour = "Yellow"
                .Username = OppStudent.Username
                .Fname = OppStudent.Fname
                .lname = OppStudent.Lname
            End With
            'But if the logged-in student is yellow...
        Else
            'Populate the RedPlayer object with the logged-in student's details
            With RedPlayer
                .Colour = "Red"
                .Username = OppStudent.Username
                .Fname = OppStudent.Fname
                .lname = OppStudent.Lname
            End With

            'Populate the YellowPlayer object with the opponent student's details
            With YellowPlayer
                .Colour = "Yellow"
                .Username = LoggedInStudent.Username
                .Fname = LoggedInStudent.Fname
                .lname = LoggedInStudent.Lname
            End With
        End If

        'Runs the AccountSection subroutine
        AccountSection()
        'Runs the MakeGrid subroutine
        MakeGrid()
        'Runs the Terminators subroutine
        Terminators()

        'Declares the variables used to store the current horizontal and vertical
coordinates of the grid
        Dim GridHor, GridVer As Integer

        'For each column of the grid...
        For GridHor = 1 To 7
            'For each square in that column...
            For GridVer = 1 To 6
```

```
                'Set the value to empty
                theGrid(GridHor, GridVer) = 0
            Next
        Next

        'Sets the scores to the defaults
        RedScore.ScoreNum = 0
        YellowScore.ScoreNum = 0

        'Sets the current player, runs the ChangePlayer and CurrPlayer subroutines
        C4Player = "Yellow"
        ChangePlayer()
        CurrPlayer()
    End Sub

    'Subroutine runs when called in DetectWinner subroutine
    Sub DatabaseDetails()
        'Declares the Recordset used to add new records to the database
        Dim RSNewRec As New ADODB.Recordset
        If RSNewRec.State = ADODB.ObjectStateEnum.adStateOpen Then RSNewRec.Close()
        'Declares the StreamWriter used to write to the game breakdown text files
        Dim writer As System.IO.StreamWriter

        'If the red player is the winner...
        If RedScore.ScoreNum = 1 Then
            'Opens a recordset
            RSNewRec.Open("SELECT * FROM tblstudents WHERE Username='" &
RedPlayer.Username & "'", DBConn, , ADODB.LockTypeEnum.adLockOptimistic)
            'Adds a win to the winning player's record
            With RSNewRec
                .Update()
                .Fields("Wins").Value = .Fields("Wins").Value + 1
                RSNewRec.Update()
            End With
            'Closes the recordset
            RSNewRec.Close()

            'Opens a recordset
            RSNewRec.Open("SELECT * FROM tblstudents WHERE Username='" &
YellowPlayer.Username & "'", DBConn, , ADODB.LockTypeEnum.adLockOptimistic)
            'Adds a loss to the losing player's record
            With RSNewRec
                .Update()
                .Fields("Losses").Value = .Fields("Losses").Value + 1
                RSNewRec.Update()
            End With
            'Closes the recordset
            RSNewRec.Close()

            'Sets the path to where the log shall be generated, and the filename
            writer = My.Computer.FileSystem.OpenTextFileWriter("G:\Computing Group\Y10
2009-10\Ben's\Edutainment Suite\Connect4\bin\Debug\WinsLosses\" & RedPlayer.Username &
".txt", True)

            'Writes the relevant data to the log
            writer.WriteLine(RedPlayer.Fname & " " & RedPlayer.lname & " beat " &
YellowPlayer.Fname & " " & YellowPlayer.lname & " in Connect Four - " & TimeOfDay & "
" & DateValue(Now))
            'Saves the log file
            writer.Close()

            'Sets the path to where the log shall be generated, and the filename
```

```
            writer = My.Computer.FileSystem.OpenTextFileWriter("G:\Computing Group\Y10
2009-10\Ben's\Edutainment Suite\Connect4\bin\Debug\WinsLosses\" &
YellowPlayer.Username & ".txt", True)

            'Writes the relevant data to the log
            writer.WriteLine(YellowPlayer.Fname & " " & YellowPlayer.lname & " was
beaten by " & RedPlayer.Fname & " " & RedPlayer.lname & " in Connect Four - " &
TimeOfDay & " " & DateValue(Now))
            'Saves the log file
            writer.Close()

            'Resets the red score back to its default
            RedScore.ScoreNum = 0
        End If

        'However, if the yellow player is the winner...
        If YellowScore.ScoreNum = 1 Then
            'Opens a recordset
            RSNewRec.Open("SELECT * FROM tblstudents WHERE Username='" &
YellowPlayer.Username & "'", DBConn, , ADODB.LockTypeEnum.adLockOptimistic)
            'Adds a win to the winning player's record
            With RSNewRec
                .Update()
                .Fields("Wins").Value = .Fields("Wins").Value + 1
                RSNewRec.Update()
            End With
            'Closes the recordset
            RSNewRec.Close()

            'Opens a recordset
            RSNewRec.Open("SELECT * FROM tblstudents WHERE Username='" &
RedPlayer.Username & "'", DBConn, , ADODB.LockTypeEnum.adLockOptimistic)
            'Adds a loss to the losing player's record
            With RSNewRec
                .Update()
                .Fields("Losses").Value = .Fields("Losses").Value + 1
                RSNewRec.Update()
            End With
            'Closes the recordset
            RSNewRec.Close()

            'Sets the path to where the log shall be generated, and the filename
            writer = My.Computer.FileSystem.OpenTextFileWriter("G:\Computing Group\Y10
2009-10\Ben's\Edutainment Suite\Connect4\bin\Debug\WinsLosses\" &
YellowPlayer.Username & ".txt", True)

            'Writes the relevant data to the log
            writer.WriteLine(YellowPlayer.Fname & " " & YellowPlayer.lname & " beat "
& RedPlayer.Fname & " " & RedPlayer.lname & " in Connect Four - " & TimeOfDay & " " &
DateValue(Now))
            'Saves the log file
            writer.Close()

            'Sets the path to where the log shall be generated, and the filename
            writer = My.Computer.FileSystem.OpenTextFileWriter("G:\Computing Group\Y10
2009-10\Ben's\Edutainment Suite\Connect4\bin\Debug\WinsLosses\" & RedPlayer.Username &
".txt", True)

            'Writes the relevant data to the log
            writer.WriteLine(RedPlayer.Fname & " " & RedPlayer.lname & " was beaten by
" & YellowPlayer.Fname & " " & YellowPlayer.lname & " in Connect Four - " & TimeOfDay
& " " & DateValue(Now))
```

```
            'Saves the log file
            writer.Close()

            'Resets the yellow score back to its default
            YellowScore.ScoreNum = 0
        End If

        'However, if the game is a draw...
        If Draw = True Then
            'Opens a recordset
            RSNewRec.Open("SELECT * FROM tblstudents WHERE Username='" &
RedPlayer.Username & "'", DBConn, , ADODB.LockTypeEnum.adLockOptimistic)
            'Adds a draw to the player's record
            With RSNewRec
                .Update()
                .Fields("Draws").Value = .Fields("Draws").Value + 1
                RSNewRec.Update()
            End With
            'Closes the recordset
            RSNewRec.Close()

            'Opens a recordset
            RSNewRec.Open("SELECT * FROM tblstudents WHERE Username='" &
YellowPlayer.Username & "'", DBConn, , ADODB.LockTypeEnum.adLockOptimistic)
            'Adds a draw to the other player's record
            With RSNewRec
                .Update()
                .Fields("Draws").Value = .Fields("Draws").Value + 1
                RSNewRec.Update()
            End With
            'Closes the recordset
            RSNewRec.Close()

            'Sets the path to where the log shall be generated, and the filename
            writer = My.Computer.FileSystem.OpenTextFileWriter("G:\Computing Group\Y10
2009-10\Ben's\Edutainment Suite\Connect4\bin\Debug\WinsLosses\" & RedPlayer.Username &
".txt", True)

            'Writes the relevant data to the log
            writer.WriteLine(RedPlayer.Fname & " " & RedPlayer.lname & " drew with " &
YellowPlayer.Fname & " " & YellowPlayer.lname & " in Connect Four - " & TimeOfDay & "
" & DateValue(Now))
            'Saves the log file
            writer.Close()

            'Sets the path to where the log shall be generated, and the filename
            writer = My.Computer.FileSystem.OpenTextFileWriter("G:\Computing Group\Y10
2009-10\Ben's\Edutainment Suite\Connect4\bin\Debug\WinsLosses\" &
YellowPlayer.Username & ".txt", True)

            'Writes the relevant data to the log
            writer.WriteLine(YellowPlayer.Fname & " " & YellowPlayer.lname & " drew
with " & RedPlayer.Fname & " " & RedPlayer.lname & " in Connect Four - " & TimeOfDay &
" " & DateValue(Now))
            'Saves the log file
            writer.Close()
        End If
    End Sub

    'Subroutine runs when called in the ChangePlayer subroutine
    Sub CurrPlayer()
        'If the logged-in player is the same as the current player, which is red...
```

```
        If LoggedInStudent.C4Player = "Red" Then
            If C4Player = "Red" Then
                'Places the current player label pointing to the logged-in student
                lblCurrPlayer.Top = StudentCurrLocationY
                'However if the current player is yellow...
            Else
                'Places the current player label pointing to the opponent student
                lblCurrPlayer.Top = OppStudentCurrLocationY
            End If
            'However, if the opponent student is the same as the current player, which
is yellow...
        Else
            If C4Player = "Yellow" Then
                'Places the current player label pointing to the logged-in student
                lblCurrPlayer.Top = StudentCurrLocationY
                'However if the current player is red...
            Else
                'Places the current player label pointing to the opponent student
                lblCurrPlayer.Top = OppStudentCurrLocationY
            End If
        End If
    End Sub

    'Subroutine runs when called in form load subroutine
    Sub AccountSection()
        'Populates the player name labels with data and get a picture of each player
        lblStudentName.Text = LoggedInStudent.Fname & " " & LoggedInStudent.Lname
        picStudent.ImageLocation = "G:\Computing Group\Y10 2009-10\Ben's\Edutainment
Suite\Connect4\bin\Debug\images\students\" & LoggedInStudent.Username & ".jpg"

        lblOppStudentName.Text = OppStudent.Fname & " " & OppStudent.Lname
        picOppStudentPic.ImageLocation = "G:\Computing Group\Y10 2009-
10\Ben's\Edutainment Suite\Connect4\bin\Debug\images\students\" & OppStudent.Username
& ".jpg"
    End Sub

    'Subroutine runs when called in form load subroutine
    Sub MakeGrid()
        'Ties the Buttons array with the button controls for dropping counters
        Buttons(1) = btn1
        Buttons(2) = btn2
        Buttons(3) = btn3
        Buttons(4) = btn4
        Buttons(5) = btn5
        Buttons(6) = btn6
        Buttons(7) = btn7

        'Ties the Group coordinates with the pictureboxes on the form
        Group(1, 1) = pb1dash1
        Group(1, 2) = pb1dash2
        Group(1, 3) = pb1dash3
        Group(1, 4) = pb1dash4
        Group(1, 5) = pb1dash5
        Group(1, 6) = pb1dash6

        Group(2, 1) = pb2dash1
        Group(2, 2) = pb2dash2
        Group(2, 3) = pb2dash3
        Group(2, 4) = pb2dash4
        Group(2, 5) = pb2dash5
        Group(2, 6) = pb2dash6
```

```
        Group(3, 1) = pb3dash1
        Group(3, 2) = pb3dash2
        Group(3, 3) = pb3dash3
        Group(3, 4) = pb3dash4
        Group(3, 5) = pb3dash5
        Group(3, 6) = pb3dash6

        Group(4, 1) = pb4dash1
        Group(4, 2) = pb4dash2
        Group(4, 3) = pb4dash3
        Group(4, 4) = pb4dash4
        Group(4, 5) = pb4dash5
        Group(4, 6) = pb4dash6

        Group(5, 1) = pb5dash1
        Group(5, 2) = pb5dash2
        Group(5, 3) = pb5dash3
        Group(5, 4) = pb5dash4
        Group(5, 5) = pb5dash5
        Group(5, 6) = pb5dash6

        Group(6, 1) = pb6dash1
        Group(6, 2) = pb6dash2
        Group(6, 3) = pb6dash3
        Group(6, 4) = pb6dash4
        Group(6, 5) = pb6dash5
        Group(6, 6) = pb6dash6

        Group(7, 1) = pb7dash1
        Group(7, 2) = pb7dash2
        Group(7, 3) = pb7dash3
        Group(7, 4) = pb7dash4
        Group(7, 5) = pb7dash5
        Group(7, 6) = pb7dash6
    End Sub

    'Subroutine runs when called in form load and btnReset_Click subroutines
    Sub Terminators()
        'Sets the extra seventh horizontal line of squares in theGrid to terminators
        While y <> 8
            theGrid(y, 7) = 3
            y = y + 1
        End While

        'Sets x back to 1 for use later
        y = 1
    End Sub

    'Subroutine runs when called in button click subroutines
    Sub CounterPlace()
        'Runs down the column to find the next blank space
        While theGrid(x, y) = 0
            y = y + 1
        End While

        'Changes the next blank space into the current player's marker and claims the
square for them
        If theGrid(x, 2) = 0 Then
            y = y - 1

            If C4Player = "Red" Then
                theGrid(x, y) = 1
```

```
                Group(x, y).Image = My.Resources.red
            Else
                theGrid(x, y) = 2
                Group(x, y).Image = My.Resources.yellow
            End If
        Else
            'If the only square left in the current row is the topmost one, the button
disables after being pressed to seal off the column
            y = y - 1
            If C4Player = "Red" Then
                theGrid(x, y) = 1
                Group(x, y).Image = My.Resources.red
            Else
                theGrid(x, y) = 2
                Group(x, y).Image = My.Resources.yellow
            End If
            Buttons(x).Enabled = False
        End If

        'Detects whether there is a winner
        DetectWinner()

        'If the game is still on, runs the ChangePlayer subroutine
        If (Won = False) And (Draw = False) Then
            ChangePlayer()
        End If
    End Sub

    'Subroutine runs when called in CounterPlace subroutine
    Sub DetectWinner()
        'Declares the variable used to determine number of spaces to the left of the
last-placed counter
        Dim SpacestoLeft As Integer
        'Declares the variable used  to determine number of spaces to the right of the
last-placed counter
        Dim SpacestoRight As Integer
        'Declares the variable used  to determine number of spaces above the last-
placed counter
        Dim SpacesAbove As Integer
        'Declares the variable used  to determine number of spaces below the last-
placed counter
        Dim SpacesBelow As Integer

        '///WIN CONDITIONS\\\

        '/HORIZONTAL\

        'Determines no. of spaces to left and right of last-placed coutner
        SpacestoLeft = x - 1
        SpacestoRight = 7 - x

        'Runs along the row of the last-placed counter to see if there are four red or
four yellows in a row
        For HorizSquare As Integer = x - SpacestoLeft To x + SpacestoRight
            If theGrid(HorizSquare, y) = 1 Then
                YellowAddUp = 0
                RedAddUp = RedAddUp + 1
                'If there are four reds in a row horizontally...
                If RedAddUp = 4 Then
                    'Gives the red player a win, pops up a messagebox, makes changes
to the database and sets up the game to be reset
                    Won = True
```

```
                    RedScore.Increase()
                    MsgBox("Red wins")
                    DatabaseDetails()
                    btn1.Enabled = False
                    btn2.Enabled = False
                    btn3.Enabled = False
                    btn4.Enabled = False
                    btn5.Enabled = False
                    btn6.Enabled = False
                    btn7.Enabled = False
                    lblCurrPlayer.Visible = False
                    btnReset.Visible = True
                End If
            End If
            If theGrid(HorizSquare, y) = 2 Then
                RedAddUp = 0
                YellowAddUp = YellowAddUp + 1
                'Of there are four yellows in a row horizontally...
                If YellowAddUp = 4 Then
                    'Gives the yellow player a win, pops up a messagebox, makes
changes to the database and sets up the game to be reset
                    Won = True
                    YellowScore.Increase()
                    MsgBox("Yellow wins")
                    DatabaseDetails()
                    btn1.Enabled = False
                    btn2.Enabled = False
                    btn3.Enabled = False
                    btn4.Enabled = False
                    btn5.Enabled = False
                    btn6.Enabled = False
                    btn7.Enabled = False
                    lblCurrPlayer.Visible = False
                    btnReset.Visible = True
                End If
            End If
            If theGrid(HorizSquare, y) = 0 Then
                RedAddUp = 0
                YellowAddUp = 0
            End If
        Next

        '/VERTICAL\

        'Determines no. of spaces above and below the last-placed coutner
        SpacesAbove = y - 1
        SpacesBelow = 7 - y

        'Runs down the column of the last-placed counter to see if there are four red
or four yellows in a row
        For VertSquare As Integer = y - SpacesAbove To y + SpacesBelow
            If theGrid(x, VertSquare) = 1 Then
                YellowAddUp = 0
                RedAddUp = RedAddUp + 1
                'If there are four reds in a row vertically...
                If RedAddUp = 4 Then
                    'Gives the red player a win, pops up a messagebox, makes changes
to the database and sets up the game to be reset
                    Won = True
                    RedScore.Increase()
                    MsgBox("Red wins")
                    DatabaseDetails()
```

```
                    btn1.Enabled = False
                    btn2.Enabled = False
                    btn3.Enabled = False
                    btn4.Enabled = False
                    btn5.Enabled = False
                    btn6.Enabled = False
                    btn7.Enabled = False
                    lblCurrPlayer.Visible = False
                    btnReset.Visible = True
                End If
            End If
            If theGrid(x, VertSquare) = 2 Then
                RedAddUp = 0
                YellowAddUp = YellowAddUp + 1
                'If there are four yellow in a row vertically...
                If RedAddUp = 4 Then
                    'Gives the yellow player a win, pops up a messagebox, makes
changes to the database and sets up the game to be reset
                    Won = True
                    YellowScore.Increase()
                    MsgBox("Yellow wins")
                    DatabaseDetails()
                    btn1.Enabled = False
                    btn2.Enabled = False
                    btn3.Enabled = False
                    btn4.Enabled = False
                    btn5.Enabled = False
                    btn6.Enabled = False
                    btn7.Enabled = False
                    lblCurrPlayer.Visible = False
                    btnReset.Visible = True
                End If
            End If
            If theGrid(x, VertSquare) = 0 Then
                RedAddUp = 0
                YellowAddUp = 0
            End If
        Next

        'If a diagonal win in possible with the position of the last-placed counter...
        If x < 5 Then

            '/DIAGONAL BOTTOM-UP\

            'Runs the Diag (Bottom-Up) Win Detection subroutine
            DiagBUWin()

        End If

        'If a diagonal win in possible with the position of the last-placed counter...
        If x > 3 Then

            '/DIAGONAL TOP-DOWN\

            'Runs the Diag (Top-Down) Win Detection subroutine
            DiagTDWin()

        End If

        'If there is no winner...
        If Won <> True Then
```

```
            'Declares the variable used to keep track of how many blank squares there
are left
            Dim Blanks As Integer = 42
            'Goes through the grid row-by-row, column-by-column, decrementing the
blanks value when a non-blank square in encountered
            For Why = 1 To 6
                For x = 1 To 7
                    If theGrid(x, Why) <> 0 Then
                        Blanks = Blanks - 1
                    End If
                Next
            Next
            'If there are no blanks left...
            If Blanks = 0 Then
                'Declares the game a draw, amends the database, sets the form up to
reset
                Draw = True
                MsgBox("No-one wins, it's a draw")
                DatabaseDetails()
                lblCurrPlayer.Visible = False
                btnReset.Visible = True
            End If
        End If
    End Sub

    'Subroutine runs when called in DetectWinner subroutine
    Sub DiagTDWin()
        'Declares the varables used to run through the diagonals
        Dim v, z, a, w As Integer

        'Sets the variables to their defaults
        x = 1
        v = x
        a = x
        z = x + 3
        y = 1

        'Runs through the diagonals to determine if there are four red or yellow
counters in a row
        For Why = 1 To 3
            For x = 1 To 4
                w = x
                y = Why
                If theGrid(x, y) = 1 Then
                    RedAddUp = 1
                    For v = a To z
                        x = x + 1
                        y = y + 1
                        If x < 8 Then
                            If theGrid(x, y) = 1 Then
                                RedAddUp = RedAddUp + 1
                            End If
                            'If there are four reds in a row diagonally...
                            If RedAddUp = 4 Then
                                'Gives the red player a win, pops up a messagebox,
makes changes to the database and sets up the game to be reset
                                Won = True
                                RedScore.Increase()
                                MsgBox("Red wins")
                                DatabaseDetails()
                                btn1.Enabled = False
                                btn2.Enabled = False
```

```
                                btn3.Enabled = False
                                btn4.Enabled = False
                                btn5.Enabled = False
                                btn6.Enabled = False
                                btn7.Enabled = False
                                lblCurrPlayer.Visible = False
                                btnReset.Visible = True
                                Exit Sub
                            End If
                        End If
                    Next
                End If

                If theGrid(w, y) = 2 Then
                    YellowAddUp = 1
                    For v = a To z
                        w = w + 1
                        y = y + 1
                        If w < 8 Then
                            If theGrid(w, y) = 1 Then
                                YellowAddUp = YellowAddUp + 1
                            End If
                            'If there are four yellow in a row diagonally...
                            If YellowAddUp = 4 Then
                                'Gives the yellow player a win, pops up a messagebox,
makes changes to the database and sets up the game to be reset
                                Won = True
                                YellowScore.Increase()
                                MsgBox("Yellow wins")
                                DatabaseDetails()
                                btn1.Enabled = False
                                btn2.Enabled = False
                                btn3.Enabled = False
                                btn4.Enabled = False
                                btn5.Enabled = False
                                btn6.Enabled = False
                                btn7.Enabled = False
                                lblCurrPlayer.Visible = False
                                btnReset.Visible = True
                                Exit Sub
                            End If
                        End If
                    Next
                End If
            Next
        Next

        'Resets the values to their defaults
        RedAddUp = 0
        YellowAddUp = 0
    End Sub

    'Subroutine runs when called in DetectWinner subroutine
    Sub DiagBUWin()
        'Declares the varables used to run through the diagonals
        Dim v, z, a, w As Integer

        'Sets the variables to their defaults
        x = 1
        v = x
        a = x
        z = x + 3
```

```
        y = 6

        'Runs through the diagonals to determine if there are four red or yellow
counters in a row
        For Why = 4 To 6
            For x = 1 To 4
                w = x
                y = Why
                If theGrid(x, y) = 1 Then
                    RedAddUp = 1
                    For v = a To z
                        x = x + 1
                        y = y - 1
                        If x < 8 Then
                            If theGrid(x, y) = 1 Then
                                RedAddUp = RedAddUp + 1
                            End If
                            'If there are four reds in a row diagonally...
                            If RedAddUp = 4 Then
                                'Gives the red player a win, pops up a messagebox,
makes changes to the database and sets up the game to be reset
                                Won = True
                                RedScore.Increase()
                                MsgBox("Red wins")
                                DatabaseDetails()
                                btn1.Enabled = False
                                btn2.Enabled = False
                                btn3.Enabled = False
                                btn4.Enabled = False
                                btn5.Enabled = False
                                btn6.Enabled = False
                                btn7.Enabled = False
                                lblCurrPlayer.Visible = False
                                btnReset.Visible = True
                                Exit Sub
                            End If
                        End If
                    Next
                End If

                If theGrid(w, y) = 2 Then
                    YellowAddUp = 1
                    For v = a To z
                        w = w + 1
                        y = y - 1
                        If w < 8 Then
                            If theGrid(w, y) = 1 Then
                                YellowAddUp = YellowAddUp + 1
                            End If
                            'If there are four yellows in a row diagonally...
                            If YellowAddUp = 4 Then
                                'Gives the yellow player a win, pops up a messagebox,
makes changes to the database and sets up the game to be reset
                                Won = True
                                YellowScore.Increase()
                                MsgBox("Yellow wins")
                                DatabaseDetails()
                                btn1.Enabled = False
                                btn2.Enabled = False
                                btn3.Enabled = False
                                btn4.Enabled = False
                                btn5.Enabled = False
```

```
                                btn6.Enabled = False
                                btn7.Enabled = False
                                lblCurrPlayer.Visible = False
                                btnReset.Visible = True
                                Exit Sub
                            End If
                        End If
                    Next
                End If
            Next
        Next

        'Resets the values to their defaults
        RedAddUp = 0
        YellowAddUp = 0
    End Sub

    'Subroutine runs when reset button is clicked
    Private Sub btnReset_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnReset.Click
        'Resets the values to their defaults
        RedAddUp = 0
        YellowAddUp = 0

        'Resets the grid to its original blank state
        For y = 1 To 7
            For x = 1 To 7
                theGrid(x, y) = 0
                If y < 7 Then
                    Group(x, y).Image = Nothing
                End If
            Next
        Next

        'Resets the game state variables
        Won = False
        Draw = False

        'Resets the x and y coords of the grid
        x = 1
        y = 1

        'Runs the Terminators subroutine
        Terminators()

        'Resets the current player
        C4Player = "Red"
        CurrPlayer()

        'Disables the counter drop buttons
        btn1.Visible = False
        btn2.Visible = False
        btn3.Visible = False
        btn4.Visible = False
        btn5.Visible = False
        btn6.Visible = False
        btn7.Visible = False
        'Enables them for later
        btn1.Enabled = True
        btn2.Enabled = True
        btn3.Enabled = True
        btn4.Enabled = True
```

```
        btn5.Enabled = True
        btn6.Enabled = True
        btn7.Enabled = True

        'Runs the Question subroutine
        Question()

        'Hides the reset button
        lblCurrPlayer.Visible = True
        btnReset.Visible = False
    End Sub

    'Subroutines run when their respective buttons are clicked
    Private Sub btn1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles btn1.Click
        'Sets x to the x-coord of the selected column
        x = 1
        'Runs the CounterPlace subroutine
        CounterPlace()
    End Sub
    Private Sub btn2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles btn2.Click
        x = 2
        CounterPlace()
    End Sub
    Private Sub btn3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles btn3.Click
        x = 3
        CounterPlace()
    End Sub
    Private Sub btn4_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles btn4.Click
        x = 4
        CounterPlace()
    End Sub
    Private Sub btn5_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles btn5.Click
        x = 5
        CounterPlace()
    End Sub
    Private Sub btn6_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles btn6.Click
        x = 6
        CounterPlace()
    End Sub
    Private Sub btn7_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles btn7.Click
        x = 7
        CounterPlace()
    End Sub

    'Subroutine runs when back button is clicked
    Private Sub btnBack_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnBack.Click
        'Shows the Connect Four menu
        frmConnect4Menu.Show()
        'Closes this form
        Me.Close()
    End Sub

    'Subroutine runs when called in ChangePlayer and btnReset_Click subroutines
    Sub Question()
```

```
        'Declares the Recordset used to view records in the database
        Dim RSStaff As New ADODB.Recordset
        If RSStaff.State = ADODB.ObjectStateEnum.adStateOpen Then RSStaff.Close()

        'Builds SQL query to execute
        RSStaff.Open("SELECT * FROM tblquestions WHERE SubjectID='" & C4HSubject & "'
AND Difficulty='" & C4HDifficulty & "' AND Topic='" & C4HTopic & "' ORDER BY RAND()
LIMIT 1", DBConn, , ADODB.LockTypeEnum.adLockOptimistic)

        'Sets the QuestionID variable to that of the selected question
        QuestionID = RSStaff.Fields("QuestionID").Value

        'Makes the question controls visible
        grpQuestion.Visible = True
        lblQuestion.Visible = True
        txtAnswer.Visible = True
        btnSubmit.Visible = True

        'Displays the selected question
        lblQuestion.Text = RSStaff.Fields("Question").Value
        'Closes the recordset
        RSStaff.Close()
    End Sub

    'Subroutine runs when called in form load and CounterPlace subroutines
    Sub ChangePlayer()
        'Sets x & y back to 1 for use later
        x = 1
        y = 1

        'Changes the current player
        If C4Player = "Red" Then
            C4Player = "Yellow"
            CurrPlayer()
        Else
            C4Player = "Red"
            CurrPlayer()
        End If

        'Hides the counter placing buttons
        btn1.Visible = False
        btn2.Visible = False
        btn3.Visible = False
        btn4.Visible = False
        btn5.Visible = False
        btn6.Visible = False
        btn7.Visible = False

        'Runs the Question subroutine
        Question()
    End Sub

    'Subroutine runs then answer submit button is clicked
    Private Sub btnSubmit_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnSubmit.Click
        'Declares the Recordset used to view records in the database
        Dim RSStaff As New ADODB.Recordset
        If RSStaff.State = ADODB.ObjectStateEnum.adStateOpen Then RSStaff.Close()

        'Hides the question controls
        grpQuestion.Visible = False
        lblQuestion.Visible = False
```

```
        txtAnswer.Visible = False
        btnSubmit.Visible = False

        'Builds SQL query to execute
        RSStaff.Open("SELECT * FROM tblquestions WHERE QuestionID='" & QuestionID &
"'", DBConn, , ADODB.LockTypeEnum.adLockOptimistic)

        'If the answer is correct...
        If txtAnswer.Text = RSStaff.Fields("Answer").Value Then
            'Display a message box
            MsgBox("Correct!")

            'Sets the question correct flag to true
            QCorrect = True

            'Makes the counter placement buttons visible
            btn1.Visible = True
            btn2.Visible = True
            btn3.Visible = True
            btn4.Visible = True
            btn5.Visible = True
            btn6.Visible = True
            btn7.Visible = True

            'Runs the QDatabase subroutine
            QDatabase()
            'However if the answer if incorrect...
        Else
            'Display a message box
            MsgBox("Incorrect!")

            'Sets the question correct flag to false
            QCorrect = False

            'Runs the QDatabase subroutine
            QDatabase()
            'Runs the ChangePlayer subroutine
            ChangePlayer()
        End If

        'Blanks the answer textbox for the next question
        txtAnswer.Text = ""
    End Sub

    'Subroutine runs when called in btnSubmit_Click subroutine
    Sub QDatabase()
        'Declares the Recordset used to add new records to the database
        Dim RSNewRec As New ADODB.Recordset
        If RSNewRec.State = ADODB.ObjectStateEnum.adStateOpen Then RSNewRec.Close()
        'Opens a recordset
        RSNewRec.Open("SELECT * FROM tblattempted", DBConn, ,
ADODB.LockTypeEnum.adLockOptimistic)
        'Adds the new record
        With RSNewRec
            .AddNew()
            .Fields("SubjectID").Value = C4HSubject
            .Fields("QuestionID").Value = QuestionID
            If C4Player = "Red" Then
                If RedPlayer.Username = LoggedInStudent.Username Then
                    .Fields("StudentID").Value = LoggedInStudent.StudentID
                Else
                    .Fields("StudentID").Value = OppStudent.StudentID
```

```
                End If
            Else
                If YellowPlayer.Username = LoggedInStudent.Username Then
                    .Fields("StudentID").Value = LoggedInStudent.StudentID
                Else
                    .Fields("StudentID").Value = OppStudent.StudentID
                End If
            End If
            .Fields("When").Value = TimeOfDay & " " & DateValue(Now)
            If QCorrect = True Then
                .Fields("Correct").Value = 1
            Else
                .Fields("Correct").Value = 0
            End If

            'Resets the question correct flag
            QCorrect = False

            'Updates the recordset
            RSNewRec.Update()
            'Closes the recordset
            RSNewRec.Close()
        End With
    End Sub

    'Subroutine runs when view logged-in student's profile button is clicked
    Private Sub btnViewProfile_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnViewProfile.Click
        'Sets viewed profile to logged-in student's
        Viewing = 1
        'Shows the student account form
        frmStudentAccount.Show()
    End Sub

    'Subroutine runs when view opponent student's profile button is clicked
    Private Sub btnOppViewProfile_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnOppViewProfile.Click
        'Sets viewed profile to opponent student's
        Viewing = 2
        'Shows the student account form
        frmStudentAccount.Show()
    End Sub

End Class
```

frmConnect4NetworkLobby.vb

```
Public Class frmConnect4NetworkLobby
    'Declares the variable used to determine if a game has a second player or not
    Dim NoOpp As Boolean = True
    'Declares the variable used to determine if the logged-in player is the host of a
game
    Dim Host As Boolean = True

    'Subroutine runs on form load
    Private Sub frmConnect4NetworkLobby_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        'Runs the OpenDB subroutine
        OpenDB()

        'Runs the UpdateList subroutine
```

```
        UpdateList()

        'Enables the update timer
        tmrUpdate.Enabled = True
    End Sub

    'Subroutine runs when called on form load or btnRefresh_Click subroutines
    Sub UpdateList()
        'Clears the lobby listbox of data
        lstLobby.Items.Clear()

        'Declares the variable used to determine if the logged-in player is currently
playing a given game
        Dim CurrPlaying As String = ""
        'Delcares the Recordset used to view records in the database
        Dim RSStaff As New ADODB.Recordset
        If RSStaff.State = ADODB.ObjectStateEnum.adStateOpen Then RSStaff.Close()

        'Builds SQL query to execute
        RSStaff.Open("SELECT * FROM tblconnect4 WHERE OppUsername ='' OR OppUsername
='" & LoggedInStudent.Username & "' OR HostUsername ='" & LoggedInStudent.Username &
"'", DBConn, , ADODB.LockTypeEnum.adLockOptimistic)

        'If records are found...
        If RSStaff.RecordCount > 0 Then
            'Whilst not at the end of the records...
            While Not RSStaff.EOF
                'If there is an opponent in the game...
                If RSStaff.Fields("OppUsername").Value <> "" Then
                    'Sets CurrPlaying to positive
                    CurrPlaying = " - CURRENTLY PLAYING"
                Else
                    'Otherwise CurrPlaying is negative
                    CurrPlaying = ""
                End If
                'Add the game details to the lobby listbox
                lstLobby.Items.Add(RSStaff.Fields("GameID").Value & " - " &
RSStaff.Fields("GameName").Value & " - " & RSStaff.Fields("HostUsername").Value &
CurrPlaying)
                'Move on to the next record
                RSStaff.MoveNext()
            End While
        Else
            'Otherwise display an error message
            MsgBox("No games found")
        End If

        'Closes the recordset
        RSStaff.Close()
    End Sub

    'Subroutine runs when the host game button is clicked
    Private Sub btnHostGame_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnHostGame.Click
        'Shows the Connect Four network hosting form
        frmConnect4NetworkHostGame.Show()
    End Sub

    'Subroutine runs when the your games button is clicked
    Private Sub btnYourGames_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnYourGames.Click
        'Shows the Connect Four network your games form
```

```
        frmConnect4NetworkYourGames.Show()
    End Sub

    'Subroutine runs when the selected item of the lobby listbox is changes
    Private Sub lstLobby_SelectedIndexChanged(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles lstLobby.SelectedIndexChanged
        'Determine the GameID of the selected game
        Dim EndofGamID As Integer = InStr(1, lstLobby.SelectedItem, " ",
CompareMethod.Text)
        Dim GamID As String = Mid(lstLobby.SelectedItem, 1, EndofGamID)

        'Declares the variables used to determine the usernames of both players in the
selected game
        Dim HUsername As String
        Dim Username As String

        'Declares the Recordset used to view records in the database
        Dim RSStaff As New ADODB.Recordset
        If RSStaff.State = ADODB.ObjectStateEnum.adStateOpen Then RSStaff.Close()

        Try
            'Builds SQL query to execute
            RSStaff.Open("SELECT * FROM tblConnect4 WHERE GameID='" & GamID & "'",
DBConn, , ADODB.LockTypeEnum.adLockOptimistic)

            'Sets the selected GameID
            GameID = GamID

            'If the logged-in user isn't the host of the selected game...
            If RSStaff.Fields("HostUsername").Value <> LoggedInStudent.Username Then
                'Sets the host flag to false
                Host = False

                'Sets the players
                LoggedInStudent.C4Player = "Yellow"
                OppStudent.C4Player = "Red"

                'Sets the opponent as the host
                Username = RSStaff.Fields("HostUsername").Value

                'Closes the recordset
                RSStaff.Close()

                'Builds SQL query to execute
                RSStaff.Open("SELECT * FROM tblstudents WHERE username='" & Username &
"'", DBConn, , ADODB.LockTypeEnum.adLockOptimistic)

                'Populates the properties of the OppStudent object with the host's
data
                With OppStudent
                    .Fname = RSStaff.Fields("fname").Value
                    .Lname = RSStaff.Fields("lname").Value
                    .StudentID = RSStaff.Fields("StudentID").Value
                    .Username = RSStaff.Fields("username").Value
                End With

                'Sets the no opponent flag to false
                NoOpp = False
            Else
                'If the logged-in user is the host, sets the host flag to true
                Host = True
                'Sets the players
```

```
                LoggedInStudent.C4Player = "Red"
                OppStudent.C4Player = "Yellow"

                'Sets the usernames of the players
                HUsername = RSStaff.Fields("HostUsername").Value
                Username = RSStaff.Fields("OppUsername").Value

                'Closes the recordset
                RSStaff.Close()

                'Builds SQL query to execute
                RSStaff.Open("SELECT * FROM tblstudents WHERE username='" & Username &
"'", DBConn, , ADODB.LockTypeEnum.adLockOptimistic)

                'If records are found...
                If RSStaff.RecordCount > 0 Then
                    'Populates the properties of the OppStudent object with data
                    With OppStudent
                        .Fname = RSStaff.Fields("fname").Value
                        .Lname = RSStaff.Fields("lname").Value
                        .StudentID = RSStaff.Fields("StudentID").Value
                        .Username = RSStaff.Fields("username").Value
                    End With
                    'Sets the no opponent flag to false
                    NoOpp = False
                Else
                    'Sets the no opponent flag to true
                    NoOpp = True
                End If
            End If

            'Closes the recordset
            RSStaff.Close()
        Catch
        End Try
    End Sub

    'Subroutine runs when join game button is clicked
    Private Sub btnJoinGame_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnJoinGame.Click
        'Declares the Recordset used to add new records to the database
        Dim RSNewRec As New ADODB.Recordset
        If RSNewRec.State = ADODB.ObjectStateEnum.adStateOpen Then RSNewRec.Close()

        'If there is an opponent in the game...
        If NoOpp = False Then
            'Builds the SQL query to execute
            RSNewRec.Open("SELECT * FROM tblConnect4 WHERE GameID='" & GameID & "'",
DBConn, , ADODB.LockTypeEnum.adLockOptimistic)
            'If the logged-in user isn't the host...
            If Host = False Then
                'Sets the opponent username to the logged-in student's
                RSNewRec.Fields("OppUsername").Value = LoggedInStudent.Username
                RSNewRec.Update()
            End If
            'Closes the recordset
            RSNewRec.Close()

            'Shows the Connect Four network game form
            frmConnect4Network.Show()
            'Closes this form
            Me.Close()
```

```
        Else
            'Displays an error message
            MsgBox("No opponent in game")
        End If
    End Sub

    'Subroutine runs when the refresh button is clicked
    Private Sub btnRefresh_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnRefresh.Click
        'Runs the UpdateList subroutine
        UpdateList()
    End Sub

    'Subroutine runs when the timer ticks
    Private Sub tmrUpdate_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles tmrUpdate.Tick
        'Updates the lobby list
        UpdateList()
    End Sub

End Class
```

frmConnect4NetworkHostGame.vb

```
Public Class frmConnect4NetworkHostGame
    'Declares the variable used to store the chosen SubjectID
    Dim SubjectID As Integer

    'Subroutine runs on form load
    Private Sub frmConnect4NetworkHostGame_Load(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles MyBase.Load
        'Runs the OpenDB subroutine
        OpenDB()

        'Runs the Populate subroutine
        Populate()
    End Sub

    'Subroutine runs then the host game button is clicked
    Private Sub btnHost_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnHost.Click
        'Declares the variable used for getting the result of the message box
        Dim Result As MsgBoxResult
        'Declares the variable used for adding a new record into the database
        Dim RSNewRec As New ADODB.Recordset
        If RSNewRec.State = ADODB.ObjectStateEnum.adStateOpen Then RSNewRec.Close()
        'Declares the variable used for detecting invalid data entry
        Dim Errors As String = ""

        'Assembles an error report if any invalid data entry detected
        If txtGameName.Text = "" Then
            Errors = Errors & "No game name input" & vbCrLf
        End If
        If cmboSubject.SelectedItem = "" Then
            Errors = Errors & "No subject selected" & vbCrLf
        End If
        If cmboDifficulty.SelectedItem = "" Then
            Errors = Errors & "No difficulty selected" & vbCrLf
        End If
        If cmboTopic.SelectedItem = "" Then
            Errors = Errors & "No topic selected" & vbCrLf
```

```
        End If

        'If no invalid data entry is detected...
        If Errors = "" Then
            'Displays a validation message box before saving the data to the database
            Result = MsgBox("Are you sure all these details are correct? Remember,
spelling is vital." & vbCrLf & vbCrLf & "Details:" & vbCrLf & txtGameName.Text &
vbCrLf & cmboSubject.SelectedItem & " (" & cmboDifficulty.SelectedItem & ")" & vbCrLf
& cmboTopic.SelectedItem, MsgBoxStyle.YesNo)
            'If the data is approved by the user...
            If Result = MsgBoxResult.Yes Then
                'Opens a recordset
                RSNewRec.Open("SELECT * FROM tblconnect4", DBConn, ,
ADODB.LockTypeEnum.adLockOptimistic)
                'Populates the record with the selected data
                With RSNewRec
                    .AddNew()
                    .Fields("GameName").Value = txtGameName.Text
                    .Fields("HostUsername").Value = LoggedInStudent.Username
                    .Fields("CurrentPlayer").Value = "Red"
                    .Fields("C4NSubject").Value = SubjectID
                    .Fields("C4NDifficulty").Value = cmboDifficulty.SelectedItem
                    .Fields("C4NTopic").Value = cmboTopic.SelectedItem
                    RSNewRec.Update()
                    'Closes the recordset
                    RSNewRec.Close()
                End With

                'Closes this form
                Me.Close()
            End If
            'If any invalid data entry is detected...
        Else
            'Displays a message box with any detected invalid data entry
            MsgBox("Invalid input:" & vbCrLf & vbCrLf & Errors)
        End If

    End Sub

    'Subroutine runs when called in form load subroutine
    Sub Populate()
        ''Declares the Recordset used to view records in the database
        Dim RSStaff As New ADODB.Recordset
        If RSStaff.State = ADODB.ObjectStateEnum.adStateOpen Then RSStaff.Close()

        'Builds SQL query to execute
        RSStaff.Open("SELECT * FROM tblsubjects", DBConn, ,
ADODB.LockTypeEnum.adLockOptimistic)

        'Whilst not at the end of the records...
        While Not RSStaff.EOF
            'Add the subject to the subject combobox
            cmboSubject.Items.Add(RSStaff.Fields("Subject").Value)
            'Move on to the next record
            RSStaff.MoveNext()
        End While

        'Closes the recordset
        RSStaff.Close()
    End Sub

    'Subroutine runs when the selected item in the subject combobox is changed
```

```
    Private Sub cmboSubject_SelectedIndexChanged(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles cmboSubject.SelectedIndexChanged
        'Declares the StreamReader used to read the topic text file
        Dim Reader As System.IO.StreamReader

        'Enables the topic combobox and clears it of any data
        cmboTopic.Enabled = True
        cmboTopic.Items.Clear()

        'Sets the path to where the file is
        Reader = My.Computer.FileSystem.OpenTextFileReader("G:\Computing Group\Y10
2009-10\Ben's\Edutainment Suite\Connect4\bin\Debug\Topics\" & cmboSubject.SelectedItem
& "Topics.txt")
        'Whilst not at the end of the text file...
        While Not Reader.EndOfStream
            'Add the topic to the combobox
            cmboTopic.Items.Add(Reader.ReadLine)
        End While

        'Closes the streamreader
        Reader.Close()

        'Declares the Recordset used to view records in the database
        Dim RSStaff As New ADODB.Recordset
        If RSStaff.State = ADODB.ObjectStateEnum.adStateOpen Then RSStaff.Close()

        'Builds SQL query to execute
        RSStaff.Open("SELECT * FROM tblsubjects WHERE Subject='" &
cmboSubject.SelectedItem & "'", DBConn, , ADODB.LockTypeEnum.adLockOptimistic)
        'Sets SubjectID to that of the selected subject
        SubjectID = RSStaff.Fields("SubjectID").Value
        'Closes the recordset
        RSStaff.Close()
    End Sub

End Class
```

frmConnect4NetworkYourGames.vb

```
Public Class frmConnect4NetworkYourGames

    'Declares the variable used to determine if a game has a second player or not
    Dim NoOpp As Boolean = True

    'Subroutine runs on form load
    Private Sub frmConnect4NetworkYourGames_Load(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles MyBase.Load
        'Runs the OpenDB subroutine
        OpenDB()

        'Runs the UpdateList subroutine
        UpdateList()

        'Brings the host game form to the front
        frmConnect4NetworkHostGame.BringToFront()

        'Enables the update timer
        tmrUpdate.Enabled = True
    End Sub

    'Subroutine runs when called on form load or btnRefresh_Click subroutines
```

```
    Sub UpdateList()
        'Clears the lobby listbox of data
        lstYourGames.Items.Clear()

        'Declares the variable used for the result of the confirmation messagebox
        Dim Result As MsgBoxResult
        'Declares the variable used to determine if the logged-in player is currently
playing a given game
        Dim CurrPlaying As String = ""
        'Delcares the Recordset used to view records in the database
        Dim RSStaff As New ADODB.Recordset
        If RSStaff.State = ADODB.ObjectStateEnum.adStateOpen Then RSStaff.Close()

        'Builds SQL query to execute
        RSStaff.Open("SELECT * FROM tblconnect4 WHERE HostUsername = '" &
LoggedInStudent.Username & "'", DBConn, , ADODB.LockTypeEnum.adLockOptimistic)

        'If records are found...
        If RSStaff.RecordCount > 0 Then
            'Whilst not at the end of the records...
            While Not RSStaff.EOF
                'If there is an opponent in the game...
                If RSStaff.Fields("OppUsername").Value <> "" Then
                    'Sets CurrPlaying to positive
                    CurrPlaying = " - Opponent: " &
RSStaff.Fields("OppUsername").Value
                Else
                    'Otherwise CurrPlaying is negative
                    CurrPlaying = " - No opponent"
                End If
                'Add the game details to the lobby listbox
                lstYourGames.Items.Add(RSStaff.Fields("GameID").Value & " - " &
RSStaff.Fields("GameName").Value & " - " & RSStaff.Fields("HostUsername").Value &
CurrPlaying)
                'Move on to the next record
                RSStaff.MoveNext()
            End While
        Else
            'Otherwise display an error message
            Result = MsgBox("No games found. Try hosting one?", MsgBoxStyle.YesNo)
            'If the user says yes to hosting a game...
            If Result = MsgBoxResult.Yes Then
                'Shows the Connect Four network host game form
                frmConnect4NetworkHostGame.Show()
            End If
        End If

        'Closes the recordset
        RSStaff.Close()
    End Sub

    'Subroutine runs when the host game button is clicked
    Private Sub btnHostGame_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnHostGame.Click
        'Shows the Connect Four network host game form
        frmConnect4NetworkHostGame.Show()
    End Sub

    'Subroutine runs when the selected item of the lobby listbox is changes
    Private Sub lstYourGames_SelectedIndexChanged(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles lstYourGames.SelectedIndexChanged
        'Determine the GameID of the selected game
```

```
        Dim EndofGamID As Integer = InStr(1, lstYourGames.SelectedItem, " ",
CompareMethod.Text)
        Dim GamID As String = Mid(lstYourGames.SelectedItem, 1, EndofGamID)

        'Declares the variable used to determine the username of the host
        Dim Username As String

        'Declares the Recordset used to view records in the database
        Dim RSStaff As New ADODB.Recordset
        If RSStaff.State = ADODB.ObjectStateEnum.adStateOpen Then RSStaff.Close()

        Try
            'Builds SQL query to execute
            RSStaff.Open("SELECT * FROM tblConnect4 WHERE GameID='" & GamID & "'",
DBConn, , ADODB.LockTypeEnum.adLockOptimistic)

            'Sets the selected GameID
            GameID = GamID

            'If there is an opponent...
            If RSStaff.Fields("OppUsername").Value <> "" Then

                'Sets the players
                LoggedInStudent.C4Player = "Red"
                OppStudent.C4Player = "Yellow"

                'Sets the host
                Username = RSStaff.Fields("OppUsername").Value

                'Closes the recordset
                RSStaff.Close()

                'Builds the SQl query to execute
                RSStaff.Open("SELECT * FROM tblstudents WHERE username='" & Username &
"'", DBConn, , ADODB.LockTypeEnum.adLockOptimistic)

                'If there are records found...
                If RSStaff.RecordCount > 0 Then
                    'Populates the properties of the OppStudent object with the host's
data
                    With OppStudent
                        .Fname = RSStaff.Fields("fname").Value
                        .Lname = RSStaff.Fields("lname").Value
                        .StudentID = RSStaff.Fields("StudentID").Value
                        .Username = RSStaff.Fields("username").Value
                    End With

                    'Sets the no opponent flag to false
                    NoOpp = False
                Else
                    'Sets the no opponent flag to true
                    NoOpp = True
                End If
            End If

            'Closes the recordset
            RSStaff.Close()
        Catch
        End Try
    End Sub

    'Subroutine runs when join game button is clicked
```

```
    Private Sub btnJoinGame_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnJoinGame.Click
        'Declares the Recordset used to add new records to the database
        Dim RSNewRec As New ADODB.Recordset
        If RSNewRec.State = ADODB.ObjectStateEnum.adStateOpen Then RSNewRec.Close()

        'If there is an opponent in the game...
        If NoOpp = False Then
            'Builds the SQL query to execute
            RSNewRec.Open("SELECT * FROM tblConnect4 WHERE GameID='" & GameID & "'",
DBConn, , ADODB.LockTypeEnum.adLockOptimistic)

            'Shows the Connect Four network game form
            frmConnect4Network.Show()
            'Closes this form
            Me.Close()
        Else
            'Displays an error message
            MsgBox("No opponent in game")
        End If
    End Sub

    'Subroutine runs when the refresh button is clicked
    Private Sub btnRefresh_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnRefresh.Click
        'Runs the UpdateList subroutine
        UpdateList()
    End Sub

    'Subroutine runs when the timer ticks
    Private Sub tmrUpdate_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles tmrUpdate.Tick
        'Updates the lobby list
        UpdateList()
    End Sub

End Class
```

frmConnect4Network.vb

```
Public Class frmConnect4Network

    'Declares the variable used to determine the horizontal coordinate of the square
being used
    Dim x As Integer = 1
    'Declares the variable used to determine the vertical coordinate of the square
being used
    Dim y As Integer = 1
    'Declares the array containing the coordinates of all the pictureboxes that make
up the grid
    Dim Group(7, 6) As PictureBox
    'Declares the array containing the coordinates of the squares of the grid and what
condition they are currently in
    '0 = Empty
    '1 = Red
    '2=  Yellow
    '3 = Terminator
    Dim theGrid(7, 7) As Integer
    'Declares the array containing the number of the buttons for dropping counters
    Dim Buttons(7) As Button
```

```
    'Declares the class used for both players
    Public Class C4Play
        'Declares the variables used for storing the details of the Connect Four
player
        Public Colour, Username, Fname, lname As String
        Public StudentID As Integer
    End Class
    'Declares the class used for both players' scores
    Public Class Score
        'Declares the variable used for storing score of the player
        Public ScoreNum As Integer
        'This subroutine runs when a player wins a game
        Public Sub Increase()
            'Increases the score by 1
            ScoreNum = ScoreNum + 1
        End Sub
    End Class
    'Creates two objects of the Score class, one for each player
    Dim RedScore As New Score
    Dim YellowScore As New Score
    'Creates two objects of the C4Play class, one for each player
    Dim RedPlayer As New C4Play
    Dim YellowPlayer As New C4Play

    'Declares the variable used to store the primary key of the question record in the
database
    Dim QuestionID As Integer
    'Declares the y-coords for moving the current player label
    Dim StudentCurrLocationY As Integer = 35
    Dim OppStudentCurrLocationY As Integer = 140
    'Declares the variable used for detecting wins along the y-axis
    Dim Why As Integer
    'Declares the variable used to keep track of how many reds in a row there are
    Dim RedAddUp As Integer = 0
    'Declares the variable used to keep track of how many yellow in a row there are
    Dim YellowAddUp As Integer = 0
    'Declares the variable used to determine if a question was answered correctly
    Dim QCorrect As Boolean = False
    'Declares the variables used to store whether there is a winner or the game is a
draw
    Dim Won As Boolean = False
    Dim Draw As Boolean = False
    'Declares the variable used to determine whose go it is
    Dim YourGo As Boolean

    'Subroutine runs on form load
    Private Sub frmConnect4Network_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        'Runs the OpenDB subroutine
        OpenDB()

        'Starts the update timer
        tmrUpdate.Enabled = True

        'If the logged-in student is red...
        If LoggedInStudent.C4Player = "Red" Then
            'Populate the RedPlayer object with the logged-in student's details
            With RedPlayer
                .Colour = "Red"
                .Username = LoggedInStudent.Username
                .Fname = LoggedInStudent.Fname
                .lname = LoggedInStudent.Lname
```

```
                .StudentID = LoggedInStudent.StudentID
            End With

            'Populate the YellowPlayer object with the opponent student's details
            With YellowPlayer
                .Colour = "Yellow"
                .Username = OppStudent.Username
                .Fname = OppStudent.Fname
                .lname = OppStudent.Lname
                .StudentID = OppStudent.StudentID
            End With
            'But if the logged-in student is yellow...
        Else
            'Populate the RedPlayer object with the logged-in student's details
            With RedPlayer
                .Colour = "Red"
                .Username = OppStudent.Username
                .Fname = OppStudent.Fname
                .lname = OppStudent.Lname
                .StudentID = OppStudent.StudentID
            End With

            'Populate the YellowPlayer object with the opponent student's details
            With YellowPlayer
                .Colour = "Yellow"
                .Username = LoggedInStudent.Username
                .Fname = LoggedInStudent.Fname
                .lname = LoggedInStudent.Lname
                .StudentID = LoggedInStudent.StudentID
            End With
        End If

        'Runs the AccountSection subroutine
        AccountSection()
        'Runs the MakeGrid subroutine
        MakeGrid()
        'Runs the Terminators subroutine
        Terminators()

        'Declares the variables used to store the current horizontal and vertical
coordinates of the grid
        Dim GridHor, GridVer As Integer

        'For each column of the grid...
        For GridHor = 1 To 7
            'For each square in that column...
            For GridVer = 1 To 6
                'Set the value to empty
                theGrid(GridHor, GridVer) = 0
            Next
        Next

        'Sets the scores to the defaults
        RedScore.ScoreNum = 0
        YellowScore.ScoreNum = 0

        'Declares the Recordset used to view records in the database
        Dim RSStaff As New ADODB.Recordset
        If RSStaff.State = ADODB.ObjectStateEnum.adStateOpen Then RSStaff.Close()

        'Builds SQL query to execute
```

```
        RSStaff.Open("SELECT * FROM tblConnect4 WHERE GameID='" & GameID & "'",
DBConn, , ADODB.LockTypeEnum.adLockOptimistic)

        'Sets the current player
        C4Player = RSStaff.Fields("CurrentPlayer").Value
        CurrPlayer()

        'Hides the counter placing buttons
        btn1.Visible = False
        btn2.Visible = False
        btn3.Visible = False
        btn4.Visible = False
        btn5.Visible = False
        btn6.Visible = False
        btn7.Visible = False

        'If the logged-in student is the current player...
        If LoggedInStudent.C4Player = C4Player Then
            'Hides the question controls
            grpQuestion.Visible = True
            txtAnswer.Visible = True
            lblQuestion.Visible = True
            btnSubmit.Visible = True

            'Runs the Question subroutine
            Question()
        Else
            'Shows the question controls
            grpQuestion.Visible = False
            txtAnswer.Visible = False
            lblQuestion.Visible = False
            btnSubmit.Visible = False
        End If
    End Sub

    'Subroutine runs when called in DetectWinner subroutine
    Sub DatabaseDetails()
        'Declares the Recordset used to add new records to the database
        Dim RSNewRec As New ADODB.Recordset
        If RSNewRec.State = ADODB.ObjectStateEnum.adStateOpen Then RSNewRec.Close()
        'Declares the StreamWriter used to write to the game breakdown text files
        Dim writer As System.IO.StreamWriter

        'If the red player is the winner...
        If RedScore.ScoreNum = 1 Then
            'Opens a recordset
            RSNewRec.Open("SELECT * FROM tblstudents WHERE Username='" &
RedPlayer.Username & "'", DBConn, , ADODB.LockTypeEnum.adLockOptimistic)
            'Adds a win to the winning player's record
            With RSNewRec
                .Update()
                .Fields("Wins").Value = .Fields("Wins").Value + 1
                RSNewRec.Update()
            End With
            'Closes the recordset
            RSNewRec.Close()

            'Opens a recordset
            RSNewRec.Open("SELECT * FROM tblstudents WHERE Username='" &
YellowPlayer.Username & "'", DBConn, , ADODB.LockTypeEnum.adLockOptimistic)
            'Adds a loss to the losing player's record
            With RSNewRec
```

```
                .Update()
                .Fields("Losses").Value = .Fields("Losses").Value + 1
                RSNewRec.Update()
            End With
            'Closes the recordset
            RSNewRec.Close()

            'Sets the path to where the log shall be generated, and the filename
            writer = My.Computer.FileSystem.OpenTextFileWriter("G:\Computing Group\Y10
2009-10\Ben's\Edutainment Suite\Connect4\bin\Debug\WinsLosses\" & RedPlayer.Username &
".txt", True)

            'Writes the relevant data to the log
            writer.WriteLine(RedPlayer.Fname & " " & RedPlayer.lname & " beat " &
YellowPlayer.Fname & " " & YellowPlayer.lname & " in Connect Four - " & TimeOfDay & "
" & DateValue(Now))
            'Saves the log file
            writer.Close()

            'Sets the path to where the log shall be generated, and the filename
            writer = My.Computer.FileSystem.OpenTextFileWriter("G:\Computing Group\Y10
2009-10\Ben's\Edutainment Suite\Connect4\bin\Debug\WinsLosses\" &
YellowPlayer.Username & ".txt", True)

            'Writes the relevant data to the log
            writer.WriteLine(YellowPlayer.Fname & " " & YellowPlayer.lname & " was
beaten by " & RedPlayer.Fname & " " & RedPlayer.lname & " in Connect Four - " &
TimeOfDay & " " & DateValue(Now))
            'Saves the log file
            writer.Close()

            'Resets the red score back to its default
            RedScore.ScoreNum = 0
        End If

        'However, if the yellow player is the winner...
        If YellowScore.ScoreNum = 1 Then
            'Opens a recordset
            RSNewRec.Open("SELECT * FROM tblstudents WHERE Username='" &
YellowPlayer.Username & "'", DBConn, , ADODB.LockTypeEnum.adLockOptimistic)
            'Adds a win to the winning player's record
            With RSNewRec
                .Update()
                .Fields("Wins").Value = .Fields("Wins").Value + 1
                RSNewRec.Update()
            End With
            'Closes the recordset
            RSNewRec.Close()

            'Opens a recordset
            RSNewRec.Open("SELECT * FROM tblstudents WHERE Username='" &
RedPlayer.Username & "'", DBConn, , ADODB.LockTypeEnum.adLockOptimistic)
            'Adds a loss to the losing player's record
            With RSNewRec
                .Update()
                .Fields("Losses").Value = .Fields("Losses").Value + 1
                RSNewRec.Update()
            End With
            'Closes the recordset
            RSNewRec.Close()

            'Sets the path to where the log shall be generated, and the filename
```

```
            writer = My.Computer.FileSystem.OpenTextFileWriter("G:\Computing Group\Y10
2009-10\Ben's\Edutainment Suite\Connect4\bin\Debug\WinsLosses\" &
YellowPlayer.Username & ".txt", True)

            'Writes the relevant data to the log
            writer.WriteLine(YellowPlayer.Fname & " " & YellowPlayer.lname & " beat "
& RedPlayer.Fname & " " & RedPlayer.lname & " in Connect Four - " & TimeOfDay & " " &
DateValue(Now))
            'Saves the log file
            writer.Close()

            'Sets the path to where the log shall be generated, and the filename
            writer = My.Computer.FileSystem.OpenTextFileWriter("G:\Computing Group\Y10
2009-10\Ben's\Edutainment Suite\Connect4\bin\Debug\WinsLosses\" & RedPlayer.Username &
".txt", True)

            'Writes the relevant data to the log
            writer.WriteLine(RedPlayer.Fname & " " & RedPlayer.lname & " was beaten by
" & YellowPlayer.Fname & " " & YellowPlayer.lname & " in Connect Four - " & TimeOfDay
& " " & DateValue(Now))
            'Saves the log file
            writer.Close()

            'Resets the red score back to its default
            YellowScore.ScoreNum = 0
        End If

        'However, if the game is a draw...
        If Draw = True Then
            'Opens a recordset
            RSNewRec.Open("SELECT * FROM tblstudents WHERE Username='" &
RedPlayer.Username & "'", DBConn, , ADODB.LockTypeEnum.adLockOptimistic)
            'Adds a draw to the player's record
            With RSNewRec
                .Update()
                .Fields("Draws").Value = .Fields("Draws").Value + 1
                RSNewRec.Update()
            End With
            'Closes the recordset
            RSNewRec.Close()

            'Opens a recordset
            RSNewRec.Open("SELECT * FROM tblstudents WHERE Username='" &
YellowPlayer.Username & "'", DBConn, , ADODB.LockTypeEnum.adLockOptimistic)
            'Adds a draw to the other player's record
            With RSNewRec
                .Update()
                .Fields("Draws").Value = .Fields("Draws").Value + 1
                RSNewRec.Update()
            End With
            'Closes the recordset
            RSNewRec.Close()

            'Sets the path to where the log shall be generated, and the filename
            writer = My.Computer.FileSystem.OpenTextFileWriter("G:\Computing Group\Y10
2009-10\Ben's\Edutainment Suite\Connect4\bin\Debug\WinsLosses\" & RedPlayer.Username &
".txt", True)

            'Writes the relevant data to the log
            writer.WriteLine(RedPlayer.Fname & " " & RedPlayer.lname & " drew with " &
YellowPlayer.Fname & " " & YellowPlayer.lname & " in Connect Four - " & TimeOfDay & "
" & DateValue(Now))
```

```
            'Saves the log file
            writer.Close()

            'Sets the path to where the log shall be generated, and the filename
            writer = My.Computer.FileSystem.OpenTextFileWriter("G:\Computing Group\Y10
2009-10\Ben's\Edutainment Suite\Connect4\bin\Debug\WinsLosses\" &
YellowPlayer.Username & ".txt", True)

            'Writes the relevant data to the log
            writer.WriteLine(YellowPlayer.Fname & " " & YellowPlayer.lname & " drew
with " & RedPlayer.Fname & " " & RedPlayer.lname & " in Connect Four - " & TimeOfDay &
" " & DateValue(Now))
            'Saves the log file
            writer.Close()
        End If
    End Sub

    'Subroutine runs when called in the ChangePlayer subroutine
    Sub CurrPlayer()
        'If the current player is red...
        If C4Player = "Red" Then
            'Places the current player label pointing to the red player
            lblCurrPlayer.Top = StudentCurrLocationY
            'However if the current player is yellow...
        Else
            'Places the current player label pointing to the yellow player
            lblCurrPlayer.Top = OppStudentCurrLocationY
        End If
    End Sub

    'Subroutine runs when called in form load subroutine
    Sub AccountSection()
        'Populates the player name labels with data and get a picture of each player
        lblStudentName.Text = RedPlayer.Fname & " " & RedPlayer.lname
        picStudent.ImageLocation = "G:\Computing Group\Y10 2009-10\Ben's\Edutainment
Suite\Connect4\bin\Debug\images\students\" & RedPlayer.Username & ".jpg"

        lblOppStudentName.Text = YellowPlayer.Fname & " " & YellowPlayer.lname
        picOppStudentPic.ImageLocation = "G:\Computing Group\Y10 2009-
10\Ben's\Edutainment Suite\Connect4\bin\Debug\images\students\" &
YellowPlayer.Username & ".jpg"
    End Sub

    'Subroutine runs when called in form load subroutine
    Sub MakeGrid()
        'Ties the Buttons array with the button controls for dropping counters
        Buttons(1) = btn1
        Buttons(2) = btn2
        Buttons(3) = btn3
        Buttons(4) = btn4
        Buttons(5) = btn5
        Buttons(6) = btn6
        Buttons(7) = btn7

        'Ties the Group coordinates with the pictureboxes on the form
        Group(1, 1) = pb1dash1
        Group(1, 2) = pb1dash2
        Group(1, 3) = pb1dash3
        Group(1, 4) = pb1dash4
        Group(1, 5) = pb1dash5
        Group(1, 6) = pb1dash6
```

```
        Group(2, 1) = pb2dash1
        Group(2, 2) = pb2dash2
        Group(2, 3) = pb2dash3
        Group(2, 4) = pb2dash4
        Group(2, 5) = pb2dash5
        Group(2, 6) = pb2dash6

        Group(3, 1) = pb3dash1
        Group(3, 2) = pb3dash2
        Group(3, 3) = pb3dash3
        Group(3, 4) = pb3dash4
        Group(3, 5) = pb3dash5
        Group(3, 6) = pb3dash6

        Group(4, 1) = pb4dash1
        Group(4, 2) = pb4dash2
        Group(4, 3) = pb4dash3
        Group(4, 4) = pb4dash4
        Group(4, 5) = pb4dash5
        Group(4, 6) = pb4dash6

        Group(5, 1) = pb5dash1
        Group(5, 2) = pb5dash2
        Group(5, 3) = pb5dash3
        Group(5, 4) = pb5dash4
        Group(5, 5) = pb5dash5
        Group(5, 6) = pb5dash6

        Group(6, 1) = pb6dash1
        Group(6, 2) = pb6dash2
        Group(6, 3) = pb6dash3
        Group(6, 4) = pb6dash4
        Group(6, 5) = pb6dash5
        Group(6, 6) = pb6dash6

        Group(7, 1) = pb7dash1
        Group(7, 2) = pb7dash2
        Group(7, 3) = pb7dash3
        Group(7, 4) = pb7dash4
        Group(7, 5) = pb7dash5
        Group(7, 6) = pb7dash6

        'Runs the NetworkCheckGrid subroutine
        NetworkCheckGrid()
    End Sub

    'Subroutine runs when called in MakeGrid subroutine
    Sub NetworkCheckGrid()
        'Declares the Recordset used to view records in the database
        Dim RSStaff As New ADODB.Recordset
        If RSStaff.State = ADODB.ObjectStateEnum.adStateOpen Then RSStaff.Close()

        'Builds SQL query to execute
        RSStaff.Open("SELECT * FROM tblConnect4 WHERE GameID='" & GameID & "'",
DBConn, , ADODB.LockTypeEnum.adLockOptimistic)

        'For each column...
        For x = 1 To 7
            'For each square in the current column...
            For y = 1 To 6
                'If the current square is taken by the red player...
                If RSStaff.Fields(x & "," & y).Value = 1 Then
```

```
                    'Update the graphics to reflect this
                    theGrid(x, y) = 1
                    Group(x, y).Image = My.Resources.red
                End If
                'If the current square is taken by the yellow player...
                If RSStaff.Fields(x & "," & y).Value = 2 Then
                    'Update the graphics to reflect this
                    theGrid(x, y) = 2
                    Group(x, y).Image = My.Resources.yellow
                End If
            Next
        Next

        'For each column...
        For x = 1 To 7
            'If any buttons are disabled...
            If RSStaff.Fields("btn" & x).Value = 0 Then
                'Disabled them in the form
                Buttons(x).Enabled = False
            End If
        Next

        'Closes the recordset
        RSStaff.Close()

        'Resets x and y to their default values
        x = 1
        y = 1
    End Sub

    'Subroutine runs when called in form load subroutine
    Sub Terminators()
        'Sets the extra seventh horizontal line of squares in theGrid to terminators
        While y <> 8
            theGrid(y, 7) = 3
            y = y + 1
        End While

        'Sets x back to 1 for use later
        y = 1
    End Sub

    'Subroutine runs when called in button click subroutines
    Sub CounterPlace()
        'Declares the Recordset used to add new records to the database
        Dim RSNewRec As New ADODB.Recordset
        If RSNewRec.State = ADODB.ObjectStateEnum.adStateOpen Then RSNewRec.Close()

        'Opens a recordset
        RSNewRec.Open("SELECT * FROM tblconnect4 WHERE GameID='" & GameID & "'",
DBConn, , ADODB.LockTypeEnum.adLockOptimistic)

        'Runs down the column to find the next blank space
        While theGrid(x, y) = 0
            y = y + 1
        End While

        'Changes the next blank space into the current player's marker and claims the
square for them
        If theGrid(x, 2) = 0 Then
            y = y - 1
```

```
            If C4Player = "Red" Then
                theGrid(x, y) = 1
                RSNewRec.Fields(x & "," & y).Value = 1
                Group(x, y).Image = My.Resources.red
            Else
                theGrid(x, y) = 2
                RSNewRec.Fields(x & "," & y).Value = 2
                Group(x, y).Image = My.Resources.yellow
            End If
        Else
            'If the only square left in the current row is the topmost one, the button
disables after being pressed to seal off the column
            y = y - 2
            If C4Player = "Red" Then
                theGrid(x, y) = 1
                RSNewRec.Fields(x & "," & y).Value = 1
                Group(x, y).Image = My.Resources.red
            Else
                theGrid(x, y) = 2
                RSNewRec.Fields(x & "," & y).Value = 2
                Group(x, y).Image = My.Resources.yellow
            End If
            Buttons(x).Enabled = False
            RSNewRec.Fields("btn" & x).Value = 0
        End If

        'Updates the database to reflect the change in the grid
        RSNewRec.Update()
        'Closes the recordset
        RSNewRec.Close()

        'Runs the DetectWinner subroutine
        DetectWinner()

        'If the game is still on, runs the ChangePlayer subroutine
        If Won = False Then
            ChangePlayer()
        End If
    End Sub

    'Subroutine runs when called in CounterPlace subroutine
    Sub DetectWinner()
        'Declares the variable used to determine number of spaces to the left of the
last-placed counter
        Dim SpacestoLeft As Integer
        'Declares the variable used  to determine number of spaces to the right of the
last-placed counter
        Dim SpacestoRight As Integer
        'Declares the variable used  to determine number of spaces above the last-
placed counter
        Dim SpacesAbove As Integer
        'Declares the variable used  to determine number of spaces below the last-
placed counter
        Dim SpacesBelow As Integer

        'Declares the Recordset used to add new records to the database
        Dim RSNewRec As New ADODB.Recordset
        If RSNewRec.State = ADODB.ObjectStateEnum.adStateOpen Then RSNewRec.Close()
        'Declares the Recordset used to view records in the database
        Dim RSStaff As New ADODB.Recordset
        If RSStaff.State = ADODB.ObjectStateEnum.adStateOpen Then RSStaff.Close()
```

```
        'Builds SQL query to execute
        RSStaff.Open("SELECT * FROM tblConnect4 WHERE GameID='" & GameID & "'",
DBConn, , ADODB.LockTypeEnum.adLockOptimistic)

        'If the game has been won by red...
        If RSStaff.Fields("Won").Value = 1 Then
            'Ends the game, Red is the winner
            tmrUpdate.Enabled = False
            MsgBox("Red wins")
            lblCurrPlayer.Visible = False
            btnQuit.Visible = True
            RSStaff.Close()
            Exit Sub
            'However if the game has been won by yellow...
        ElseIf RSStaff.Fields("Won").Value = 2 Then
            'Ends the game, Yellow is the winner
            tmrUpdate.Enabled = False
            MsgBox("Yellow wins")
            lblCurrPlayer.Visible = False
            btnQuit.Visible = True
            RSStaff.Close()
            Exit Sub
        End If

        '///WIN CONDITIONS\\\

        '/HORIZONTAL\

        'Determines no. of spaces to left and right of last-placed coutner
        SpacestoLeft = x - 1
        SpacestoRight = 7 - x

        'Opens a recordset
        RSNewRec.Open("SELECT * FROM tblconnect4 WHERE GameID='" & GameID & "'",
DBConn, , ADODB.LockTypeEnum.adLockOptimistic)


        'Runs along the row of the last-placed counter to see if there are four red or
four yellows in a row
        For HorizSquare As Integer = x - SpacestoLeft To x + SpacestoRight
            If theGrid(HorizSquare, y) = 1 Then
                YellowAddUp = 0
                RedAddUp = RedAddUp + 1
                'If there are four reds in a row horizontally...
                If RedAddUp = 4 Then
                    'Gives the red player a win, pops up a messagebox, makes changes
to the database and sets up the game to be reset
                    Won = True
                    tmrUpdate.Enabled = False
                    RedScore.Increase()
                    MsgBox("Red wins")
                    DatabaseDetails()
                    lblCurrPlayer.Visible = False
                    btnQuit.Visible = True
                    RSNewRec.Fields("Won").Value = 1
                    RSNewRec.Update()
                    RSNewRec.Close()
                    Exit Sub
                End If
            End If
            If theGrid(HorizSquare, y) = 2 Then
                RedAddUp = 0
```

```
                YellowAddUp = YellowAddUp + 1
                'If there are four yellow in a row horizontally...
                If YellowAddUp = 4 Then
                    'Gives the yellow player a win, pops up a messagebox, makes
changes to the database and sets up the game to be reset
                    Won = True
                    tmrUpdate.Enabled = False
                    YellowScore.Increase()
                    MsgBox("Yellow wins")
                    DatabaseDetails()
                    lblCurrPlayer.Visible = False
                    btnQuit.Visible = True
                    RSNewRec.Fields("Won").Value = 2
                    RSNewRec.Update()
                    RSNewRec.Close()
                    Exit Sub
                End If
            End If
            If theGrid(HorizSquare, y) = 0 Then
                RedAddUp = 0
                YellowAddUp = 0
            End If
        Next

        '/VERTICAL\

        'Determines no. of spaces above and below the last-placed coutner
        SpacesAbove = y - 1
        SpacesBelow = 7 - y

        'Runs down the column of the last-placed counter to see if there are four red
or four yellows in a row
        For VertSquare As Integer = y - SpacesAbove To y + SpacesBelow
            If theGrid(x, VertSquare) = 1 Then
                YellowAddUp = 0
                RedAddUp = RedAddUp + 1
                'If there are four reds in a row vertically...
                If RedAddUp = 4 Then
                    'Gives the red player a win, pops up a messagebox, makes changes
to the database and sets up the game to be reset
                    Won = True
                    tmrUpdate.Enabled = False
                    RedScore.Increase()
                    MsgBox("Red wins")
                    DatabaseDetails()
                    lblCurrPlayer.Visible = False
                    btnQuit.Visible = True
                    RSNewRec.Fields("Won").Value = 1
                    RSNewRec.Update()
                    RSNewRec.Close()
                    Exit Sub
                End If
            End If
            If theGrid(x, VertSquare) = 2 Then
                RedAddUp = 0
                YellowAddUp = YellowAddUp + 1
                'If there are four yellow in a row vertically...
                If YellowAddUp = 4 Then
                    'Gives the yellow player a win, pops up a messagebox, makes
changes to the database and sets up the game to be reset
                    Won = True
                    tmrUpdate.Enabled = False
```

```
                    YellowScore.Increase()
                    MsgBox("Yellow wins")
                    DatabaseDetails()
                    lblCurrPlayer.Visible = False
                    btnQuit.Visible = True
                    RSNewRec.Fields("Won").Value = 2
                    RSNewRec.Update()
                    RSNewRec.Close()
                    Exit Sub
                End If
            End If
            If theGrid(x, VertSquare) = 0 Then
                RedAddUp = 0
                YellowAddUp = 0
            End If
        Next

        'If a diagonal win in possible with the position of the last-placed counter...
        If x < 5 Then

            '/DIAGONAL BOTTOM-UP\

            'Runs the Diag (Bottom-Up) Win Detection subroutine
            DiagBUWin()

        End If

        'If a diagonal win in possible with the position of the last-placed counter...
        If x > 3 Then

            '/DIAGONAL TOP-DOWN\

            'Runs the Diag (Top-Down) Win Detection subroutine
            DiagTDWin()

        End If

        'If there is no winner...
        If Won <> True Then
            'Declares the variable used to keep track of how many blank squares there
are left
            Dim Blanks As Integer = 42
            'Goes through the grid row-by-row, column-by-column, decrementing the
blanks value when a non-blank square in encountered
            For Why = 1 To 6
                For x = 1 To 7
                    If theGrid(x, Why) <> 0 Then
                        Blanks = Blanks - 1
                    End If
                Next
            Next
            'If there are no blanks left...
            If Blanks = 0 Then
                'Declares the game a draw, amends the database, sets the form up to
reset
                Draw = True
                MsgBox("No-one wins, it's a draw")
                DatabaseDetails()
                lblCurrPlayer.Visible = False
                btnQuit.Visible = True
            End If
        End If
```

```
        'Closes the recordsets
        RSStaff.Close()
        RSNewRec.Close()
    End Sub

    'Subroutine runs when called in DetectWinner subroutine
    Sub DiagTDWin()
       'Declares the varables used to run through the diagonals
        Dim v, z, a, w As Integer
        'Declares the Recordset used to add new records to the database
        Dim RSNewRec As New ADODB.Recordset
        If RSNewRec.State = ADODB.ObjectStateEnum.adStateOpen Then RSNewRec.Close()
        'Declares the Recordset used to view records in the database
        Dim RSStaff As New ADODB.Recordset
        If RSStaff.State = ADODB.ObjectStateEnum.adStateOpen Then RSStaff.Close()

        'Builds SQL query to execute
        RSStaff.Open("SELECT * FROM tblConnect4 WHERE GameID='" & GameID & "'",
DBConn, , ADODB.LockTypeEnum.adLockOptimistic)

        'Sets the variables to their defaults
        x = 1
        v = x
        a = x
        z = x + 3
        y = 1

        'Runs through the diagonals to determine if there are four red or yellow
counters in a row
        For Why = 1 To 3
            For x = 1 To 4
                w = x
                y = Why
                If theGrid(x, y) = 1 Then
                    RedAddUp = 1
                    For v = a To z
                        x = x + 1
                        y = y + 1
                        If x < 8 Then
                            If theGrid(x, y) = 1 Then
                                RedAddUp = RedAddUp + 1
                            End If
                            'If there are four reds in a row diagonally...
                            If RedAddUp = 4 Then
                                'Gives the red player a win, pops up a messagebox,
makes changes to the database and sets up the game to be reset
                                Won = True
                                tmrUpdate.Enabled = False
                                RedScore.Increase()
                                MsgBox("Red wins")
                                DatabaseDetails()
                                lblCurrPlayer.Visible = False
                                btnQuit.Visible = True
                                RSNewRec.Fields("Won").Value = 1
                                RSNewRec.Update()
                                RSNewRec.Close()
                                Exit Sub
                            End If
                        End If
                    Next
                End If
```

```
                If theGrid(w, y) = 2 Then
                    YellowAddUp = 1
                    For v = a To z
                        w = w + 1
                        y = y + 1
                        If w < 8 Then
                            If theGrid(w, y) = 1 Then
                                YellowAddUp = YellowAddUp + 1
                            End If
                            'If there are four yellow in a row diagonally...
                            If YellowAddUp = 4 Then
                                'Gives the yellow player a win, pops up a messagebox,
makes changes to the database and sets up the game to be reset
                                Won = True
                                tmrUpdate.Enabled = False
                                YellowScore.Increase()
                                MsgBox("Yellow wins")
                                DatabaseDetails()
                                lblCurrPlayer.Visible = False
                                btnQuit.Visible = True
                                RSNewRec.Fields("Won").Value = 2
                                RSNewRec.Update()
                                RSNewRec.Close()
                                Exit Sub
                            End If
                        End If
                    Next
                End If
            Next
        Next



        'Resets the values to their defaults
        RedAddUp = 0
        YellowAddUp = 0
    End Sub

    'Subroutine runs when called in DetectWinner subroutine
    Sub DiagBUWin()
   'Declares the varables used to run through the diagonals
        Dim v, z, a, w As Integer
        'Declares the Recordset used to add new records to the database
        Dim RSNewRec As New ADODB.Recordset
        If RSNewRec.State = ADODB.ObjectStateEnum.adStateOpen Then RSNewRec.Close()
        'Declares the Recordset used to view records in the database
        Dim RSStaff As New ADODB.Recordset
        If RSStaff.State = ADODB.ObjectStateEnum.adStateOpen Then RSStaff.Close()

        'Builds SQL query to execute
        RSStaff.Open("SELECT * FROM tblConnect4 WHERE GameID='" & GameID & "'",
DBConn, , ADODB.LockTypeEnum.adLockOptimistic)

        'Sets the variables to their defaults
        x = 1
        v = x
        a = x
        z = x + 3
        y = 6
```

```
        'Runs through the diagonals to determine if there are four red or yellow
counters in a row
        For Why = 4 To 6
            For x = 1 To 4
                w = x
                y = Why
                If theGrid(x, y) = 1 Then
                    RedAddUp = 1
                    For v = a To z
                        x = x + 1
                        y = y - 1
                        If x < 8 Then
                            If theGrid(x, y) = 1 Then
                                RedAddUp = RedAddUp + 1
                            End If
                            'If there are four reds in a row diagonally...
                            If RedAddUp = 4 Then
                                'Gives the red player a win, pops up a messagebox,
makes changes to the database and sets up the game to be reset
                                Won = True
                                tmrUpdate.Enabled = False
                                RedScore.Increase()
                                MsgBox("Red wins")
                                DatabaseDetails()
                                lblCurrPlayer.Visible = False
                                btnQuit.Visible = True
                                RSNewRec.Fields("Won").Value = 1
                                RSNewRec.Update()
                                RSNewRec.Close()
                                Exit Sub
                            End If
                        End If
                    Next
                End If

                If theGrid(w, y) = 2 Then
                    YellowAddUp = 1
                    For v = a To z
                        w = w + 1
                        y = y - 1
                        If w < 8 Then
                            If theGrid(w, y) = 1 Then
                                YellowAddUp = YellowAddUp + 1
                            End If
                            'If there are four yellows in a row diagonally...
                            If YellowAddUp = 4 Then
                                'Gives the yellow player a win, pops up a messagebox,
makes changes to the database and sets up the game to be reset
                                Won = True
                                tmrUpdate.Enabled = False
                                YellowScore.Increase()
                                MsgBox("Yellow wins")
                                DatabaseDetails()
                                lblCurrPlayer.Visible = False
                                btnQuit.Visible = True
                                RSNewRec.Fields("Won").Value = 2
                                RSNewRec.Update()
                                RSNewRec.Close()
                                Exit Sub
                            End If
                        End If
                    Next
```

```
                End If
            Next
        Next

        'Resets the values to their defaults
        RedAddUp = 0
        YellowAddUp = 0
    End Sub

    'Subroutines run when their respective buttons are clicked
    Private Sub btn1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles btn1.Click
        'Sets x to the x-coord of the selected column
        x = 1
        'Runs the CounterPlace subroutine
        CounterPlace()
    End Sub
    Private Sub btn2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles btn2.Click
        x = 2
        CounterPlace()
    End Sub
    Private Sub btn3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles btn3.Click
        x = 3
        CounterPlace()
    End Sub
    Private Sub btn4_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles btn4.Click
        x = 4
        CounterPlace()
    End Sub
    Private Sub btn5_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles btn5.Click
        x = 5
        CounterPlace()
    End Sub
    Private Sub btn6_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles btn6.Click
        x = 6
        CounterPlace()
    End Sub
    Private Sub btn7_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles btn7.Click
        x = 7
        CounterPlace()
    End Sub

    'Subroutine runs when back button is clicked
    Private Sub btnBack_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnBack.Click
        'Shows the Connect Four menu
        frmConnect4Menu.Show()
        'Closes this form
        Me.Close()
    End Sub

    'Subroutine runs when called in ChangePlayer and btnReset_Click subroutines
    Sub Question()
        'Declares the Recordset used to view records in the database
        Dim RSStaff As New ADODB.Recordset
        If RSStaff.State = ADODB.ObjectStateEnum.adStateOpen Then RSStaff.Close()
```

```
        'Builds SQL query to execute
        RSStaff.Open("SELECT * FROM tblConnect4 WHERE GameID='" & GameID & "'",
DBConn, , ADODB.LockTypeEnum.adLockOptimistic)

        'Gets the question parameters of the current game
        C4HSubject = RSStaff.Fields("C4NSubject").Value
        C4HDifficulty = RSStaff.Fields("C4NDifficulty").Value
        C4HTopic = RSStaff.Fields("C4NTopic").Value

        'Closes the recordset
        RSStaff.Close()

        'Builds SQL query to execute
        RSStaff.Open("SELECT * FROM tblquestions WHERE SubjectID='" & C4HSubject & "'
AND Difficulty='" & C4HDifficulty & "' AND Topic='" & C4HTopic & "' ORDER BY RAND()
LIMIT 1", DBConn, , ADODB.LockTypeEnum.adLockOptimistic)

        'Sets the QuestionID variable to that of the selected question
        QuestionID = RSStaff.Fields("QuestionID").Value

        'Displays the selected question
        lblQuestion.Text = RSStaff.Fields("Question").Value
        'Closes the recordset
        RSStaff.Close()
    End Sub

    'Subroutine runs when called in form load and CounterPlace subroutines
    Sub ChangePlayer()
        'Sets x & y back to 1 for use later
        x = 1
        y = 1

        'Changes the current player
        If C4Player = "Red" Then
            If LoggedInStudent.C4Player = "Yellow" Then
                YourGo = True
            Else
                YourGo = False
            End If

            C4Player = "Yellow"
            CurrPlayer()
        Else
            If LoggedInStudent.C4Player = "Red" Then
                YourGo = True
            Else
                YourGo = False
            End If

            C4Player = "Red"
            CurrPlayer()
        End If

        'If the logged-in player is up...
        If YourGo = True Then
            'Displays the question controls
            grpQuestion.Visible = True
            txtAnswer.Visible = True
            lblQuestion.Visible = True
            btnSubmit.Visible = True
```

```
            'Runs the Question subroutine
            Question()
        Else
            'Hides the question controls
            grpQuestion.Visible = False
            txtAnswer.Visible = False
            lblQuestion.Visible = False
            btnSubmit.Visible = False
        End If

        'Declares the Recordset used to add new records to the database
        Dim RSNewRec As New ADODB.Recordset
        If RSNewRec.State = ADODB.ObjectStateEnum.adStateOpen Then RSNewRec.Close()

        'Opens a recordset
        RSNewRec.Open("SELECT * FROM tblconnect4 WHERE GameID='" & GameID & "'",
DBConn, , ADODB.LockTypeEnum.adLockOptimistic)
        'Changes the record for the current field
        RSNewRec.Fields("CurrentPlayer").Value = C4Player
        'Updates the database
        RSNewRec.Update()
        'Closes the recordset
        RSNewRec.Close()

        'Hides the counter placing buttons
        btn1.Visible = False
        btn2.Visible = False
        btn3.Visible = False
        btn4.Visible = False
        btn5.Visible = False
        btn6.Visible = False
        btn7.Visible = False
    End Sub

    'Subroutine runs when called in the tmrUpdate_Tick subroutine
    Sub CheckPlayer()
        'Declares the Recordset used to view records in the database
        Dim RSStaff As New ADODB.Recordset
        If RSStaff.State = ADODB.ObjectStateEnum.adStateOpen Then RSStaff.Close()

        'Builds SQL query to execute
        RSStaff.Open("SELECT * FROM tblConnect4 WHERE GameID='" & GameID & "'",
DBConn, , ADODB.LockTypeEnum.adLockOptimistic)

        'If the logged-in student and current player in the database are the same but
the current player on the form isn't, changes the current player on the form
        If LoggedInStudent.C4Player = "Red" Then
            If RSStaff.Fields("CurrentPlayer").Value = "Red" Then
                If C4Player = "Yellow" Then
                    ChangePlayer()
                End If
            End If
        ElseIf LoggedInStudent.C4Player = "Yellow" Then
            If RSStaff.Fields("CurrentPlayer").Value = "Yellow" Then
                If C4Player = "Red" Then
                    ChangePlayer()
                End If
            End If
        End If
    End Sub

    'Subroutine runs then answer submit button is clicked
```

```
    Private Sub btnSubmit_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnSubmit.Click
        'Declares the Recordset used to view records in the database
        Dim RSStaff As New ADODB.Recordset
        If RSStaff.State = ADODB.ObjectStateEnum.adStateOpen Then RSStaff.Close()

        'Hides the question controls
        grpQuestion.Visible = False
        lblQuestion.Visible = False
        txtAnswer.Visible = False
        btnSubmit.Visible = False

        'Builds SQL query to execute
        RSStaff.Open("SELECT * FROM tblquestions WHERE QuestionID='" & QuestionID &
"'", DBConn, , ADODB.LockTypeEnum.adLockOptimistic)

        'If the answer is correct...
        If txtAnswer.Text = RSStaff.Fields("Answer").Value Then
            'Display a message box
            MsgBox("Correct!")

            'Sets the question correct flag to true
            QCorrect = True

            'Makes the counter placement buttons visible
            btn1.Visible = True
            btn2.Visible = True
            btn3.Visible = True
            btn4.Visible = True
            btn5.Visible = True
            btn6.Visible = True
            btn7.Visible = True

            'Runs the QDatabase subroutine
            QDatabase()
            'However if the answer if incorrect...
        Else
            'Display a message box
            MsgBox("Incorrect!")

            'Sets the question correct flag to false
            QCorrect = False

            'Runs the QDatabase subroutine
            QDatabase()
            'Runs the ChangePlayer subroutine
            ChangePlayer()
        End If

        'Blanks the answer textbox for the next question
        txtAnswer.Text = ""
    End Sub

    'Subroutine runs when called in btnSubmit_Click subroutine
    Sub QDatabase()
        'Declares the Recordset used to add new records to the database
        Dim RSNewRec As New ADODB.Recordset
        If RSNewRec.State = ADODB.ObjectStateEnum.adStateOpen Then RSNewRec.Close()
        'Opens a recordset
        RSNewRec.Open("SELECT * FROM tblattempted", DBConn, ,
ADODB.LockTypeEnum.adLockOptimistic)
        'Adds the new record
```

```
        With RSNewRec
            .AddNew()
            .Fields("SubjectID").Value = C4HSubject
            .Fields("QuestionID").Value = QuestionID
            If C4Player = "Red" Then
                If LoggedInStudent.C4Player = "Red" Then
                    .Fields("StudentID").Value = RedPlayer.StudentID
                Else
                    .Fields("StudentID").Value = YellowPlayer.StudentID
                End If
            Else
                If LoggedInStudent.C4Player = "Yellow" Then
                    .Fields("StudentID").Value = LoggedInStudent.StudentID
                Else
                    .Fields("StudentID").Value = OppStudent.StudentID
                End If
            End If
            .Fields("When").Value = TimeOfDay & " " & DateValue(Now)
            If QCorrect = True Then
                .Fields("Correct").Value = 1
            Else
                .Fields("Correct").Value = 0
            End If

            'Resets the question correct flag
            QCorrect = False

            'Updates the recordset
            RSNewRec.Update()
            'Closes the recordset
            RSNewRec.Close()
        End With
    End Sub

    'Subroutine runs then the quit button is clicked
    Private Sub btnQuit_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnQuit.Click
        'Declares the recordset used to delete records from the database
        Dim RSDelRec As New ADODB.Recordset

        'Opens a recordset and deletes the record of the current game
        RSDelRec.Open("SELECT * FROM tblConnect4 WHERE GameID='" & GameID & "'",
DBConn, , ADODB.LockTypeEnum.adLockOptimistic)
        If RSDelRec.RecordCount > 0 Then
            RSDelRec.Delete()
        End If

        'Closes the recordset
        RSDelRec.Close()

        frmConnect4NetworkLobby.Show()
        Me.Close()
    End Sub

    'Subroutine runs every time the update timer ticks
    Private Sub tmrUpdate_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles tmrUpdate.Tick
        Try
            'Runs the NetworkCheckGrid subroutine
            NetworkCheckGrid()
            'Runs the DetectWinner subroutine
            DetectWinner()
```

```
            'Runs the CheckPlayer subroutine
            CheckPlayer()
        Catch
        End Try
    End Sub

    'Subroutine runs when view logged-in student's profile button is clicked
    Private Sub btnViewProfile_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnViewProfile.Click
        'If the logged-in student is the red player...
        If LoggedInStudent.C4Player = "Red" Then
            'Sets viewed profile to logged-in student's
            Viewing = 1
            'Shows the student account form
            frmStudentAccount.Show()
        Else
            'Sets viewed profile to opponent student's
            Viewing = 2
            'Shows the student account form
            frmStudentAccount.Show()
        End If
    End Sub

    'Subroutine runs when view opponent student's profile button is clicked
    Private Sub btnOppViewProfile_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnOppViewProfile.Click
        'If the logged-in student is the red player...
        If LoggedInStudent.C4Player = "Red" Then
            'Sets viewed profile to opponent student's
            Viewing = 2
            'Shows the student account form
            frmStudentAccount.Show()
        Else
            'Sets viewed profile to logged-in student's
            Viewing = 1
            'Shows the student account form
            frmStudentAccount.Show()
        End If
    End Sub

End Class
```

frmNoughtsandCrossesMenu.vb

```
Public Class frmNoughtsandCrossesMenu

    'Subroutine runs when the hotseat game button is clicked
    Private Sub btnHotseat_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnHotseat.Click
        'Shows the Noughts and Crosses hotseat login form
        frmNaCHotseatLogin.Show()
        'Closes this form
        Me.Close()
    End Sub

    'Subroutine runs when the back button is clicked
    Private Sub btnBack_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnBack.Click
        'Shows the student home form
        frmStudentHome.Show()
        'Closes this form
```

```
        Me.Close()
    End Sub

End Class
```

frmNaCHotseatLogin.vb

```
Public Class frmNaCHotseatLogin

    'Declares the variables used to log in
    Dim EnteredUsername, EnteredPassword As String

    'Subroutine runs when the form loads
    Private Sub frmNaCHotseatLogin_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        'Runs the OpenDB subroutine
        OpenDB()
    End Sub

    'Subroutine runs when the okay button is clicked
    Private Sub btnOK_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnOK.Click
        'Runs the Login subroutine
        Login()
    End Sub

    'Subroutine runs when called in btnOK_Click subroutine
    Sub Login()
        'Sets the EnteredUsername and EnteredPassword variables to the entered
username and password
        EnteredUsername = txtUsername.Text
        EnteredPassword = txtPassword.Text

        'Declares the Recordset used to view records in the database
        Dim RSStaff As New ADODB.Recordset
        If RSStaff.State = ADODB.ObjectStateEnum.adStateOpen Then RSStaff.Close()

        'Builds SQL query to execute
        RSStaff.Open("SELECT * FROM tblstudents WHERE Username='" & EnteredUsername &
"' AND Password='" & EnteredPassword & "'", DBConn, ,
ADODB.LockTypeEnum.adLockOptimistic)

        'If results are found...
        If RSStaff.RecordCount > 0 Then
            'Fills the various properties of the OppStudent object with their
respective values from the database
            OppStudent.Fname = RSStaff.Fields("Fname").Value
            OppStudent.Lname = RSStaff.Fields("Lname").Value
            OppStudent.Form = RSStaff.Fields("FormNum").Value &
RSStaff.Fields("FormLetter").Value
            OppStudent.Wins = RSStaff.Fields("Wins").Value
            OppStudent.Losses = RSStaff.Fields("Losses").Value
            OppStudent.Draws = RSStaff.Fields("Draws").Value
            OppStudent.Username = RSStaff.Fields("Username").Value
            OppStudent.StudentID = RSStaff.Fields("StudentID").Value

            'Opens the Noughts and Crosses subject selection form
            frmNaCHotseatSubject.Show()
            'Closes this form
            Me.Close()
            'However if no results are found...
```

```
        Else
            'If the login details were invalid an error message will appear and the
username and password textboxes will be blanked out
            MsgBox("Invalid: Incorrect username or password.")
            txtUsername.Text = ""
            txtPassword.Text = ""
        End If
    End Sub

    'Subroutine runs when the cancel button is clicked
    Private Sub btnCancel_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnCancel.Click
        'Opens the Noughts and Crosses menu form
        frmNoughtsandCrossesMenu.Show()
        'Closes this form
        Me.Close()
    End Sub

End Class
```

frmNaCHotseatSubject.vb

```
Public Class frmNaCHotseatSubject

    'Declares the variable used to store the chosen SubjectID
    Dim SubjectID As Integer

    'Subroutine runs on form load
    Private Sub frmNaCHotseatSubject_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        'Runs the OpenDB subroutine
        OpenDB()

        'Runs the Populate subroutine
        Populate()
    End Sub

    'Subroutine runs then the okay button is clicked
    Private Sub btnOK_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnOK.Click
        'Sets the chosen subject, difficulty and topic variables
        NaCSubject = SubjectID
        NaCDifficulty = cmboDifficulty.SelectedItem
        NaCTopic = cmboTopic.SelectedItem

        'Shows the Noughts and Crosses hotseat player selection form
        frmNaCHotseatPlayerSelect.Show()
        'Closes this form
        Me.Close()
    End Sub

    'Subroutine runs when called at form load
    Sub Populate()
        'Declares the Recordset used to view records in the database
        Dim RSStaff As New ADODB.Recordset
        If RSStaff.State = ADODB.ObjectStateEnum.adStateOpen Then RSStaff.Close()

        'Builds SQL query to execute
        RSStaff.Open("SELECT * FROM tblsubjects", DBConn, ,
ADODB.LockTypeEnum.adLockOptimistic)
```

```
        'Whilst not at the end of the data...
        While Not RSStaff.EOF
            'Add the subject to the combobox
            cmboSubject.Items.Add(RSStaff.Fields("Subject").Value)
            'Move on to the next record
            RSStaff.MoveNext()
        End While

        'Closes the recordset
        RSStaff.Close()
    End Sub

    'Subroutine runs when the selected item of the subject combobox is changed
    Private Sub cmboSubject_SelectedIndexChanged(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles cmboSubject.SelectedIndexChanged
        'Declares the StreamReader used to read the topic text file
        Dim Reader As System.IO.StreamReader

        'Enables the topic combobox and clears it of any data
        cmboTopic.Enabled = True
        cmboTopic.Items.Clear()

        'Sets the path to where the file is
        Reader = My.Computer.FileSystem.OpenTextFileReader("G:\Computing Group\Y10
2009-10\Ben's\Edutainment Suite\Connect4\bin\Debug\Topics\" & cmboSubject.SelectedItem
& "Topics.txt")
        'Whilst not at the end of the text file...
        While Not Reader.EndOfStream
            'Add the topic to the combobox
            cmboTopic.Items.Add(Reader.ReadLine)
        End While

        'Closes the streamreader
        Reader.Close()

        'Declares the Recordset used to view records in the database
        Dim RSStaff As New ADODB.Recordset
        If RSStaff.State = ADODB.ObjectStateEnum.adStateOpen Then RSStaff.Close()

        'Builds SQL query to execute
        RSStaff.Open("SELECT * FROM tblsubjects WHERE Subject='" &
cmboSubject.SelectedItem & "'", DBConn, , ADODB.LockTypeEnum.adLockOptimistic)
        'Sets SubjectID to that of the selected subject
        SubjectID = RSStaff.Fields("SubjectID").Value
        'Closes the recordset
        RSStaff.Close()
    End Sub

    'Subroutine runs when the cancel button is clicked
    Private Sub btnCancel_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnCancel.Click
        'Shows the Noughts and Crosses hotseat login form
        frmNaCHotseatLogin.Show()
        'Closes this form
        Me.Close()
    End Sub

End Class
```

frmNaCHotseatPlayerSelect.vb

```
Public Class frmNaCHotseatPlayerSelect

    'Subroutine runs on form load
    Private Sub frmNACHotseatPlayerSelect_Load(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles MyBase.Load
        'Populates the player name labels with data
        lblLoggedInStudent.Text = LoggedInStudent.Fname & " " & LoggedInStudent.Lname
        lblOppStudent.Text = OppStudent.Fname & " " & OppStudent.Lname
    End Sub

    'Subroutines run when the counter selection buttons are clicked
    Private Sub btnOSX_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnOSX.Click
        'Sets the opponent student's counter to X
        OppStudent.NaCPlayer = "X"
        'Updates the appearance of the form
        picOSX.Image = My.Resources.x
        'Disables changing the counter and the logged-in student also picking X
        btnOSX.Enabled = False
        btnOSO.Enabled = False
        btnLISX.Enabled = False
        With picX
            .Image = Nothing
            .Tag = "None"
        End With
        'Runs the CheckBoth subroutine
        CheckBoth()
    End Sub
    Private Sub btnOSO_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnOSO.Click
        OppStudent.NaCPlayer = "O"
        picOSO.Image = My.Resources.o
        btnOSO.Enabled = False
        btnOSX.Enabled = False
        btnLISO.Enabled = False
        With picO
            .Image = Nothing
            .Tag = "None"
        End With
        CheckBoth()
    End Sub
    Private Sub btnLISX_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnLISX.Click
        LoggedInStudent.NaCPlayer = "X"
        picLISX.Image = My.Resources.x
        btnLISX.Enabled = False
        btnLISO.Enabled = False
        btnOSX.Enabled = False
        With picX
            .Image = Nothing
            .Tag = "None"
        End With
        CheckBoth()
    End Sub
    Private Sub btnLISO_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnLISO.Click
        LoggedInStudent.NaCPlayer = "O"
        picLISO.Image = My.Resources.o
        btnLISO.Enabled = False
        btnLISX.Enabled = False
        btnOSO.Enabled = False
        With picO
```

```
            .Image = Nothing
            .Tag = "None"
        End With
        CheckBoth()
    End Sub

    'Subroutine runs when called in the counter selection button click subroutines
    Sub CheckBoth()
        'If both players have chosen...
        If picO.Tag = "None" And picX.Tag = "None" Then
            'Enables the button to leave the form
            btnContinue.Enabled = True
        End If
    End Sub

    'Subroutine runs when the continue button is clicked
    Private Sub btnContinue_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnContinue.Click
        'Shows the Noughts and Crosses hotseat form
        frmNaCHotseat.Show()
        'Closes this form
        Me.Close()
    End Sub

    'Subroutine runs when the cancel button is clicked
    Private Sub btnCancel_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnCancel.Click
        'Shows the Noughts and Crosses hotseat question selection form
        frmNaCHotseatSubject.Show()
        'Closes this form
        Me.Close()
    End Sub

End Class
```

frmNaCHotseat.vb

```
Public Class frmNaCHotseat

    'Declares the array containing the coordinates of the squares of the grid and what
condition they are currently in
    Dim theGrid(9) As String

    'Declares the class used for both players
    Public Class NaCPlay
        'Declares the variables used for storing the details of the Connect Four
player
        Public Letter, Username, Fname, lname As String
    End Class
    'Declares the class used for both players' scores
    Public Class Score
        'Declares the variable used for storing score of the player
        Public ScoreNum As Integer
        'This subroutine runs when a player wins a game
        Public Sub Increase()
            'Increases the score by 1
            ScoreNum = ScoreNum + 1
        End Sub
    End Class
    'Creates two objects of the score class, one for each player
    Dim XScore As New Score
```

```
    Dim OScore As New Score
    'Creates two objects of the C4Play class, one for each player
    Dim XPlayer As New NaCPlay
    Dim OPlayer As New NaCPlay

    'Declares the variable used to store the primary key of the question record in the
database
    Dim QuestionID As Integer
    'Declares the y-coords for moving the current player label
    Dim StudentCurrLocationY As Integer = 35
    Dim OppStudentCurrLocationY As Integer = 140
    'Declares the variable used to determine if a question was answered correctly
    Dim QCorrect As Boolean = False

    'Subroutine runs on form load
    Private Sub frmNaCHotseat_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        'Runs the OpenDB subroutine
        OpenDB()

        'Hides the reset button
        btnReset.Visible = False

        'If the logged-in student is X...
        If LoggedInStudent.NaCPlayer = "X" Then
            'Populate the XPlayer object with the logged-in student's details
            With XPlayer
                .Letter = "X"
                .Username = LoggedInStudent.Username
                .Fname = LoggedInStudent.Fname
                .lname = LoggedInStudent.Lname
            End With

            'Populate the OPlayer object with the opponent student's details
            With OPlayer
                .Letter = "O"
                .Username = OppStudent.Username
                .Fname = OppStudent.Fname
                .lname = OppStudent.Lname
            End With
            'But if the logged-in student is yellow...
        Else
            'Populate the XPlayer object with the logged-in student's details
            With XPlayer
                .Letter = "X"
                .Username = OppStudent.Username
                .Fname = OppStudent.Fname
                .lname = OppStudent.Lname
            End With

            'Populate the OPlayer object with the opponent student's details
            With OPlayer
                .Letter = "O"
                .Username = LoggedInStudent.Username
                .Fname = LoggedInStudent.Fname
                .lname = LoggedInStudent.Lname
            End With
        End If

        'Runs the AccountSection subroutine
        AccountSection()
```

```
        'Sets the scores to the defaults
        XScore.ScoreNum = 0
        OScore.ScoreNum = 0

        'Sets the current player, runs the ChangePlayer and CurrPlayer subroutines
        NaCPlayer = "O"
        ChangePlayer()
        CurrPlayer()
    End Sub

    'Subroutines run when their respective buttons are clicked
    Private Sub btntopleft_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btntopleft.Click
        'Sets the chosen square to the current player
        btntopleft.Text = NaCPlayer
        theGrid(1) = NaCPlayer

        'Runs the ChangePlayer subroutine
        ChangePlayer()
        'Disables the button
        btntopleft.Enabled = False
        'Runs the CheckForAWinner subroutine
        CheckForAWinner()
    End Sub
    Private Sub btntop_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btntop.Click
        btntop.Text = NaCPlayer
        theGrid(2) = NaCPlayer

        ChangePlayer()
        btntop.Enabled = False
        CheckForAWinner()
    End Sub
    Private Sub btntopright_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btntopright.Click
        btntopright.Text = NaCPlayer
        theGrid(3) = NaCPlayer

        ChangePlayer()
        btntopright.Enabled = False
        CheckForAWinner()
    End Sub
    Private Sub btnleft_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnleft.Click
        btnleft.Text = NaCPlayer
        theGrid(4) = NaCPlayer

        ChangePlayer()
        btnleft.Enabled = False
        CheckForAWinner()
    End Sub
    Private Sub btnmiddle_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnmiddle.Click
        btnmiddle.Text = NaCPlayer
        theGrid(5) = NaCPlayer

        ChangePlayer()
        btnmiddle.Enabled = False
        CheckForAWinner()
    End Sub
    Private Sub btnright_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnright.Click
```

```
        btnright.Text = NaCPlayer
        theGrid(6) = NaCPlayer

        ChangePlayer()
        btnright.Enabled = False
        CheckForAWinner()
    End Sub
    Private Sub btnbottomleft_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnbottomleft.Click
        btnbottomleft.Text = NaCPlayer
        theGrid(7) = NaCPlayer

        ChangePlayer()
        btnbottomleft.Enabled = False
        CheckForAWinner()
    End Sub
    Private Sub btnbottom_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnbottom.Click
        btnbottom.Text = NaCPlayer
        theGrid(8) = NaCPlayer

        ChangePlayer()
        btnbottom.Enabled = False
        CheckForAWinner()
    End Sub
    Private Sub btnbottomright_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnbottomright.Click
        btnbottomright.Text = NaCPlayer
        theGrid(9) = NaCPlayer

        ChangePlayer()
        btnbottomright.Enabled = False
        CheckForAWinner()
    End Sub

    'Subroutine runs when called in button click subroutines
    Private Sub CheckForAWinner()
        'Declares the variable used to store the winner
        Dim WhoWon As String = ""

        'Checks top row to see if there are three Xs or Os in a row
        If theGrid(1) = theGrid(2) And theGrid(2) = theGrid(3) And theGrid(1) <> ""
Then
            'Declares the player who has claimed the three squares the winner
            WhoWon = theGrid(1)
            If WhoWon = "X" Then
                XScore.Increase()
            Else
                OScore.Increase()
            End If
            DatabaseDetails()

            'Disables grid
            btntopleft.Enabled = False
            btntop.Enabled = False
            btntopright.Enabled = False
            btnleft.Enabled = False
            btnmiddle.Enabled = False
            btnright.Enabled = False
            btnbottomleft.Enabled = False
            btnbottom.Enabled = False
            btnbottomright.Enabled = False
```

```
        End If

        'Checks middle row to see if there are three Xs or Os in a row
        If theGrid(4) = theGrid(5) And theGrid(5) = theGrid(6) And theGrid(4) <> ""
Then
            'Declares the player who has claimed the three squares the winner
            WhoWon = theGrid(4)
            If WhoWon = "X" Then
                XScore.Increase()
            Else
                OScore.Increase()
            End If
            DatabaseDetails()

            'Disables grid
            btntopleft.Enabled = False
            btntop.Enabled = False
            btntopright.Enabled = False
            btnleft.Enabled = False
            btnmiddle.Enabled = False
            btnright.Enabled = False
            btnbottomleft.Enabled = False
            btnbottom.Enabled = False
            btnbottomright.Enabled = False
        End If

        'Checks bottom row to see if there are three Xs or Os in a row
        If theGrid(7) = theGrid(8) And theGrid(8) = theGrid(9) And theGrid(7) <> ""
Then
            'Declares the player who has claimed the three squares the winner
            WhoWon = theGrid(7)
            If WhoWon = "X" Then
                XScore.Increase()
            Else
                OScore.Increase()
            End If
            DatabaseDetails()

            'Disables grid
            btntopleft.Enabled = False
            btntop.Enabled = False
            btntopright.Enabled = False
            btnleft.Enabled = False
            btnmiddle.Enabled = False
            btnright.Enabled = False
            btnbottomleft.Enabled = False
            btnbottom.Enabled = False
            btnbottomright.Enabled = False
        End If

        'Checks left column to see if there are three Xs or Os in a row
        If theGrid(1) = theGrid(4) And theGrid(4) = theGrid(7) And theGrid(1) <> ""
Then
            'Declares the player who has claimed the three squares the winner
            WhoWon = theGrid(1)
            If WhoWon = "X" Then
                XScore.Increase()
            Else
                OScore.Increase()
            End If
            DatabaseDetails()
```

```
            'Disables grid
            btntopleft.Enabled = False
            btntop.Enabled = False
            btntopright.Enabled = False
            btnleft.Enabled = False
            btnmiddle.Enabled = False
            btnright.Enabled = False
            btnbottomleft.Enabled = False
            btnbottom.Enabled = False
            btnbottomright.Enabled = False
        End If

        'Checks middle column to see if there are three Xs or Os in a row
        If theGrid(2) = theGrid(5) And theGrid(5) = theGrid(8) And theGrid(2) <> ""
Then
            'Declares the player who has claimed the three squares the winner
            WhoWon = theGrid(2)
            If WhoWon = "X" Then
                XScore.Increase()
            Else
                OScore.Increase()
            End If
            DatabaseDetails()

            'Disables grid
            btntopleft.Enabled = False
            btntop.Enabled = False
            btntopright.Enabled = False
            btnleft.Enabled = False
            btnmiddle.Enabled = False
            btnright.Enabled = False
            btnbottomleft.Enabled = False
            btnbottom.Enabled = False
            btnbottomright.Enabled = False
        End If

        'Checks right column to see if there are three Xs or Os in a row
        If theGrid(3) = theGrid(6) And theGrid(6) = theGrid(9) And theGrid(3) <> ""
Then
            'Declares the player who has claimed the three squares the winner
            WhoWon = theGrid(3)
            If WhoWon = "X" Then
                XScore.Increase()
            Else
                OScore.Increase()
            End If
            DatabaseDetails()

            'Disables grid
            btntopleft.Enabled = False
            btntop.Enabled = False
            btntopright.Enabled = False
            btnleft.Enabled = False
            btnmiddle.Enabled = False
            btnright.Enabled = False
            btnbottomleft.Enabled = False
            btnbottom.Enabled = False
            btnbottomright.Enabled = False
        End If

        'Checks top-left to bottom-right diagonal to see if there are three Xs or Os
in a row
```

```
        If theGrid(1) = theGrid(5) And theGrid(5) = theGrid(9) And theGrid(1) <> ""
Then
            'Declares the player who has claimed the three squares the winner
            WhoWon = theGrid(1)
            If WhoWon = "X" Then
                XScore.Increase()
            Else
                OScore.Increase()
            End If
            DatabaseDetails()

            'Disables grid
            btntopleft.Enabled = False
            btntop.Enabled = False
            btntopright.Enabled = False
            btnleft.Enabled = False
            btnmiddle.Enabled = False
            btnright.Enabled = False
            btnbottomleft.Enabled = False
            btnbottom.Enabled = False
            btnbottomright.Enabled = False
        End If

        'Checks top-right to bottom-left diagonal to see if there are three Xs or Os
in a row
        If theGrid(3) = theGrid(5) And theGrid(5) = theGrid(7) And theGrid(3) <> ""
Then
            'Declares the player who has claimed the three squares the winner
            WhoWon = theGrid(3)
            If WhoWon = "X" Then
                XScore.Increase()
            Else
                OScore.Increase()
            End If
            DatabaseDetails()

            'Disables grid
            btntopleft.Enabled = False
            btntop.Enabled = False
            btntopright.Enabled = False
            btnleft.Enabled = False
            btnmiddle.Enabled = False
            btnright.Enabled = False
            btnbottomleft.Enabled = False
            btnbottom.Enabled = False
            btnbottomright.Enabled = False
        End If

        'Declares the variables used to count the number of blank squares
        Dim n As Integer
        Dim Blanks As Integer

        'For each square...
        For n = 1 To 9
            'If the square isn't blank, increment blanks
            If theGrid(n) = "" Then Blanks = Blanks + 1
        Next

        'If there is a winner...
        If WhoWon <> "" Then
            'Makes buttons invisible
            btnReset.Visible = True
```

```
            lblQuestion.Visible = False
            lblCurrPlayer.Visible = False
            grpQuestion.Visible = False
            txtAnswer.Visible = False
            btnSubmit.Visible = False
            'Declares winner
            MsgBox(WhoWon & " wins.")
            'Stops game
            Exit Sub
        End If

        'If there a draw...
        If Blanks = 0 Then
            'Declares the Recordset used to add new records to the database
            Dim RSNewRec As New ADODB.Recordset
            If RSNewRec.State = ADODB.ObjectStateEnum.adStateOpen Then
RSNewRec.Close()
            'Declares the StreamWriter used to write to the game breakdown text files
            Dim writer As System.IO.StreamWriter

            'Declares draw
            MsgBox("Draw")
            'Makes buttons invisible
            btnReset.Visible = True
            grpQuestion.Visible = False
            lblQuestion.Visible = False
            txtAnswer.Visible = False
            btnSubmit.Visible = False
            lblCurrPlayer.Visible = False

            'Opens a recordset
            RSNewRec.Open("SELECT * FROM tblstudents WHERE Username='" &
LoggedInStudent.Username & "'", DBConn, , ADODB.LockTypeEnum.adLockOptimistic)
            'Adds a draw to the player's record
            With RSNewRec
                .Update()
                .Fields("Draws").Value = .Fields("Draws").Value + 1
                RSNewRec.Update()
            End With
            'Closes the recordset
            RSNewRec.Close()

            'Opens a recordset
            RSNewRec.Open("SELECT * FROM tblstudents WHERE Username='" &
OppStudent.Username & "'", DBConn, , ADODB.LockTypeEnum.adLockOptimistic)
            'Adds a draw to the other player's record
            With RSNewRec
                .Update()
                .Fields("Draws").Value = .Fields("Draws").Value + 1
                RSNewRec.Update()
            End With
            'Closes the recordset
            RSNewRec.Close()

            'Sets the path to where the log shall be generated, and the filename
            writer = My.Computer.FileSystem.OpenTextFileWriter("G:\Computing Group\Y10
2009-10\Ben's\Edutainment Suite\Connect4\bin\Debug\WinsLosses\" & XPlayer.Username &
".txt", True)

            'Writes the relevant data to the log
```

```
            writer.WriteLine(XPlayer.Fname & " " & XPlayer.lname & " drew with " &
OPlayer.Fname & " " & OPlayer.lname & " in Noughts and Crosses - " & TimeOfDay & " " &
DateValue(Now))
            'Saves the log file
            writer.Close()

            'Sets the path to where the log shall be generated, and the filename
            writer = My.Computer.FileSystem.OpenTextFileWriter("G:\Computing Group\Y10
2009-10\Ben's\Edutainment Suite\Connect4\bin\Debug\WinsLosses\" & OPlayer.Username &
".txt", True)

            'Writes the relevant data to the log
            writer.WriteLine(OPlayer.Fname & " " & OPlayer.lname & " drew with " &
XPlayer.Fname & " " & XPlayer.lname & " in Noughts and Crosses - " & TimeOfDay & " " &
DateValue(Now))
            'Saves the log file
            writer.Close()
        End If
    End Sub

    'Subroutine runs when called in winner detection subroutine
    Sub DatabaseDetails()
        'Declares the Recordset used to add new records to the database
        Dim RSNewRec As New ADODB.Recordset
        If RSNewRec.State = ADODB.ObjectStateEnum.adStateOpen Then RSNewRec.Close()
        'Declares the StreamWriter used to write to the game breakdown text files
        Dim writer As System.IO.StreamWriter

        'If X is the winner...
        If XScore.ScoreNum = 1 Then
            'Opens a recordset
            RSNewRec.Open("SELECT * FROM tblstudents WHERE Username='" &
XPlayer.Username & "'", DBConn, , ADODB.LockTypeEnum.adLockOptimistic)
            'Adds a win to the winning player's record
            With RSNewRec
                .Update()
                .Fields("Wins").Value = .Fields("Wins").Value + 1
                RSNewRec.Update()
            End With
            'Closes the recordset
            RSNewRec.Close()

            'Opens a recordset
            RSNewRec.Open("SELECT * FROM tblstudents WHERE Username='" &
OPlayer.Username & "'", DBConn, , ADODB.LockTypeEnum.adLockOptimistic)
            'Adds a loss to the losing player's record
            With RSNewRec
                .Update()
                .Fields("Losses").Value = .Fields("Losses").Value + 1
                RSNewRec.Update()
            End With
            'Closes the recordset
            RSNewRec.Close()

            'Sets the path to where the log shall be generated, and the filename
            writer = My.Computer.FileSystem.OpenTextFileWriter("G:\Computing Group\Y10
2009-10\Ben's\Edutainment Suite\Connect4\bin\Debug\WinsLosses\" & XPlayer.Username &
".txt", True)

            'Writes the relevant data to the log
```

```
            writer.WriteLine(XPlayer.Fname & " " & XPlayer.lname & " beat " &
OPlayer.Fname & " " & OPlayer.lname & " in Noughts and Crosses - " & TimeOfDay & " " &
DateValue(Now))
            'Saves the log file
            writer.Close()

            'Sets the path to where the log shall be generated, and the filename
            writer = My.Computer.FileSystem.OpenTextFileWriter("G:\Computing Group\Y10
2009-10\Ben's\Edutainment Suite\Connect4\bin\Debug\WinsLosses\" & OPlayer.Username &
".txt", True)

            'Writes the relevant data to the log
            writer.WriteLine(OPlayer.Fname & " " & OPlayer.lname & " was beaten by " &
XPlayer.Fname & " " & XPlayer.lname & " in Noughts and Crosses - " & TimeOfDay & " " &
DateValue(Now))
            'Saves the log file
            writer.Close()

            'Resets the X score back to its default
            XScore.ScoreNum = 0
        End If

        'However, if the O player is the winner...
        If OScore.ScoreNum = 1 Then
            'Opens a recordset
            RSNewRec.Open("SELECT * FROM tblstudents WHERE Username='" &
OPlayer.Username & "'", DBConn, , ADODB.LockTypeEnum.adLockOptimistic)
            'Adds a win to the winning player's record
            With RSNewRec
                .Update()
                .Fields("Wins").Value = .Fields("Wins").Value + 1
                RSNewRec.Update()
            End With
            'Closes the recordset
            RSNewRec.Close()

            'Opens a recordset
            RSNewRec.Open("SELECT * FROM tblstudents WHERE Username='" &
XPlayer.Username & "'", DBConn, , ADODB.LockTypeEnum.adLockOptimistic)
            'Adds a loss to the losing player's record
            With RSNewRec
                .Update()
                .Fields("Losses").Value = .Fields("Losses").Value + 1
                RSNewRec.Update()
            End With
            'Closes the recordset
            RSNewRec.Close()

            'Sets the path to where the log shall be generated, and the filename
            writer = My.Computer.FileSystem.OpenTextFileWriter("G:\Computing Group\Y10
2009-10\Ben's\Edutainment Suite\Connect4\bin\Debug\WinsLosses\" & OPlayer.Username &
".txt", True)

            'Writes the relevant data to the log
            writer.WriteLine(OPlayer.Fname & " " & OPlayer.lname & " beat " &
XPlayer.Fname & " " & XPlayer.lname & " in Noughts and Crosses - " & TimeOfDay & " " &
DateValue(Now))
            'Saves the log file
            writer.Close()

            'Sets the path to where the log shall be generated, and the filename
```

```
            writer = My.Computer.FileSystem.OpenTextFileWriter("G:\Computing Group\Y10
2009-10\Ben's\Edutainment Suite\Connect4\bin\Debug\WinsLosses\" & XPlayer.Username &
".txt", True)

            'Writes the relevant data to the log
            writer.WriteLine(XPlayer.Fname & " " & XPlayer.lname & " was beaten by " &
OPlayer.Fname & " " & OPlayer.lname & " in Noughts and Crosses - " & TimeOfDay & " " &
DateValue(Now))
            'Saves the log file
            writer.Close()

            'Resets the O score back to its default
            OScore.ScoreNum = 0
        End If

        grpQuestion.Visible = False

        btnReset.Location = New Point(331, 206)
        btnReset.Visible = True
    End Sub

    'Subroutine runs when called in the ChangePlayer subroutine
    Sub CurrPlayer()
        'If the logged-in player is the same as the current player, which is X...
        If LoggedInStudent.NaCPlayer = "X" Then
            If NaCPlayer = "X" Then
                'Places the current player label pointing to the logged-in student
                lblCurrPlayer.Top = StudentCurrLocationY
                'However if the current player is O...
            Else
                'Places the current player label pointing to the opponent student
                lblCurrPlayer.Top = OppStudentCurrLocationY
            End If
            'However, if the opponent student is the same as the current player, which
is O...
        Else
            If NaCPlayer = "O" Then
                'Places the current player label pointing to the logged-in student
                lblCurrPlayer.Top = StudentCurrLocationY
                'However if the current player is yellow...
            Else
                'Places the current player label pointing to the opponent student
                lblCurrPlayer.Top = OppStudentCurrLocationY
            End If
        End If
    End Sub

    'Subroutine runs when called in form load subroutine
    Sub AccountSection()
        'Populates the player name labels with data and get a picture of each player
        lblStudentName.Text = LoggedInStudent.Fname & " " & LoggedInStudent.Lname
        picStudent.ImageLocation = "G:\Computing Group\Y10 2009-10\Ben's\Edutainment
Suite\Connect4\bin\Debug\images\students\" & LoggedInStudent.Username & ".jpg"

        lblOppStudentName.Text = OppStudent.Fname & " " & OppStudent.Lname
        picOppStudentPic.ImageLocation = "G:\Computing Group\Y10 2009-
10\Ben's\Edutainment Suite\Connect4\bin\Debug\images\students\" & OppStudent.Username
& ".jpg"
    End Sub

    'Subroutine runs when reset button is clicked
```

```
    Private Sub btnReset_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnReset.Click
        'This re-enables all of the buttons, removes the Xs and 0s from them as well
as resetting the array
        btntopleft.Enabled = True
        btntop.Enabled = True
        btntopright.Enabled = True
        btnleft.Enabled = True
        btnmiddle.Enabled = True
        btnright.Enabled = True
        btnbottomleft.Enabled = True
        btnbottom.Enabled = True
        btnbottomright.Enabled = True

        btntopleft.Text = ""
        btntop.Text = ""
        btntopright.Text = ""
        btnleft.Text = ""
        btnmiddle.Text = ""
        btnright.Text = ""
        btnbottomleft.Text = ""
        btnbottom.Text = ""
        btnbottomright.Text = ""

        Dim n As Integer

        For n = 1 To 9
            theGrid(n) = ""
        Next

        'Resets the current player
        NaCPlayer = "X"
        CurrPlayer()

        'Runs the Question subroutine
        Question()

        'Makes the controls visible
        lblCurrPlayer.Visible = True
        grpQuestion.Visible = True
        lblQuestion.Visible = True
        txtAnswer.Visible = True
        btnSubmit.Visible = True
        btnReset.Visible = False
        btnReset.Location = New Point(19, 90)
    End Sub

    'Subroutine runs when back button is clicked
    Private Sub btnBack_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnBack.Click
        'Shows the Noughts and Crosses menu
        frmNoughtsandCrossesMenu.Show()
        'Closes this form
        Me.Close()
    End Sub

    'Subroutine runs when called in ChangePlayer and btnReset_Click subroutines
    Sub Question()
        'Declares the Recordset used to view records in the database
        Dim RSStaff As New ADODB.Recordset
        If RSStaff.State = ADODB.ObjectStateEnum.adStateOpen Then RSStaff.Close()
```

```
        'Builds SQL query to execute
        RSStaff.Open("SELECT * FROM tblquestions WHERE SubjectID='" & NaCSubject & "'
AND Difficulty='" & NaCDifficulty & "' AND Topic='" & NaCTopic & "' ORDER BY RAND()
LIMIT 1", DBConn, , ADODB.LockTypeEnum.adLockOptimistic)

        'Sets the QuestionID variable to that of the selected question
        QuestionID = RSStaff.Fields("QuestionID").Value

        'Makes the question controls visible
        grpQuestion.Visible = True
        lblQuestion.Visible = True
        txtAnswer.Visible = True
        btnSubmit.Visible = True

        'Displays the selected question
        lblQuestion.Text = RSStaff.Fields("Question").Value
        'Closes the recordset
        RSStaff.Close()
    End Sub

    'Subroutine runs when called in form load and button click subroutines
    Sub ChangePlayer()
        'Changes the current player
        If NaCPlayer = "X" Then
            NaCPlayer = "O"
            CurrPlayer()
        Else
            NaCPlayer = "X"
            CurrPlayer()
        End If

        'Hides the buttons
        btntopleft.Enabled = False
        btntop.Enabled = False
        btntopright.Enabled = False
        btnleft.Enabled = False
        btnmiddle.Enabled = False
        btnright.Enabled = False
        btnbottomleft.Enabled = False
        btnbottom.Enabled = False
        btnbottomright.Enabled = False

        'Runs the Question subroutine
        Question()
    End Sub

    'Subroutine runs then answer submit button is clicked
    Private Sub btnSubmit_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnSubmit.Click
        'Declares the Recordset used to view records in the database
        Dim RSStaff As New ADODB.Recordset
        If RSStaff.State = ADODB.ObjectStateEnum.adStateOpen Then RSStaff.Close()

        'Hides the question controls
        grpQuestion.Visible = False
        lblQuestion.Visible = False
        txtAnswer.Visible = False
        btnSubmit.Visible = False

        'Builds SQL query to execute
        RSStaff.Open("SELECT * FROM tblquestions WHERE QuestionID='" & QuestionID &
"'", DBConn, , ADODB.LockTypeEnum.adLockOptimistic)
```

```
        'If the answer is correct...
        If txtAnswer.Text = RSStaff.Fields("Answer").Value Then
            'Display a message box
            MsgBox("Correct!")

            'Sets the question correct flag to true
            QCorrect = True

            'Makes the buttons visible
            btntopleft.Enabled = True
            btntop.Enabled = True
            btntopright.Enabled = True
            btnleft.Enabled = True
            btnmiddle.Enabled = True
            btnright.Enabled = True
            btnbottomleft.Enabled = True
            btnbottom.Enabled = True
            btnbottomright.Enabled = True

            'Runs the QDatabase subroutine
            QDatabase()
            'However if the answer if incorrect...
        Else
            'Display a message box
            MsgBox("Incorrect!")

            'Sets the question correct flag to false
            QCorrect = False

            'Runs the QDatabase subroutine
            QDatabase()
            'Runs the ChangePlayer subroutine
            ChangePlayer()
        End If

        'Blanks the answer textbox for the next question
        txtAnswer.Text = ""
    End Sub

    'Subroutine runs when called in btnSubmit_Click subroutine
    Sub QDatabase()
        'Declares the Recordset used to add new records to the database
        Dim RSNewRec As New ADODB.Recordset
        If RSNewRec.State = ADODB.ObjectStateEnum.adStateOpen Then RSNewRec.Close()
        'Opens a recordset
        RSNewRec.Open("SELECT * FROM tblattempted", DBConn, ,
ADODB.LockTypeEnum.adLockOptimistic)
        'Adds the new record
        With RSNewRec
            .AddNew()
            .Fields("SubjectID").Value = NaCSubject
            .Fields("QuestionID").Value = QuestionID
            If NaCPlayer = "X" Then
                If XPlayer.Username = LoggedInStudent.Username Then
                    .Fields("StudentID").Value = LoggedInStudent.StudentID
                Else
                    .Fields("StudentID").Value = OppStudent.StudentID
                End If
            Else
                If OPlayer.Username = LoggedInStudent.Username Then
                    .Fields("StudentID").Value = LoggedInStudent.StudentID
```

```
                Else
                    .Fields("StudentID").Value = OppStudent.StudentID
                End If
            End If
            .Fields("When").Value = TimeOfDay & " " & DateValue(Now)
            If QCorrect = True Then
                .Fields("Correct").Value = 1
            Else
                .Fields("Correct").Value = 0
            End If

            'Resets the question correct flag
            QCorrect = False

            'Updates the recordset
            RSNewRec.Update()
            'Closes the recordset
            RSNewRec.Close()
        End With
    End Sub

    'Subroutine runs when view logged-in student's profile button is clicked
    Private Sub btnViewProfile_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnViewProfile.Click
        'Sets viewed profile to logged-in student's
        Viewing = 1
        'Shows the student account form
        frmStudentAccount.Show()
    End Sub

    'Subroutine runs when view opponent student's profile button is clicked
    Private Sub btnOppViewProfile_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnOppViewProfile.Click
        'Sets viewed profile to opponent student's
        Viewing = 2
        'Shows the student account form
        frmStudentAccount.Show()
    End Sub

End Class
```

frmRockPaperScissorsMenu.vb

```
Public Class frmRockPaperScissorsMenu

    'Subroutine runs when the hotseat game button is clicked
    Private Sub btnHotseat_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnHotseat.Click
        'Shows the Rock, Paper, Scissors hotseat login form
        frmRPSHotseatLogin.Show()
        'Closes this form
        Me.Close()
    End Sub

    'Subroutine runs when the back button is clicked
    Private Sub btnBack_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnBack.Click
        'Shows the student home form
        frmStudentHome.Show()
        'Closes this form
        Me.Close()
```

```
    End Sub

End Class
```

frmRPSHotseatLogin.vb

```
Public Class frmRPSHotseatLogin

    'Declares the variables used to log in
    Dim EnteredUsername, EnteredPassword As String

    'Subroutine runs when the form loads
    Private Sub frmRPSHotseatLogin_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        'Runs the OpenDB subroutine
        OpenDB()
    End Sub

    'Subroutine runs when the okay button is clicked
    Private Sub btnOK_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnOK.Click
        'Runs the Login subroutine
        Login()
    End Sub

    'Subroutine runs when called in btnOK_Click subroutine
    Sub Login()
        'Sets the EnteredUsername and EnteredPassword variables to the entered
username and password
        EnteredUsername = txtUsername.Text
        EnteredPassword = txtPassword.Text

        'Declares the Recordset used to view records in the database
        Dim RSStaff As New ADODB.Recordset
        If RSStaff.State = ADODB.ObjectStateEnum.adStateOpen Then RSStaff.Close()

        'Builds SQL query to execute
        RSStaff.Open("SELECT * FROM tblstudents WHERE Username='" & EnteredUsername &
"' AND Password='" & EnteredPassword & "'", DBConn, ,
ADODB.LockTypeEnum.adLockOptimistic)

        'If results are found...
        If RSStaff.RecordCount > 0 Then
            'Fills the various properties of the OppStudent object with their
respective values from the database
            OppStudent.Fname = RSStaff.Fields("Fname").Value
            OppStudent.Lname = RSStaff.Fields("Lname").Value
            OppStudent.Form = RSStaff.Fields("FormNum").Value &
RSStaff.Fields("FormLetter").Value
            OppStudent.Wins = RSStaff.Fields("Wins").Value
            OppStudent.Losses = RSStaff.Fields("Losses").Value
            OppStudent.Draws = RSStaff.Fields("Draws").Value
            OppStudent.Username = RSStaff.Fields("Username").Value
            OppStudent.StudentID = RSStaff.Fields("StudentID").Value

            'Opens the Rock, Paper, Scissors subject selection form
            frmRPSHotseatSubject.Show()
            'Closes this form
            Me.Close()
            'However if no results are found...
        Else
```

```
            'If the login details were invalid an error message will appear and the
username and password textboxes will be blanked out
            MsgBox("Invalid: Incorrect username or password.")
            txtUsername.Text = ""
            txtPassword.Text = ""
        End If
    End Sub

    'Subroutine runs when the cancel button is clicked
    Private Sub btnCancel_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnCancel.Click
        'Opens the Rock, Paper, Scissors menu form
        frmRockPaperScissorsMenu.Show()
        'Closes this form
        Me.Close()
    End Sub

End Class
```

frmRPSHotseatSubject.vb

```
Public Class frmRPSHotseatSubject
    'Declares the variable used to store the chosen SubjectID
    Dim SubjectID As Integer

    'Subroutine runs on form load
    Private Sub frmNaCHotseatSubject_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        'Runs the OpenDB subroutine
        OpenDB()

        'Runs the Populate subroutine
        Populate()
    End Sub

    'Subroutine runs then the okay button is clicked
    Private Sub btnOK_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnOK.Click
        'Sets the chosen subject, difficulty and topic variables
        RPSSubject = SubjectID
        RPSDifficulty = cmboDifficulty.SelectedItem
        RPSTopic = cmboTopic.SelectedItem

        'Shows the Rock, Paper, Scissors hotseat game form
        frmRPSHotseat.Show()
        'Closes this form
        Me.Close()
    End Sub

    'Subroutine runs when called at form load
    Sub Populate()
        'Declares the Recordset used to view records in the database
        Dim RSStaff As New ADODB.Recordset
        If RSStaff.State = ADODB.ObjectStateEnum.adStateOpen Then RSStaff.Close()

        'Builds SQL query to execute
        RSStaff.Open("SELECT * FROM tblsubjects", DBConn, ,
ADODB.LockTypeEnum.adLockOptimistic)

        'Whilst not at the end of the data...
        While Not RSStaff.EOF
```

```
            'Add the subject to the combobox
            cmboSubject.Items.Add(RSStaff.Fields("Subject").Value)
            'Move on to the next record
            RSStaff.MoveNext()
        End While

        'Closes the recordset
        RSStaff.Close()
    End Sub

    'Subroutine runs when the selected item of the subject combobox is changed
    Private Sub cmboSubject_SelectedIndexChanged(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles cmboSubject.SelectedIndexChanged
        'Declares the StreamReader used to read the topic text file
        Dim Reader As System.IO.StreamReader

        'Enables the topic combobox and clears it of any data
        cmboTopic.Enabled = True
        cmboTopic.Items.Clear()

        'Sets the path to where the file is
        Reader = My.Computer.FileSystem.OpenTextFileReader("G:\Computing Group\Y10
2009-10\Ben's\Edutainment Suite\Connect4\bin\Debug\Topics\" & cmboSubject.SelectedItem
& "Topics.txt")
        'Whilst not at the end of the text file...
        While Not Reader.EndOfStream
            'Add the topic to the combobox
            cmboTopic.Items.Add(Reader.ReadLine)
        End While

        'Closes the streamreader
        Reader.Close()

        'Declares the Recordset used to view records in the database
        Dim RSStaff As New ADODB.Recordset
        If RSStaff.State = ADODB.ObjectStateEnum.adStateOpen Then RSStaff.Close()

        'Builds SQL query to execute
        RSStaff.Open("SELECT * FROM tblsubjects WHERE Subject='" &
cmboSubject.SelectedItem & "'", DBConn, , ADODB.LockTypeEnum.adLockOptimistic)
        'Sets SubjectID to that of the selected subject
        SubjectID = RSStaff.Fields("SubjectID").Value
        'Closes the recordset
        RSStaff.Close()
    End Sub

    'Subroutine runs when the cancel button is clicked
    Private Sub btnCancel_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnCancel.Click
        'Shows the Rock, Paper, Scissors hotseat login form
        frmRPSHotseatLogin.Show()
        'Closes this form
        Me.Close()
    End Sub

End Class
```

frmRPSHotseat.vb

```
Public Class frmRPSHotseatLogin
```

```
    'Declares the variables used to log in
    Dim EnteredUsername, EnteredPassword As String

    'Subroutine runs when the form loads
    Private Sub frmRPSHotseatLogin_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        'Runs the OpenDB subroutine
        OpenDB()
    End Sub

    'Subroutine runs when the okay button is clicked
    Private Sub btnOK_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnOK.Click
        'Runs the Login subroutine
        Login()
    End Sub

    'Subroutine runs when called in btnOK_Click subroutine
    Sub Login()
        'Sets the EnteredUsername and EnteredPassword variables to the entered
username and password
        EnteredUsername = txtUsername.Text
        EnteredPassword = txtPassword.Text

        'Declares the Recordset used to view records in the database
        Dim RSStaff As New ADODB.Recordset
        If RSStaff.State = ADODB.ObjectStateEnum.adStateOpen Then RSStaff.Close()

        'Builds SQL query to execute
        RSStaff.Open("SELECT * FROM tblstudents WHERE Username='" & EnteredUsername &
"' AND Password='" & EnteredPassword & "'", DBConn, ,
ADODB.LockTypeEnum.adLockOptimistic)

        'If results are found...
        If RSStaff.RecordCount > 0 Then
            'Fills the various properties of the OppStudent object with their
respective values from the database
            OppStudent.Fname = RSStaff.Fields("Fname").Value
            OppStudent.Lname = RSStaff.Fields("Lname").Value
            OppStudent.Form = RSStaff.Fields("FormNum").Value &
RSStaff.Fields("FormLetter").Value
            OppStudent.Wins = RSStaff.Fields("Wins").Value
            OppStudent.Losses = RSStaff.Fields("Losses").Value
            OppStudent.Draws = RSStaff.Fields("Draws").Value
            OppStudent.Username = RSStaff.Fields("Username").Value
            OppStudent.StudentID = RSStaff.Fields("StudentID").Value

            'Opens the Rock, Paper, Scissors subject selection form
            frmRPSHotseatSubject.Show()
            'Closes this form
            Me.Close()
            'However if no results are found...
        Else
            'If the login details were invalid an error message will appear and the
username and password textboxes will be blanked out
            MsgBox("Invalid: Incorrect username or password.")
            txtUsername.Text = ""
            txtPassword.Text = ""
        End If
    End Sub

    'Subroutine runs when the cancel button is clicked
```

```
    Private Sub btnCancel_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnCancel.Click
        'Opens the Rock, Paper, Scissors menu form
        frmRockPaperScissorsMenu.Show()
        'Closes this form
        Me.Close()
    End Sub

End Class
```

frmRPSFight.vb

```
Public Class frmRPSFight

    'Declares the variables for the timers
    Dim count As Integer = 100
    Dim countdown As Integer = 3
    Dim time As Integer = 0

    'Declares the variables used for storing the winner
    Dim WinnerName As String
    Dim Winner As Integer

    'Subroutine runs on form load
    Private Sub frmRPSFight_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load

        'Runs the AccountSection subroutine
        AccountSection()

        'Runs the Weapons subroutine
        Weapons()

        'Enables the countdown timer
        tmrCountdown.Enabled = True
    End Sub

    'Subroutine runs when called in form load subroutine
    Sub Weapons()
        'Sets the images to the players' chosen weapons
        If LoggedInWeapon = 1 Then
            picLoggedIn.Image = My.Resources.bigrock
        ElseIf LoggedInWeapon = 2 Then
            picLoggedIn.Image = My.Resources.bigpaper
        ElseIf LoggedInWeapon = 3 Then
            picLoggedIn.Image = My.Resources.bigscissorsloggedin
        ElseIf LoggedInWeapon = 0 Then
            picLoggedIn.Image = My.Resources.bigchickenloggedin
        End If

        If OppWeapon = 1 Then
            picOpp.Image = My.Resources.bigrock
        ElseIf OppWeapon = 2 Then
            picOpp.Image = My.Resources.bigpaper
        ElseIf OppWeapon = 3 Then
            picOpp.Image = My.Resources.bigscissorsopp
        ElseIf OppWeapon = 0 Then
            picOpp.Image = My.Resources.bigchickenopp
        End If
    End Sub
```

```
    'Subroutine runs when called in form load subroutine
    Sub AccountSection()
        'Populates the player name labels with data
        lblLoggedInName.Text = LoggedInStudent.Fname & " " & LoggedInStudent.Lname
        lblOppName.Text = OppStudent.Fname & " " & OppStudent.Lname
    End Sub

    'Subroutine runs when countdown timer ticks
    Private Sub tmrCountdown_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles tmrCountdown.Tick
        'Decrements count by 10
        count = count - 10

        'If count reaches 0...
        If count = 0 Then
            'Changes the number of the countdown displayed
            countdown = countdown - 1
            If countdown > 0 Then
                lblCountdown.Text = countdown
                count = 100
                'If the coundown reaches the end...
            Else
                'Disables the timer, displays the weapons
                picLoggedIn.Visible = True
                picOpp.Visible = True
                lblCountdown.Visible = False
                count = 1000
                tmrCountdown.Enabled = False
                tmrAnimate.Enabled = True

                'Plays 3 Inches of Blood - Deady Sinners
                My.Computer.Audio.Play("G:\Computing Group\Y10 2009-
10\Ben's\Edutainment Suite\Connect4\Resources\deadly.wav")
            End If
        End If
    End Sub

    'Subroutine runs when animation timer ticks
    Private Sub tmrAnimate_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles tmrAnimate.Tick
        'Declares the variables used for the coords of the pictures
        Dim x, y, xx, yy As Integer

        'Decrements count by 1000
        count = count - 1000

        'Runs the random number generator subroutine
        Randomize()

        'Gets the random coords of both pictureboxes
        x = CInt(Int((140) * Rnd() - 20))
        y = CInt(Int((50) * Rnd()) + 60)
        xx = CInt(Int((140) * Rnd()) + 390)
        yy = CInt(Int((50) * Rnd()) + 60)

        'Changes the position of both pictureboxes fast enough to resemble animation
        If count = 0 Then
            If time <> 40 Then
                count = 1000
                time = time + 1
                picLoggedIn.Location = New Point(x, y)
                picOpp.Location = New Point(xx, yy)
```

```
            'If the time has come for the animation to stop...
        Else
            'Stops playing the music
            My.Computer.Audio.Stop()
            'Runs the CheckWinner subroutine
            CheckWinner()
            'Disables the timer
            tmrAnimate.Enabled = False
            'Runs the DatabaseDetails subroutine
            DatabaseDetails()
            'Runs the ChangeImage subroutine
            ChangeImage()
            'Declares the winner
            MsgBox("Winner: " & WinnerName)
            'Shows the Rock, Paper, Scissors hotseat game form
            frmRPSHotseat.Reset()
            'Closes this form
            Me.Close()
        End If
    End If
End Sub

'Subroutine runs when called in tmrAnimate_Tick subroutine
Sub ChangeImage()
    'Changes the images of the players to reflect the result of the fight
    If Winner = 1 Then
        ‘If the opponent user has paper...
        If OppWeapon <> 0 Then
            ‘If the logged-in user has rock...
            If LoggedInWeapon = 1 Then
                ‘Change the images appropriately
                picLoggedIn.Image = My.Resources.bigrockwinner
                picOpp.Image = My.Resources.bigscissorsdeadopp
            ElseIf LoggedInWeapon = 2 Then
                picLoggedIn.Image = My.Resources.bigpaperwinner
                picOpp.Image = My.Resources.bigrockdead
            ElseIf LoggedInWeapon = 3 Then
                picLoggedIn.Image = My.Resources.bigscissorswinloggedin
                picOpp.Image = My.Resources.bigpaperdead
            End If
        Else
            If LoggedInWeapon = 1 Then
                picLoggedIn.Image = My.Resources.bigrockwinner
                picOpp.Image = My.Resources.bigchickendeadopp
            ElseIf LoggedInWeapon = 2 Then
                picLoggedIn.Image = My.Resources.bigpaperwinner
                picOpp.Image = My.Resources.bigchickendeadopp
            ElseIf LoggedInWeapon = 3 Then
                picLoggedIn.Image = My.Resources.bigscissorswinloggedin
                picOpp.Image = My.Resources.bigchickendeadopp
            End If
        End If
        If LoggedInWeapon <> 0 Then
            If LoggedInWeapon = 1 Then
                picLoggedIn.Image = My.Resources.bigrockdead
                picOpp.Image = My.Resources.bigpaperwinner
            ElseIf LoggedInWeapon = 2 Then
                picLoggedIn.Image = My.Resources.bigpaperdead
                picOpp.Image = My.Resources.bigscissorswinopp
            ElseIf LoggedInWeapon = 3 Then
                picLoggedIn.Image = My.Resources.bigscissorsdeadloggedin
                picOpp.Image = My.Resources.bigrockwinner
```

```
            End If
        Else
            If LoggedInWeapon = 1 Then
                picLoggedIn.Image = My.Resources.bigchickendeadloggedin
                picOpp.Image = My.Resources.bigpaperwinner
            ElseIf LoggedInWeapon = 2 Then
                picLoggedIn.Image = My.Resources.bigchickendeadloggedin
                picOpp.Image = My.Resources.bigscissorswinopp
            ElseIf LoggedInWeapon = 3 Then
                picLoggedIn.Image = My.Resources.bigchickendeadloggedin
                picOpp.Image = My.Resources.bigrockwinner
            End If
        End If
    ElseIf Winner = 2 Then
        If OppWeapon <> 0 Then
            If LoggedInWeapon = 1 Then
                picLoggedIn.Image = My.Resources.bigrockdead
                picOpp.Image = My.Resources.bigpaperwinner
            ElseIf LoggedInWeapon = 2 Then
                picLoggedIn.Image = My.Resources.bigpaperdead
                picOpp.Image = My.Resources.bigscissorswinopp
            ElseIf LoggedInWeapon = 3 Then
                picLoggedIn.Image = My.Resources.bigscissorsdeadloggedin
                picOpp.Image = My.Resources.bigrockwinner
            End If
        Else
            If LoggedInWeapon = 1 Then
                picLoggedIn.Image = My.Resources.bigrockwinner
                picOpp.Image = My.Resources.bigchickendeadopp
            ElseIf LoggedInWeapon = 2 Then
                picLoggedIn.Image = My.Resources.bigpaperwinner
                picOpp.Image = My.Resources.bigchickendeadopp
            ElseIf LoggedInWeapon = 3 Then
                picLoggedIn.Image = My.Resources.bigscissorswinloggedin
                picOpp.Image = My.Resources.bigchickendeadopp
            End If
        End If
        If LoggedInWeapon <> 0 Then
            If LoggedInWeapon = 1 Then
                picLoggedIn.Image = My.Resources.bigrockdead
                picOpp.Image = My.Resources.bigpaperwinner
            ElseIf LoggedInWeapon = 2 Then
                picLoggedIn.Image = My.Resources.bigpaperdead
                picOpp.Image = My.Resources.bigscissorswinopp
            ElseIf LoggedInWeapon = 3 Then
                picLoggedIn.Image = My.Resources.bigscissorsdeadloggedin
                picOpp.Image = My.Resources.bigrockwinner
            End If
        Else
            If LoggedInWeapon = 1 Then
                picLoggedIn.Image = My.Resources.bigchickendeadloggedin
                picOpp.Image = My.Resources.bigpaperwinner
            ElseIf LoggedInWeapon = 2 Then
                picLoggedIn.Image = My.Resources.bigchickendeadloggedin
                picOpp.Image = My.Resources.bigscissorswinopp
            ElseIf LoggedInWeapon = 3 Then
                picLoggedIn.Image = My.Resources.bigchickendeadloggedin
                picOpp.Image = My.Resources.bigrockwinner
            End If
        End If
    End If
End Sub
```

```
    'Subroutine runs when called in tmrAnimate_Tick subroutine
    Sub CheckWinner()
        'Determines the winner of the game
        If (LoggedInWeapon = 2 And OppWeapon = 1) Or (LoggedInWeapon = 3 And OppWeapon
= 2) Or (LoggedInWeapon = 1 And OppWeapon = 3) Or (LoggedInWeapon <> 0 And OppWeapon =
0) Then
            WinnerName = LoggedInStudent.Fname & " " & LoggedInStudent.Lname
            Winner = 1
        End If
        If (LoggedInWeapon = 1 And OppWeapon = 2) Or (LoggedInWeapon = 2 And OppWeapon
= 3) Or (LoggedInWeapon = 3 And OppWeapon = 1) Or (LoggedInWeapon = 0 And OppWeapon <>
0) Then
            WinnerName = OppStudent.Fname & " " & OppStudent.Lname
            Winner = 2
        End If
        If LoggedInWeapon = OppWeapon Then
            WinnerName = "No-one, it's a draw"
            Winner = 0
        End If
    End Sub

    'Subroutine runs when called in winner detection subroutine
    Sub DatabaseDetails()
        'Declares the Recordset used to add new records to the database
        Dim RSNewRec As New ADODB.Recordset
        If RSNewRec.State = ADODB.ObjectStateEnum.adStateOpen Then RSNewRec.Close()
        'Declares the StreamWriter used to write to the game breakdown text files
        Dim writer As System.IO.StreamWriter

        If Winner = 1 Then
            'Opens a recordset
            RSNewRec.Open("SELECT * FROM tblstudents WHERE Username='" &
LoggedInStudent.Username & "'", DBConn, , ADODB.LockTypeEnum.adLockOptimistic)
            'Adds a win to the winning player's record
            With RSNewRec
                .Update()
                .Fields("Wins").Value = .Fields("Wins").Value + 1
                RSNewRec.Update()
            End With
            'Closes the recordset
            RSNewRec.Close()

            'Opens a recordset
            RSNewRec.Open("SELECT * FROM tblstudents WHERE Username='" &
OppStudent.Username & "'", DBConn, , ADODB.LockTypeEnum.adLockOptimistic)
            'Adds a loss to the losing player's record
            With RSNewRec
                .Update()
                .Fields("Losses").Value = .Fields("Losses").Value + 1
                RSNewRec.Update()
            End With
            'Closes the recordset
            RSNewRec.Close()

            'Sets the path to where the log shall be generated, and the filename
            writer = My.Computer.FileSystem.OpenTextFileWriter("G:\Computing Group\Y10
2009-10\Ben's\Edutainment Suite\Connect4\bin\Debug\WinsLosses\" &
LoggedInStudent.Username & ".txt", True)

            'Writes the relevant data to the log
```

```
            writer.WriteLine(LoggedInStudent.Fname & " " & LoggedInStudent.Lname & "
beat " & OppStudent.Fname & " " & OppStudent.Lname & " in Rock, Paper, Scissors - " &
TimeOfDay & " " & DateValue(Now))
            'Saves the log file
            writer.Close()

            'Sets the path to where the log shall be generated, and the filename
            writer = My.Computer.FileSystem.OpenTextFileWriter("G:\Computing Group\Y10
2009-10\Ben's\Edutainment Suite\Connect4\bin\Debug\WinsLosses\" & OppStudent.Username
& ".txt", True)

            'Writes the relevant data to the log
            writer.WriteLine(LoggedInStudent.Fname & " " & LoggedInStudent.Lname & "
beat " & OppStudent.Fname & " " & OppStudent.Lname & " in Rock, Paper, Scissors - " &
TimeOfDay & " " & DateValue(Now))
            'Saves the log file
            writer.Close()
        ElseIf Winner = 2 Then
            'Opens a recordset
            RSNewRec.Open("SELECT * FROM tblstudents WHERE Username='" &
OppStudent.Username & "'", DBConn, , ADODB.LockTypeEnum.adLockOptimistic)
            'Adds a win to the winning player's record
            With RSNewRec
                .Update()
                .Fields("Wins").Value = .Fields("Wins").Value + 1
                RSNewRec.Update()
            End With
            'Closes the recordset
            RSNewRec.Close()

            'Opens a recordset
            RSNewRec.Open("SELECT * FROM tblstudents WHERE Username='" &
LoggedInStudent.Username & "'", DBConn, , ADODB.LockTypeEnum.adLockOptimistic)
            'Adds a loss to the losing player's record
            With RSNewRec
                .Update()
                .Fields("Losses").Value = .Fields("Losses").Value + 1
                RSNewRec.Update()
            End With
            'Closes the recordset
            RSNewRec.Close()

            'Sets the path to where the log shall be generated, and the filename
            writer = My.Computer.FileSystem.OpenTextFileWriter("G:\Computing Group\Y10
2009-10\Ben's\Edutainment Suite\Connect4\bin\Debug\WinsLosses\" &
LoggedInStudent.Username & ".txt", True)

            'Writes the relevant data to the log
            writer.WriteLine(OppStudent.Fname & " " & OppStudent.Lname & " beat " &
LoggedInStudent.Fname & " " & LoggedInStudent.Lname & " in Rock, Paper, Scissors - " &
TimeOfDay & " " & DateValue(Now))
            'Saves the log file
            writer.Close()

            'Sets the path to where the log shall be generated, and the filename
            writer = My.Computer.FileSystem.OpenTextFileWriter("G:\Computing Group\Y10
2009-10\Ben's\Edutainment Suite\Connect4\bin\Debug\WinsLosses\" & OppStudent.Username
& ".txt", True)

            'Writes the relevant data to the log
```

```
            writer.WriteLine(OppStudent.Fname & " " & OppStudent.Lname & " beat " &
LoggedInStudent.Fname & " " & LoggedInStudent.Lname & " in Rock, Paper, Scissors - " &
TimeOfDay & " " & DateValue(Now))
            'Saves the log file
            writer.Close()
        ElseIf Winner = 0 Then
            'Opens a recordset
            RSNewRec.Open("SELECT * FROM tblstudents WHERE Username='" &
LoggedInStudent.Username & "'", DBConn, , ADODB.LockTypeEnum.adLockOptimistic)
            'Adds a draw to the player's record
            With RSNewRec
                .Update()
                .Fields("Draws").Value = .Fields("Draws").Value + 1
                RSNewRec.Update()
            End With
            'Closes the recordset
            RSNewRec.Close()

            'Opens a recordset
            RSNewRec.Open("SELECT * FROM tblstudents WHERE Username='" &
OppStudent.Username & "'", DBConn, , ADODB.LockTypeEnum.adLockOptimistic)
            'Adds a draw to the player's record
            With RSNewRec
                .Update()
                .Fields("Draws").Value = .Fields("Draws").Value + 1
                RSNewRec.Update()
            End With
            'Closes the recordset
            RSNewRec.Close()

            'Sets the path to where the log shall be generated, and the filename
            writer = My.Computer.FileSystem.OpenTextFileWriter("G:\Computing Group\Y10
2009-10\Ben's\Edutainment Suite\Connect4\bin\Debug\WinsLosses\" &
LoggedInStudent.Username & ".txt", True)

            'Writes the relevant data to the log
            writer.WriteLine(LoggedInStudent.Fname & " " & LoggedInStudent.Lname & "
drew with " & OppStudent.Fname & " " & OppStudent.Lname & " in Rock, Paper, Scissors -
" & TimeOfDay & " " & DateValue(Now))
            'Saves the log file
            writer.Close()

            'Sets the path to where the log shall be generated, and the filename
            writer = My.Computer.FileSystem.OpenTextFileWriter("G:\Computing Group\Y10
2009-10\Ben's\Edutainment Suite\Connect4\bin\Debug\WinsLosses\" & OppStudent.Username
& ".txt", True)

            'Writes the relevant data to the log
            writer.WriteLine(LoggedInStudent.Fname & " " & LoggedInStudent.Lname & "
drew with " & OppStudent.Fname & " " & OppStudent.Lname & " in Rock, Paper, Scissors -
" & TimeOfDay & " " & DateValue(Now))
            'Saves the log file
            writer.Close()
        End If
    End Sub
End Class
```

frmTeacherHome.vb

```
Public Class frmTeacherHome
```

```
    'Subroutine runs when the form loads
    Private Sub frmTeacherHome_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        'Runs the account section population subroutine
        AccountSection()
    End Sub

    'Subroutine runs when called in the form load sub
    Sub AccountSection()
        'Places the logged-in teacher's name onto the form
        lblTeacherName.Text = LoggedInTeacher.Fname & " " & LoggedInTeacher.Lname
        'Places the logged-in teacher's picture onto the form
        picTeacher.ImageLocation = "G:\Computing Group\Y10 2009-10\Ben's\Edutainment
Suite\Connect4\bin\Debug\images\teachers\" & LoggedInTeacher.Username & ".jpg"
    End Sub

    'Subroutine runs when the view profile button is clicked
    Private Sub btnViewProfile_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnViewProfile.Click
        'Opens the teacher account form
        frmTeacherAccount.Show()
    End Sub

    'Subroutine runs when the view student button is clicked
    Private Sub btnViewStudent_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnViewStudent.Click
        'Opens the view student form
        frmViewStudent.Show()
        'Closes this form
        Me.Close()
    End Sub

    'Subroutine runs when the create question button is clicked
    Private Sub btnCreateQuestion_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnCreateQuestion.Click
        'Opens the add question form
        frmAddQuestion.Show()
    End Sub

    'Subroutine runs when the add student button is clicked
    Private Sub btnAddStudent_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnAddStudent.Click
        'Opens the add student form
        frmAddStudent.Show()
    End Sub

End Class
```

frmTeacherAccount.vb

```
Public Class frmTeacherAccount

    'Subroutine runs when the form loads
    Private Sub frmAccount_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        'Establishes the connection to the database
        OpenDB()

        'Runs the account section population subroutine
        AccountSection()
    End Sub
```

```
    'Subroutine runs when the form loads
    Sub AccountSection()
        'Places the logged-in teacher's name onto the form
        lblTeacherName.Text = LoggedInTeacher.Fname & " " & LoggedInTeacher.Lname
        'Places the logged-in teacher's picture onto the form
        picTeacher.ImageLocation = "G:\Computing Group\Y10 2009-10\Ben's\Edutainment
Suite\Connect4\bin\Debug\images\teachers\" & LoggedInTeacher.Username & ".jpg"

        'Declares the variable used for reading data from the database
        Dim RSStaff As New ADODB.Recordset
        If RSStaff.State = ADODB.ObjectStateEnum.adStateOpen Then RSStaff.Close()

        'Builds SQL query to execute
        RSStaff.Open("SELECT * FROM `tblquestions` WHERE `TeacherID`='" &
LoggedInTeacher.TeacherID & "'", DBConn, , ADODB.LockTypeEnum.adLockOptimistic)
        'Populates the questions created label
        lblQsCreated.Text = "Questions Created: " & RSStaff.RecordCount
        'Closes the recordset
        RSStaff.Close()
    End Sub

    'Subroutine runs when the view breakdown button is clicked
    Private Sub btnBreakdown_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnBreakdown.Click
        'Shows the teacher breakdown form
        frmTeacherBreakdown.Show()
    End Sub

    'Subroutine runs when the view achievements button is clicked
    Private Sub btnAchievements_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnAchievements.Click
        'Shows the teacher achievements form
        frmTeacherAchievements.Show()
    End Sub

End Class
```

frmTeacherBreakdown.vb

```
Public Class frmTeacherBreakdown

    'Subroutine runs when the form loads
    Private Sub frmTeacherBreakdown_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        'Declares the variable used to read the teacher breakdown text file
        Dim Reader As System.IO.StreamReader

        'Places the logged-in teacher's name onto the form
        lblBreakdown.Text = LoggedInTeacher.Fname & " " & LoggedInTeacher.Lname & "
Question Breakdown"

        'Gets the path to the logged-in teacher's breakdown text file
        Reader = My.Computer.FileSystem.OpenTextFileReader("G:\Computing Group\Y10
2009-10\Ben's\Edutainment Suite\Connect4\bin\Debug\QuestionCreation\" &
LoggedInTeacher.Username & ".txt")

        'Whilst the end of the text file hasn't been reached...
        While Not Reader.EndOfStream
            'Adds a line to the breakdown listbox
            lstBreakdown.Items.Add(Reader.ReadLine)
```

```
        End While

        'Closes the recordset
        Reader.Close()
    End Sub

End Class
```

frmTeacherAchievements.vb

```
Public Class frmTeacherAchievements

    'Subroutine runs when the form loads
    Private Sub frmTeacherAchievements_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        'Establishes the connection to the database
        OpenDB()

        'Places the logged-in teacher's name onto the form
        lblAchievements.Text = LoggedInTeacher.Fname & " " & LoggedInTeacher.Lname & "
Achievements"

        'Runs the achievements population subroutine
        Achievements()
    End Sub

    'Subroutine runs when the form runs
    Sub Achievements()
        'Runs the achievment subroutines
        TeachingAssistant()
        TenuredProf()
    End Sub

    'Subroutines run when the achievement subroutine calls them
    Sub TeachingAssistant()
        'Declares the variable used for reading data from the database
        Dim RSStaff As New ADODB.Recordset
        If RSStaff.State = ADODB.ObjectStateEnum.adStateOpen Then RSStaff.Close()

        'Builds SQL query to execute
        RSStaff.Open("SELECT * FROM `tblquestions` WHERE `TeacherID`='" &
LoggedInTeacher.TeacherID & "'", DBConn, , ADODB.LockTypeEnum.adLockOptimistic)

        'If any records are found...
        If RSStaff.RecordCount > 0 Then
            'Unlocks the achievement
            lblTeachingAssistant.Text = "Teaching Assistant"
            lblTeachingAssistantDeets.Text = "Create your first question"
            picTeachingAssistant.Image = My.Resources.Assistant
        End If

        'Closes the recordset
        RSStaff.Close()
    End Sub

    Sub TenuredProf()
        'Declares the variable used for reading data from the database
        Dim RSStaff As New ADODB.Recordset
        If RSStaff.State = ADODB.ObjectStateEnum.adStateOpen Then RSStaff.Close()

        'Builds SQL query to execute
```

```
        RSStaff.Open("SELECT * FROM `tblquestions` WHERE `TeacherID`='" &
LoggedInTeacher.TeacherID & "'", DBConn, , ADODB.LockTypeEnum.adLockOptimistic)

        'If 20+ records are found...
        If RSStaff.RecordCount > 19 Then
            'Unlocks the achievement
            lblTenuredProf.Text = "Tenured Professor"
            lblTenuredProfDeets.Text = "Create 20 questions"
            picTenuredProf.Image = My.Resources.professor
        End If

        'Closes the recordset
        RSStaff.Close()
    End Sub

End Class
```

frmViewStudent.vb

```
Public Class frmViewStudent

    'Subroutine runs when the form loads
    Private Sub frmViewStudent_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        'Establishes the connection to the database
        OpenDB()

        'Runs the account section population subroutine
        AccountSection()

        'Declares the variable used for reading data from the database
        Dim RSStaff As New ADODB.Recordset
        If RSStaff.State = ADODB.ObjectStateEnum.adStateOpen Then RSStaff.Close()

        'Builds SQL query to execute
        RSStaff.Open("SELECT * FROM tblstudents", DBConn, ,
ADODB.LockTypeEnum.adLockOptimistic)
        'Whilst the end of the database hasn't been reached...
        While Not RSStaff.EOF
            'Adds a line to the students listbox
            lstStudents.Items.Add(RSStaff.Fields("StudentID").Value & " - " &
RSStaff.Fields("Fname").Value & " " & RSStaff.Fields("Lname").Value)
            'Moves onto the next record
            RSStaff.MoveNext()
        End While
    End Sub

    'Subroutine runs when called in the form loads
    Sub AccountSection()
        'Places the logged-in teacher's name onto the form
        lblTeacherName.Text = LoggedInTeacher.Fname & " " & LoggedInTeacher.Lname
        'Places the logged-in teacher's picture onto the form
        picTeacher.ImageLocation = "G:\Computing Group\Y10 2009-10\Ben's\Edutainment
Suite\Connect4\bin\Debug\images\teachers\" & LoggedInTeacher.Username & ".jpg"
    End Sub

    'Subroutine runs when the view profile button is clicked
    Private Sub btnViewProfile_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnViewProfile.Click
        'Opens teacher account form
        frmTeacherAccount.Show()
```

```
    End Sub

    'Subroutine runs when the search button is clicked
    Private Sub btnSearch_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnSearch.Click
        'Declares the variable used for reading data from the database
        Dim RSStaff As New ADODB.Recordset
        If RSStaff.State = ADODB.ObjectStateEnum.adStateOpen Then RSStaff.Close()

        'Blanks the listbox
        lstStudents.Items.Clear()

        If (txtFname.Text <> "") And (txtLname.Text <> "") Then
            'Builds SQL query to execute
            RSStaff.Open("SELECT * FROM tblstudents WHERE Fname LIKE '%" &
txtFname.Text & "%' OR Lname LIKE '%" & txtLname.Text & "%'", DBConn, ,
ADODB.LockTypeEnum.adLockOptimistic)
        ElseIf (txtFname.Text <> "") And (txtLname.Text = "") Then
            'Builds SQL query to execute
            RSStaff.Open("SELECT * FROM tblstudents WHERE Fname LIKE '%" &
txtFname.Text & "%'", DBConn, , ADODB.LockTypeEnum.adLockOptimistic)
        ElseIf (txtFname.Text = "") And (txtLname.Text <> "") Then
            'Builds SQL query to execute
            RSStaff.Open("SELECT * FROM tblstudents WHERE tblstudents WHERE Lname LIKE
'%" & txtLname.Text & "%'", DBConn, , ADODB.LockTypeEnum.adLockOptimistic)
        End If

        'If records are found...
        If RSStaff.RecordCount > 0 Then
            'Whilst the end of the database hasn't been reached...
            While Not RSStaff.EOF
                'Adds a line to the students listbox
                lstStudents.Items.Add(RSStaff.Fields("StudentID").Value & " - " &
RSStaff.Fields("Fname").Value & " " & RSStaff.Fields("Lname").Value)
                'Moves onto the next record
                RSStaff.MoveNext()
            End While
            'If no records are found...
        Else
            'Displays a message box
            MsgBox("No records found")
        End If

        'Closes the recordset
        RSStaff.Close()
    End Sub

    'Subroutine runs when the selected index of the students listbox changes
    Private Sub lstStudents_SelectedIndexChanged(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles lstStudents.SelectedIndexChanged
        'Declares the variable used for getting the end of the StudentID
        Dim EndofStudentID As Integer = InStr(1, lstStudents.SelectedItem, " ",
CompareMethod.Text)
        'Declares the variable used for storing the StudentID
        Dim StudID As String = Mid(lstStudents.SelectedItem, 1, EndofStudentID)

        'Declares the variable used for reading data from the database
        Dim RSStaff As New ADODB.Recordset
        If RSStaff.State = ADODB.ObjectStateEnum.adStateOpen Then RSStaff.Close()

        'Builds SQL query to execute
```

```
        RSStaff.Open("SELECT * FROM tblstudents WHERE StudentID='" & StudID & "'",
DBConn, , ADODB.LockTypeEnum.adLockOptimistic)

        'Fills the various properties of the LoggedInStudent object with their
respective values from the database
        With LoggedInStudent
            .Fname = RSStaff.Fields("Fname").Value
            .Lname = RSStaff.Fields("Lname").Value
            .Form = RSStaff.Fields("FormNum").Value &
RSStaff.Fields("FormLetter").Value
            .Wins = RSStaff.Fields("Wins").Value
            .Losses = RSStaff.Fields("Losses").Value
            .Draws = RSStaff.Fields("Draws").Value
            .Username = RSStaff.Fields("Username").Value
        End With

        'Closes the recordset
        RSStaff.Close()

        'Enables the view student profile button
        btnViewStudentProfile.Enabled = True
    End Sub

    'Subroutine runs when the view student button is clicked
    Private Sub btnViewStudentProfile_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnViewStudentProfile.Click
        'Opens the student account form
        frmStudentAccount.Show()
    End Sub

    'Subroutine runs when the back button is clicked
    Private Sub btnBack_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnBack.Click
        'Opens the teacher home form
        frmTeacherHome.Show()
        'Closes this form
        Me.Close()
    End Sub

End Class
```

frmAddQuestion.vb

```
Public Class frmAddQuestion

    'Declares the variable used for storing the SubjectID
    Dim SubjectID As Integer

    'Subroutine runs when the form loads
    Private Sub frmAddQuestion_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        'Establishes the connection to the database
        OpenDB()

        'Runs the form population subroutine
        Populate()
    End Sub

    'Subroutine runs when called in the form load sub
    Sub Populate()
        'Declares the variable used for reading data from the database
```

```
        Dim RSStaff As New ADODB.Recordset
        If RSStaff.State = ADODB.ObjectStateEnum.adStateOpen Then RSStaff.Close()

        'Builds SQL query to execute
        RSStaff.Open("SELECT * FROM tblsubjects", DBConn, ,
ADODB.LockTypeEnum.adLockOptimistic)

        'Whilst the end of the database hasn't been reached...
        While Not RSStaff.EOF
            'Populates the subject combobox with data
            cmboSubject.Items.Add(RSStaff.Fields("Subject").Value)
            'Moves onto the next record
            RSStaff.MoveNext()
        End While

        'Closes the recordset
        RSStaff.Close()
    End Sub

    'Subroutine runs when the back button is clicked
    Private Sub btnBack_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnBack.Click
        'Closes the form
        Me.Close()
    End Sub

    'Subroutine runs when the selected index in the subject combobox changes
    Private Sub cmboSubject_SelectedIndexChanged(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles cmboSubject.SelectedIndexChanged
        'Declares the varibale used to read the topics text file
        Dim Reader As System.IO.StreamReader

        'Enables the topics combobox
        cmboTopic.Enabled = True

        'Gets the path to the selected subject's topics text file
        Reader = My.Computer.FileSystem.OpenTextFileReader("G:\Computing Group\Y10
2009-10\Ben's\Edutainment Suite\Connect4\bin\Debug\Topics\" & cmboSubject.SelectedItem
& "Topics.txt")
        'Whilst the end of the text file hasn't been reacher...
        While Not Reader.EndOfStream
            'Adds an item to the topics combobox
            cmboTopic.Items.Add(Reader.ReadLine)
        End While

        'Closes the recordset
        Reader.Close()

        'Declares the variable used for reading data from the database
        Dim RSStaff As New ADODB.Recordset
        If RSStaff.State = ADODB.ObjectStateEnum.adStateOpen Then RSStaff.Close()

        'Builds SQL query to execute
        RSStaff.Open("SELECT * FROM tblsubjects WHERE Subject='" &
cmboSubject.SelectedItem & "'", DBConn, , ADODB.LockTypeEnum.adLockOptimistic)

        'Sets 'SubjectID' as the select subject's SubjectID from the database
        SubjectID = RSStaff.Fields("SubjectID").Value

        'Closes the recordset
        RSStaff.Close()
    End Sub
```

```
    'Subroutine runs when the question textbox is clicked
    Private Sub txtQuestion_Click(ByVal sender As Object, ByVal e As System.EventArgs)
Handles txtQuestion.Click
        'Blanks out the textbox
        txtQuestion.Text = ""
    End Sub

    'Subroutine runs when the answer textbox is clicked
    Private Sub txtAnswer_Click(ByVal sender As Object, ByVal e As System.EventArgs)
Handles txtAnswer.Click
        'Blanks out the textbox
        txtAnswer.Text = ""
    End Sub

    'Subroutine runs when the submit button is clicked
    Private Sub btnSubmit_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnSubmit.Click
        'Declares the variable used for getting the result of the message box
        Dim Result As MsgBoxResult
        'Declares the variable used for adding a new record into the database
        Dim RSNewRec As New ADODB.Recordset
        If RSNewRec.State = ADODB.ObjectStateEnum.adStateOpen Then RSNewRec.Close()
        'Declares the variable used for detecting invalid data entry
        Dim Errors As String = ""

        'Assembles an error report if any invalid data entry detected
        If cmboSubject.Text = "" Then
            Errors = Errors & "No subject selected" & vbCrLf
        End If
        If cmboDifficulty.Text = "" Then
            Errors = Errors & "No difficulty selected" & vbCrLf
        End If
        If cmboTopic.Text = "" Then
            Errors = Errors & "No topic selected" & vbCrLf
        End If
        If txtQuestion.Text = "" Then
            Errors = Errors & "No question input" & vbCrLf
        End If
        If txtAnswer.Text = "" Then
            Errors = Errors & "No answer input" & vbCrLf
        End If

        'If no invalid data entry is detected...
        If Errors = "" Then
            'Displays a validation message box before saving the data to the database
            Result = MsgBox("Are you sure all these details are correct? Remember,
spelling is vital." & vbCrLf & vbCrLf & "Details:" & vbCrLf &
cmboDifficulty.SelectedItem & " " & cmboSubject.SelectedItem & vbCrLf &
cmboTopic.SelectedItem & vbCrLf & vbCrLf & "Question: '" & txtQuestion.Text & "'" &
vbCrLf & "Answer: '" & txtAnswer.Text & "'", MsgBoxStyle.YesNo)
            'If the data is approved by the user...
            If Result = MsgBoxResult.Yes Then
                'Builds SQL query to execute
                RSNewRec.Open("SELECT * FROM tblquestions", DBConn, ,
ADODB.LockTypeEnum.adLockOptimistic)
                'Adds a new record and fills it with information before closing
                With RSNewRec
                    .AddNew()
                    .Fields("TeacherID").Value = LoggedInTeacher.TeacherID
                    .Fields("SubjectID").Value = SubjectID
                    .Fields("Question").Value = txtQuestion.Text
```

```
                    .Fields("Answer").Value = txtAnswer.Text
                    .Fields("Difficulty").Value = cmboDifficulty.SelectedItem
                    .Fields("Topic").Value = cmboTopic.SelectedItem
                    RSNewRec.Update()
                End With
            End If
            'If any invalid data entry is detected...
        Else
            'Displays a message box with any detected invalid data entry
            MsgBox("Invalid input:" & vbCrLf & vbCrLf & Errors)
        End If

        'Declares the variable used for writing to the text file
        Dim writer As System.IO.StreamWriter
        'Gets the filepath to the logged-in teacher's question creation log text file
        writer = My.Computer.FileSystem.OpenTextFileWriter("G:\Computing Group\Y10
2009-10\Ben's\Edutainment Suite\Connect4\bin\Debug\QuestionCreation\" &
LoggedInTeacher.Username & ".txt", True)
        'Adds the log of the question creation to the logged-in teacher's log file
        writer.WriteLine(LoggedInTeacher.Fname & " " & LoggedInTeacher.Lname & "
created a question for " & cmboDifficulty.SelectedItem & " " &
cmboSubject.SelectedItem & " ('" & txtQuestion.Text & "') - " & TimeOfDay & " " &
DateValue(Now))
        'Saves the log file
        writer.Close()
    End Sub

End Class
```

Procedures and Subroutines

mdlPublicVars.vb

OpenDB

Establishes a connection with the MySQL database used throughout the program

frmSplash.vb

frmSplash_Load

Loads the splash screen and gets everything going

tmrLoading_Tick

Simulates the loading

frmFTSMsg.vb

btnYes_Click

Proceeds on with the first-time setup

btnNo_Click

Cancels the first-time setup

frmFirstTimeSetup.vb

btnAddSubject_Click

Loads up the add subject form

btnAddTopic_Click

Loads up the add topic form

btnAddTeacher_Click

Loads up the add teacher form

btnAddStudent_Click

Loads up the add student form

btnDone_Click

Completes the first-time setup and loads up the login form_

frmAddSubject.vb

frmAddSubject_Load

Loads the form, connects to database

btnAddSubject_Click

Adds the subject record to the database

txtSubjectName_Click

Blanks the textbox

frmAddTopic.vb

frmAddTopic_Load

Loads the form, connects to database

btnAddTopic_Click

Runs the subroutine to check if the data entry is valid

Populate

Populates the comboboxes on the form

CheckValid

Checks whether the data entry is valid, and if so, adds the data to the database

frmAddTeacher.vb

frmAddTeacher_Load

Loads the form, connects to database

txtTeacherFirstName_Click

Blanks the textbox

txtTeacherLastName_Click

Blanks the textbox

btnAddTeacher_Click

Checks whether the data entry is valid, and if so, adds the data to the database

Username

Creates the username based off of the entered user name

btnBrowsePic_Click

Opens a file viewer to select an image for the new user

frmAddStudent.vb

frmAddStudent_Load

Loads the form, connects to database

txtStudentFirstName_Click

Blanks the textbox

txtStudentLastName_Click

Blanks the textbox

btnAddStudent_Click

Checks whether the data entry is valid, and if so, adds the data to the database

Username

Creates the username based off of the entered user name

btnBrowsePic_Click

Opens a file viewer to select an image for the new user

frmLogin.vb

frmLogin_Load

Loads the form, connects to database

btnOK_Click

Runs the subroutine to log in

btnCancel_Click

Closes the program

DetectFTS

Checks whether the first-time setup has been run before, and if not, runs it

Login

Checks if the login details are valid, and if so, logs in

frmStudentHome.vb

frmStudentHome_Load

Runs the subroutine to populate the account section

AccountSection

Populates the account section

btnViewProfile_Click

Displays the profile form

btnConnectFour_Click

Displays the Connect Four form

btnNoughtsandCrosses_Click

Displays the Noughts and Crosses form

btnRPS_Click

Displays the Rock, Paper, Scissors form

btnConnectFour_MouseMove

Sets the displayed game description and image to Connect Four

btnNoughtsandCrosses_MouseMove

Sets the displayed game description and image to Noughts and Crosses

btnRPS_MouseMove

Sets the displayed game description and image to Rock, Paper, Scissors

frmStudentHome_MouseMove

Sets the displayed game description and image back to their defaults

lblGameDesc_MouseMove

Sets the displayed game description and image back to their defaults

frmStudentAccount.vb

frmStudentAccount_Load

Runs the subroutine to populate the account section

AccountSection

Populates the account section

btnBreakdown_Click

Displays the profile breakdown form

btnAchievements_Click

Displays the profile achievements form

frmStudentBreakdown.vb

frmStudentBreakdown_Load

Connects to the database and displays the data

frmStudentAchievements.vb

frmStudentAchievements_Load

Runs the subroutine to populate the achievements, connects to database

Achievements

Runs the achievement subroutines

FirstBlood

Determines whether or not the ‘First Blood’ achievement has been unlocked

frmConnect4Menu.vb

btnHotseat_Click

Starts a hotseat Connect Four game

btnNetwork_Click

Displays the Connect Four network game lobby

btnBack_Click

Goes back

frmConnect4HotseatLogin.vb

btnConnect4HotseatLogin_Load

Conncts to database

btnNetwork_Click

Runs the subroutine to log in

Login

Checks if the login details are valid, and if so, logs the opponent in

btnCancel_Click

Goes back

frmConnect4HotseatSubject.vb

btnConnect4HotseatSubject_Load

Connects to database, runs subroutine to populate comboboxes on the form

btnOK_Click

Sets the question parameters for the game and goes on to the next form

Populate

Populates the comboboxes on the form

cmboSubject_SelectedIndexChanged

Populates the topic combobox when a subject is selected

btnCancel_Click

Goes back

frmConnect4HotseatPlayerSelect.vb

btnConnect4HotseatPlayerSelect_Load

Displays player names

btnOSred_Click

Selects red for the opponent

btnOSyellow_Click

Selects yellow for the opponent

btnLISred_Click

Selects red for the logged-in user

btnLISyellow_Click

Selects red for the logged-in user

CheckBoth

Tests whether both players have selected counters

btnContinue_Click

Moves on to the game itself

btnCancel_Click

Goes back

frmConnect4Hotseat.vb

Increase

Increments the score of an object of the ‘Score’ class

frmConnect4Hotseat_Load

Populates the form, connects to database, sets up game for playing

DatabaseDetails

Updates the database and the users’ breakdown file after a game

CurrPlayer

Moves the currently playing label

AccountSection

Populates the account section

MakeGrid

Makes the Connect Four grid

Terminators

Adds a row of terminators to the grid

CounterPlace

Places a counter on the grid

DetectWinner

Detects a winner horizontally, vertically or diagonally, or a draw

DiagTDWin

Detects a diagonal top-left to bottom-right win

DiagBUWin

Detects a diagonal bottom-left to top-right win

btnReset_Click

Resets the game after one is finished

btn1_Click

Runs the subroutine for placing a counter in the first column

bn2_Click

Runs the subroutine for placing a counter in the second column

btn3_Click

Runs the subroutine for placing a counter in the third column

btn4_Click

Runs the subroutine for placing a counter in the fourth column

btn5_Click

Runs the subroutine for placing a counter in the fifth column

btn6_Click

Runs the subroutine for placing a counter in the sixth column

btn7_Click

Runs the subroutine for placing a counter in the seventh column

btnBack_Click

Goes back

Question

Asks a new question

ChangePlayer

Changes the current player

btnSubmit_Click

Submits the answer to the current question

QDatabase

Updates the database when a question is answered

btnViewProfile_Click

Displays the logged-in user's profile form

btnOppViewProfile_Click

Displays the opponent's profile form

frmConnect4NetworkLobby.vb

btnConnect4NetworkLobby_Load

Connects to databse, updates lobby list

UpdateList

Updates the list of ongoing games

btnHostGame_Click

Displays the host game form

btnYourGames_Click

Displays the your games form

lstLobby_SelectedIndexChanged

Gets the details of the selected game for later

btnJoinGame_Click

Joins the selected game

btnRefresh_Click

Refreshes the lobby

tmrUpdate_Tick

Refreshes the lobby

frmConnect4NetworkHostGame.vb

btnConnect4NetworkHostGame_Load

Connects to databse, populates comboboxes

btnHost_Click

Hosts the game is data entry is valid

Populate

Populates the comboboxes of the form

cmboSubject_SelectedIndexChanged

Populates the topic combobox when a subject is selected

lstLobby_SelectedIndexChanged

Gets the details of the selected game for later

btnJoinGame_Click

Joins the selected game

btnRefresh_Click

Refreshes the lobby

tmrUpdate_Tick

Refreshes the lobby

frmConnect4NetworkYourGames.vb

btnConnect4NetworkYourGames_Load

Connects to databse, updates lobby list

UpdateList

Updates the list of ongoing games

btnHostGame_Click

Displays the host game form

lstLobby_SelectedIndexChanged

Gets the details of the selected game for later

btnJoinGame_Click

Joins the selected game

btnRefresh_Click

Refreshes the lobby

frmConnect4Network.vb

Increase

Increments the score of an object of the ‘Score’ class

frmConnect4Network_Load

Populates the form, connects to database, sets up game for playing

DatabaseDetails

Updates the database and the users’ breakdown file after a game

CurrPlayer

Moves the currently playing label

AccountSection

Populates the account section

MakeGrid

Makes the Connect Four grid

NetworkCheckGrid

Updates the grid over the network

Terminators

Adds a row of terminators to the grid

CounterPlace

Places a counter on the grid

DetectWinner

Detects a winner horizontally, vertically or diagonally, or a draw

DiagTDWin

Detects a diagonal top-left to bottom-right win

DiagBUWin

Detects a diagonal bottom-left to top-right win

btnReset_Click

Resets the game after one is finished

btn1_Click

Runs the subroutine for placing a counter in the first column

bn2_Click

Runs the subroutine for placing a counter in the second column

btn3_Click

Runs the subroutine for placing a counter in the third column

btn4_Click

Runs the subroutine for placing a counter in the fourth column

btn5_Click

Runs the subroutine for placing a counter in the fifth column

btn6_Click

Runs the subroutine for placing a counter in the sixth column

btn7_Click

Runs the subroutine for placing a counter in the seventh column

btnBack_Click

Goes back

Question

Asks a new question

ChangePlayer

Changes the current player

CheckPlayer

Checks for the current player over the network

btnSubmit_Click

Submits the answer to the current question

QDatabase

Updates the database when a question is answered

btnQuit_Click

Quits the game

tmrUpdate_Tick

Checks the database to see if the other player has made any changes to the state of the game

btnViewProfile_Click

Displays the logged-in user's profile form

btnOppViewProfile_Click

Displays the opponent's profile form

frmNoughtsAndCrossesMenu.vb

btnHotseat_Click

Starts a hotseat Noughts and Crosses game

btnBack_Click

Goes back

frmNaCHotseatLogin.vb

btnNaCHotseatLogin_Load

Conncts to database

btnOK_Click

Runs the subroutine to log in

Login

Checks if the login details are valid, and if so, logs the opponent in

btnCancel_Click

Goes back

frmNaCHotseatSubject.vb

btnNaCHotseatSubject_Load

Connects to database, runs subroutine to populate comboboxes on the form

btnOK_Click

Sets the question parameters for the game and goes on to the next form

Populate

Populates the comboboxes on the form

cmboSubject_SelectedIndexChanged

Populates the topic combobox when a subject is selected

btnCancel_Click

Goes back

frmNaCHotseatPlayerSelect.vb

btnNaCHotseatPlayerSelect_Load

Displays player names

btnOSX_Click

Selects X for the opponent

btnOSO_Click

Selects O for the opponent

btnLISX_Click

Selects X for the logged-in user

btnLISO_Click

Selects O for the logged-in user

CheckBoth

Tests whether both players have selected counters

btnContinue_Click

Moves on to the game itself

btnCancel_Click

Goes back

frmNaCHotseat.vb

Increase

Increments the score of an object of the ‘Score’ class

frmNaCHotseat_Load

Populates the form, connects to database, sets up game for playing

btntopleft_Click

Runs the subroutine for placing a counter in top-left square

bntop_Click

Runs the subroutine for placing a counter in the top square

btntopright_Click

Runs the subroutine for placing a counter in the top-right square

btnleft_Click

Runs the subroutine for placing a counter in the left square

btnmiddle_Click

Runs the subroutine for placing a counter in the middle square

btnright_Click

Runs the subroutine for placing a counter in the right square

btnbottomleft_Click

Runs the subroutine for placing a counter in the bottom-left square

btnbottom_Click

Runs the subroutine for placing a counter in the bottom square

btnbottomright_Click

Runs the subroutine for placing a counter in the bottom-right square

CheckForAWinner

Detects a winner horizontally, vertically or diagonally, or a draw

DatabaseDetails

Updates the database and the users' breakdown file after a game

CurrPlayer

Moves the currently playing label

AccountSection

Populates the account section

btnReset_Click

Resets the game after one is finished

btnBack_Click

Goes back

Question

Asks a new question

ChangePlayer

Changes the current player

btnSubmit_Click

Submits the answer to the current question

QDatabase

Updates the database when a question is answered

btnViewProfile_Click

Displays the logged-in user's profile form

btnOppViewProfile_Click

Displays the opponent's profile form

frmRockPaperScissorsMenu.vb

btnHotseat_Click

Starts a hotseat Rock, Paper, Scissors game

btnBack_Click

Goes back

frmRPSHotseatLogin.vb

btnRPSHotseatLogin_Load

Conncts to database

btnOK_Click

Runs the subroutine to log in

Login

Checks if the login details are valid, and if so, logs the opponent in

btnCancel_Click

Goes back

frmRPSHotseatSubject.vb

btnRPSHotseatSubject_Load

Connects to database, runs subroutine to populate comboboxes on the form

btnOK_Click

Sets the question parameters for the game and goes on to the next form

Populate

Populates the comboboxes on the form

cmboSubject_SelectedIndexChanged

Populates the topic combobox when a subject is selected

btnCancel_Click

Goes back

frmRPSHotseat.vb

frmRPSHotseat_Load

Populates the form, connects to database, sets up game for playing

AccountSection

Populates the account section

ChangePlayer

Changes the current player

CurrPlayer

Moves the currently playing label

Question

Asks a new question

btnSubmit_Click

Submits the answer to the current question

QDatabase

Updates the database when a question is answered

btnViewProfile_Click

Displays the logged-in user's profile form

btnOppViewProfile_Click

Displays the opponent's profile form

btnBack_Click

Goes back

btnRock_Click

Selects rock as the weapon

btnPaper_Click

Selects paper as the weapon

*btnScissors*_Click

Selects scissors as the weapon

Reset

Resets the game for another match

frmRPSFight.vb

frmRPSFight_Load

Populates the form, starts the countdown timer

Weapons

Populates the players' weapons

AccountSection

Populates the account section

tmrCountDown_Tick

Counts down to the start of the fight

tmrAnimate_Tick

Animates the fight

ChangeImage

Changes the images to reflect the result of the fight

CheckWinner

Checks which player has won

DatabaseDetails

Updates the database after the game ends

btnOppViewProfile_Click

Displays the opponent's profile form

btnBack_Click

Goes back

btnRock_Click

Selects rock as the weapon

btnPaper_Click

Selects paper as the weapon

*btnScissors*_Click

Selects scissors as the weapon

Reset

Resets the game for another match

frmTeacherHome.vb

frmTeacherHome_Load

Runs the subroutine to populate the account section

AccountSection

Populates the account section

btnViewProfile_Click

Displays the profile form

btnViewStudent_Click

Displays the view student form

btnCreateQuestion_Click

Displays the create question form

btnAddStudent_Click

Displays the add student form

frmTeacherAccount.vb

frmTeacherAccount_Load

Runs the subroutine to populate the account section

AccountSection

Populates the account section

btnBreakdown_Click

Displays the profile breakdown form

btnAchievements_Click

Displays the profile achievements form

frmTeacherBreakdown.vb

frmTeacherBreakdown_Load

Connects to the database and displays the data

frmTeacherAchievements.vb

frmTeacherAchievements_Load

Runs the subroutine to populate the achievements, connects to database

Achievements

Runs the achievement subroutines

TeachingAssistant

Determines whether or not the ‘Teaching Assistant’ achievement has been unlocked

TenuredProf

Determines whether or not the ‘Tenured Professor’ achievement has been unlocked

frmViewStudent.vb

btnViewStudent_Load

Connects to databse, populates the account section

AccountSection

Populates the account section

btnViewProfile_Click

Displays the view profile form

btnSearch_Click

Searches the database for the searched-for student and displays them if they are found

lstStudents_SelectedIndexChanged

Gets the details of the selected student in order to populate the view profile form

btnViewStudentProfile_Click

Displays the view profile form for the selected student

btnBack_Click

Goes back

frmAddQuestion.vb

frmAddQuestion_Load

Loads the form, connects to database

Populate

Populates the comboboxes on the form

btnBack_Click

Goes back

cmboSubject_SelectedIndexChanged

Populates the topics combobox if a subject is selected

txtQuestion_Click

Blanks the textbox

txtAnswer_Click

Blanks the textbox

btnSubmit_Click

Checks whether the data entry is valid, and if so, adds the data to the database

List of Variables

mdlPublicVars.vb

C4Player *String*

Stores the current player in a game of Connect Four

C4Subject *String*

Stores the questions subject for a game of Connect Four

C4Difficulty *String*

Stores the questions difficulty for a game of Connect Four

C4Topic *String*

Stores the questions topic for a game of Connect Four

NaCPlayer *String*

Stores the current player in a game of Noughts and Crosses

NaCSubject *String*

Stores the questions subject for a game of Noughts and Crosses

NaCDifficulty *String*

Stores the questions difficulty for a game of Noughts and Crosses

NaCTopic *String*

Stores the questions topic for a game of Noughts and Crosses

RPSPlayer *String*

Stores the current player in a game of Rock, Paper, Scissors

RPSSubject *String*

Stores the questions subject for a game of Rock, Paper, Scissors

RPSDifficulty *String*

Stores the questions difficulty for a game of Rock, Paper, Scissors

RPSTopic *String*

Stores the questions topic for a game of Rock, Paper, Scissors

LoggedInWeapon *Integer*

Stores the logged-in player's weapon in a game of Rock, Paper, Scissors

OppWeapon *Integer*

Stores the opponent player's weapon in a game of Rock, Paper, Scissors

Path *String*

Stores the relative path to the program executable

DBPath *String*

Stores the path to the database

DBConn *ADODB.Connection*

Used to connect to the MySQL datbase

Server *String*

Stores the database server

Dtabase *String*

Stores the database name

UID *String*

Stores the database user login

Pwd *String*

Stores the database password

ConnStr *String*

Stores the connection string for the database

GameID *String*

Stores the GameID of a network Connect Four game

FnameStudent *String*

Stores the firstname of a logged-in student

LnameStudent *String*

Stores the firstname of a logged-in student

FnameOppStudent *String*

Stores the firstname of an opponent student

LnameOppStudent *String*

Stores the firstname of an opponent student

Form *String*

Stores the form of a logged-in student

ImageStudentLoc *String*

Stores the location of the image of a logged-in student

ImageOppStudentLoc *String*

Stores the location of the image of an opponent student

StudentID *Integer*

Stores the StudentID from the database

Student *Class*

Stores the details of the student class

Fname *String*

Stores the firstname of an object of the **Student** class

Lname *String*

Stores the lastname of an object of the **Student** class

Form *String*

Stores the form of an object of the **Student** class

Username *String*

Stores the username of an object of the **Student** class

C4Player *String*

Stores the Connect Four player of an object of the **Student** class

NaCPlayer *String*

Stores the Noughts and Crosses player of an object of the **Student** class

StudentID *Integer*

Stores the StudentID of an object of the **Student** class

Wins *Integer*

Stores the wins of an object of the **Student** class

Losses *Integer*

Stores the losses of an object of the **Student** class

Draws *Integer*

Stores the draws of an object of the **Student** class

RPSPlayer *Integer*

Stores the Rock, Paper, Scissors player of an object of the **Student** class

LoggedInStudent *Student*

Creates an object of the **Student** class for the logged-in user

OppStudent *Student*

Creates an object of the **Student** class for the opponent user

Viewing *Integer*

Used to populate the view profile form with the correct user's information

FnameTeacher *String*

Stores the firstname of a logged-in teacher

LnameTeacher *String*

Stores the firstname of a logged-in teacher

ImageTeacherLoc *String*

Stores the location of the image of a logged-in teacher

TeacherID *Integer*

Stores the TeacherID from the database

Teacher *Class*

Stores the details of the teacher class

Fname *String*

Stores the firstname of an object of the **Teacher** class

Lname *String*

Stores the lastname of an object of the **Teacher** class

Username *String*

Stores the username of an object of the **Teacher** class

TeacherID *Integer*

Stores the TeacherID of an object of the **Teacher** class

LoggedInTeacher *Teacher*

Creates an object of the **Teacher** class for the logged-in user

frmSplash.vb

Count *Integer*

Used to count down the loading

Tick *Integer*

Sets the step size for the timer tick event

frmFTSMsg.vb

Writer *System.IO.StreamWriter*

Used to write to the the First-Time Setup text file

frmFirstTimeSetup.vb

Writer *System.IO.StreamWriter*

Used to write to the the First-Time Setup text file

frmAddSubject.vb

RsNewRec *ADODB.Recordset*

Used to add a new record to the database

Errors *String*

Used to construct the error message for data entry validation

file *System.IO.FileStream*

Used to create a text file for the subject's topics

frmAddTopic.vb

RSStaff *ADODB.RecordSet*

Used to read records from the database

Result *MsgBoxResult*

Used for validation of data entry

Errors *String*

Used to construct the error message for data entry validation

Writer *System.IO.StreamWriter*

Used to write to the the subject topics text file

frmAddTeacher.vb

Uname *String*

Used to construct the username for the new user

Filepath*String*

Stores the path to the location of the new user's profile picture

Result *MsgBoxResult*

Used for validation of data entry

RsNewRec *ADODB.Recordset*

Used to add a new record to the database

Errors *String*

Used to construct the error message for data entry validation

file *System.IO.FileStream*

Used to create a text file for the teacher's breakdown

RSStaff *ADODB.RecordSet*

Used to read records from the database

L1 *String*

Gets the first letter of the new teacher's first name

L2 *String*

Gets the first letter of the new teacher's last name

Numbers *Integer*

Gets the two numbers for the end of the new teacher's username

frmAddStudent.vb

Uname *String*

Used to construct the username for the new user

Filepath*String*

Stores the path to the location of the new user's profile picture

Result *MsgBoxResult*

Used for validation of data entry

RsNewRec *ADODB.Recordset*

Used to add a new record to the database

Errors *String*

Used to construct the error message for data entry validation

file *System.IO.FileStream*

Used to create a text file for the student's breakdown

RSStaff *ADODB.RecordSet*

Used to read records from the database

L1 *String*

Gets the first letter of the new student's first name

L2 *String*

Gets the first letter of the new student's last name

Numbers *Integer*

Gets the two numbers for the end of the new teacher's username

frmLogin.vb

EnteredUsername *String*

Stores the entered username

EnteredPassword *String*

Stores the entered password

FTS *String*

Determines whether the First-Time Setup has been run already

Reader *System.IO.StreamReader*

Used to read the First-Time Setup text file

RSStaff *ADODB.RecordSet*

Used to read records from the database

frmStudentBreakdown.vb

Reader *System.IO.StreamReader*

Used to read the student's breakdown text file

frmStudentAchievements.vb

RSStaff *ADODB.RecordSet*

Used to read records from the database

frmConnect4HotseatLogin.vb

EnteredUsername *String*

Stores the entered username of the opponent

EnteredPassword *String*

Stores the entered password of the opponent

RSStaff *ADODB.RecordSet*

Used to read records from the database

frmConnect4HotseatSubject.vb

SubjectID *Integer*

Stores the SubjectID from the database

RSStaff *ADODB.RecordSet*

Used to read records from the database

Reader *System.IO.StreamReader*

Used to read the student's breakdown text file

frmConnect4Hotseat.vb

x *Integer*

Stores the x co-ord of the grid

y *Integer*

Stores the y co-ord of the grid

Group(7, 6) *PictureBox*

Stores the picturebox co-ords of the grid squares

TheGrid(7, 7) *Integer*

Stores the co-ords of the grid squares

Buttons (7) *Button*

Stores the buttons at the top of each column

C4Play *Class*

Stores the details of the C4Player class

Colour *String*

Stores the counter colour of an object of the **C4Play** class

Username *String*

Stores the username of an object of the **C4Play** class

Fname *String*

Stores the firstname of an object of the **C4Play** class

Iname *String*

Stores the lastname of an object of the **C4Play** class

Score *Class*

Stores the details of the Score class

ScoreNum *Integer*

Stores the score of an object of the **Score** class

RedScore *Score*

Creates an object of the **Score** class for the red player

YellowScore *Score*

Creates an object of the **Score** class for the yellow player

RedPlayer *Score*

Creates an object of the **C4Play** class for the red player

YellowPlayer *Score*

Creates an object of the **C4Play** class for the yellow player

QuestionID *Integer*

Stores the QuestionID from the database

StudentCurrLocationY *Integer*

Stores the y-coord of the current player label for the logged-in student

OppStudentCurrLocationY *Integer*

Stores the y-coord of the current player label for the opponent student

Why *Integer*

Used to detect wins along the y-axis (‘why’ used because ‘y’ was already taken, keeps it semi-self-documenting)

OppStudentCurrLocationY *Integer*

Stores the y-coord of the current player label for the opponent student

RedAddUp *Integer*

Tracks how many consecutive red counters there are

YellowAddUp *Integer*

Tracks how many consecutive yellow counters there are

QCorrect *Boolean*

Stores whether the user answered a question correctly or not

Won *Boolean*

Determines whether the game has been won

Draw *Boolean*

Determines whether the game has been drawn

GridHor *Integer*

Stores the current x-coord on the grid

GridVer *Integer*

Stores the current y-coord on the grid

RsNewRec *ADODB.Recordset*

Used to add a new record to the database

Writer *System.IO.StreamWriter*

Used to write to the the users' breakdown text files

SpacestoLeft *Integer*

Determines how many spaces are to the left of the last-dropped counter

SpacestoRight *Integer*

Determines how many spaces are to the right of the last-dropped counter

SpacesAbove *Integer*

Determines how many spaces are above the last-dropped counter

SpacesBelow *Integer*

Determines how many spaces are below the last-dropped counter

Blanks *Integer*

Determines how many blank squares remain

v *Integer*

Used for determining diagonal wins

z *Integer*

Used for determining diagonal wins

a *Integer*

Used for determining diagonal wins

w *Integer*

Used for determining diagonal wins

RSStaff *ADODB.RecordSet*

Used to read records from the database

frmConnect4NetworkLobby.vb

NoOpp *Boolean*

Determines if a game has an opponent in it or not

Host *Boolean*

Determines if the logged-in user is host of a game or not

CurrPlaying *String*

Indicates if the logged-in user is currently playing a game

RSStaff *ADODB.RecordSet*

Used to read records from the database

EndofGamID *Integer*

Determines the end of the GameID of a game

GamID *String*

Stores the GameID of a game

HUsername *String*

Stores the username of the host of a game

Username *String*

Stores the username of the other player

RsNewRec *ADODB.Recordset*

Used to add a new record to the database

frmConnect4NetworkHostGame.vb

SubjectID *String*

Stores the SubjectID from the database

Result *MsgBoxResult*

Used for validation of data entry

RsNewRec *ADODB.Recordset*

Used to add a new record to the database

Errors *String*

Used to construct the error message for data entry validation

RSStaff *ADODB.RecordSet*

Used to read records from the database

Reader *System.IO.StreamReader*

Used to read the subject's topics text file

frmConnect4NetworkYourGames.vb

NoOpp *Boolean*

Determines if a game has an opponent in it or not

Result *MsgBoxResult*

Used for hosting game

CurrPlaying *String*

Indicates if the logged-in user is currently playing a game

RSStaff *ADODB.RecordSet*

Used to read records from the database

EndofGamID *Integer*

Determines the end of the GameID of a game

GamID *String*

Stores the GameID of a game

Username *String*

Stores the username of the other player

RsNewRec *ADODB.Recordset*

Used to add a new record to the database

frmConnect4Network.vb

x *Integer*

Stores the x co-ord of the grid

y *Integer*

Stores the y co-ord of the grid

Group(7, 6) *PictureBox*

Stores the picturebox co-ords of the grid squares

TheGrid(7, 7) *Integer*

Stores the co-ords of the grid squares

Buttons (7) *Button*

Stores the buttons at the top of each column

C4Play *Class*

Stores the details of the C4Player class

Colour *String*

Stores the counter colour of an object of the **C4Play** class

Username *String*

Stores the username of an object of the **C4Play** class

Fname *String*

Stores the firstname of an object of the **C4Play** class

Iname *String*

Stores the lastname of an object of the **C4Play** class

StudentID *Integer*

Stores the StudentID of an object of the **C4Play** class

Score *Class*

Stores the details of the Score class

ScoreNum *Integer*

Stores the score of an object of the **Score** class

RedScore *Score*

Creates an object of the **Score** class for the red player

YellowScore *Score*

Creates an object of the **Score** class for the yellow player

RedPlayer *Score*

Creates an object of the **C4Play** class for the red player

YellowPlayer *Score*

Creates an object of the **C4Play** class for the yellow player

QuestionID *Integer*

Stores the QuestionID from the database

StudentCurrLocationY *Integer*

Stores the y-coord of the current player label for the logged-in student

OppStudentCurrLocationY *Integer*

Stores the y-coord of the current player label for the opponent student

Why *Integer*

Used to detect wins along the y-axis ('why' used because 'y' was already taken, keeps it semi-self-documenting)

RedAddUp *Integer*

Tracks how many consecutive red counters there are

YellowAddUp *Integer*

Tracks how many consecutive yellow counters there are

QCorrect *Boolean*

Stores whether the user answered a question correctly or not

Won *Boolean*

Determines whether the game has been won

Draw *Boolean*

Determines whether the game has been drawn

YourGo *Boolean*

Determines whose go it is

GridHor *Integer*

Stores the current x-coord on the grid

GridVer *Integer*

Stores the current y-coord on the grid

RSStaff *ADODB.RecordSet*

Used to read records from the database

RsNewRec *ADODB.Recordset*

Used to add a new record to the database

Writer *System.IO.StreamWriter*

Used to write to the the users' breakdown text files

SpacestoLeft *Integer*

Determines how many spaces are to the left of the last-dropped counter

SpacestoRight *Integer*

Determines how many spaces are to the right of the last-dropped counter

SpacesAbove *Integer*

Determines how many spaces are above the last-dropped counter

SpacesBelow *Integer*

Determines how many spaces are below the last-dropped counter

Blanks *Integer*

Determines how many blank squares remain

v *Integer*

Used for determining diagonal wins

z *Integer*

Used for determining diagonal wins

a *Integer*

Used for determining diagonal wins

w *Integer*

Used for determining diagonal wins

RSDelRec *ADODB.Recordset*

Used to delete a record from the database

frmNaCHotseatLogin.vb

EnteredUsername *String*

Stores the entered username of the opponent

EnteredPassword *String*

Stores the entered password of the opponent

RSStaff *ADODB.RecordSet*

Used to read records from the database

frmNaCHotseatSubject.vb

SubjectID *Integer*

Stores the SubjectID from the database

RSStaff *ADODB.RecordSet*

Used to read records from the database

Reader *System.IO.StreamReader*

Used to read the student's breakdown text file

frmNaCHotseat.vb

TheGrid(9) *Integer*

Stores the co-ords of the grid squares

NaCPlay *Class*

Stores the details of the NaCPlayer class

Letter *String*

Stores the counter letter of an object of the **NaCPlay** class

Username *String*

Stores the username of an object of the **NaCPlay** class

Fname *String*

Stores the firstname of an object of the **NaCPlay** class

lname *String*

Stores the lastname of an object of the **NaCPlay** class

Score *Class*

Stores the details of the Score class

ScoreNum *Integer*

Stores the score of an object of the **Score** class

XScore *Score*

Creates an object of the **Score** class for the X player

OScore *Score*

Creates an object of the **Score** class for the O player

XPlayer *Score*

Creates an object of the **C4Play** class for the X player

OPlayer *Score*

Creates an object of the **C4Play** class for the O player

QuestionID *Integer*

Stores the QuestionID from the database

StudentCurrLocationY *Integer*

Stores the y-coord of the current player label for the logged-in student

OppStudentCurrLocationY *Integer*

Stores the y-coord of the current player label for the opponent student

QCorrect *Boolean*

Stores whether the user answered a question correctly or not

WhoWon *Boolean*

Stores the winning user

RsNewRec *ADODB.Recordset*

Used to add a new record to the database

Writer *System.IO.StreamWriter*

Used to write to the the users' breakdown text files

n *Integer*

Used to blank the grid upon game reset

RSStaff *ADODB.RecordSet*

Used to read records from the database

frmRPSHotseatLogin.vb

EnteredUsername *String*

Stores the entered username of the opponent

EnteredPassword *String*

Stores the entered password of the opponent

RSStaff *ADODB.RecordSet*

Used to read records from the database

frmRPSHotseatSubject.vb

SubjectID *Integer*

Stores the SubjectID from the database

RSStaff *ADODB.RecordSet*

Used to read records from the database

Reader *System.IO.StreamReader*

Used to read the student's breakdown text file

frmRPSHotseat.vb

QuestionID *Integer*

Stores the QuestionID from the database

StudentCurrLocationY *Integer*

Stores the y-coord of the current player label for the logged-in student

OppStudentCurrLocationY *Integer*

Stores the y-coord of the current player label for the opponent student

QCorrect *Boolean*

Stores whether the user answered a question correctly or not

goes *Integer*

Determines when both players have gone

RSStaff *ADODB.RecordSet*

Used to read records from the database

RsNewRec *ADODB.Recordset*

Used to add a new record to the database

frmRPSFight.vb

count *Integer*

Used for the timers

countdown *Integer*

Used for the initial countdown

time *Integer*

Used to end the fight

x *Integer*

Used for the cords of the weapon picture

y *Integer*

Used for the cords of the weapon picture

xx *Integer*

Used for the cords of the weapon picture

yy *Integer*

Used for the cords of the weapon picture

RsNewRec *ADODB.Recordset*

Used to add a new record to the database

Writer *System.IO.StreamWriter*

Used to write to the the users' breakdown text files

frmTeacherAccount.vb

RSStaff *ADODB.RecordSet*

Used to read records from the database

frmTeacherBreakdown.vb

Reader *System.IO.StreamReader*

Used to read the student's breakdown text file

frmTeacherAchievements.vb

RSStaff *ADODB.RecordSet*

Used to read records from the database

frmViewStudent.vb

RSStaff *ADODB.RecordSet*

Used to read records from the database

EndofStudentID *Integer*

Determines the end of the StudentID of a student

StudID *String*

Stores the StudentID of a game

frmAddQuestion.vb

SubjectID *String*

Stores the SubjectID from the database

RSStaff *ADODB.RecordSet*

Used to read records from the database

Reader *System.IO.StreamReader*

Used to read the student's breakdown text file

Result *MsgBoxResult*

Used for validation of data entry

RsNewRec *ADODB.Recordset*

Used to add a new record to the database

Errors *String*

Used to construct the error message for data entry validation

Writer *System.IO.StreamWriter*

Used to write to the the user's breakdown text file

Testing

Test Strategy

In software development, as in just about any other form of development, testing is a vital step in the process of releasing a new product. A product that isn't sufficiently tested could be shipped in an unsatisfactory condition, damaging the reputation of the company and possibly leading to costly mass recalls of the product. Examples of poorly-tested products are plentiful, such as the operating system Microsoft Vista, which was disregarded by the global computing community as slow, bloated and unstable.

You need to perform multiple types of testing to catch the multiple ways in which your program could fail. One example would be validation testing for text inputs, in which normal, extreme and erroneous data would be input. For example, if a textbox was intended to receive an input of a number between and including one and ten, the normal input would be something like 5, the extreme input would be 10 and the erroneous input could be a text string like "seven" or an invalid numerical value such as 24.

Within the world of testing, there are two methods. Black box testing is when a user unfamiliar with the inner workings of the program attempts to find errors and recreate them, so that they can then be passed on to the programmer to fix, an example being computer game QA testing. White box testing, on the other hand, is when an experienced programmer, who has an in-depth knowledge of how computer programs work, predicts ways the system might be caused to fail and tests these vulnerabilities.

In my application I shall be using both black and white box testing, utilising a detailed test plan and external users. I shall test all aspects of my program, e.g. the interface, validation and functionality of the system.

Test Plan

Section 1: Loading & Setup

1.1

Scope

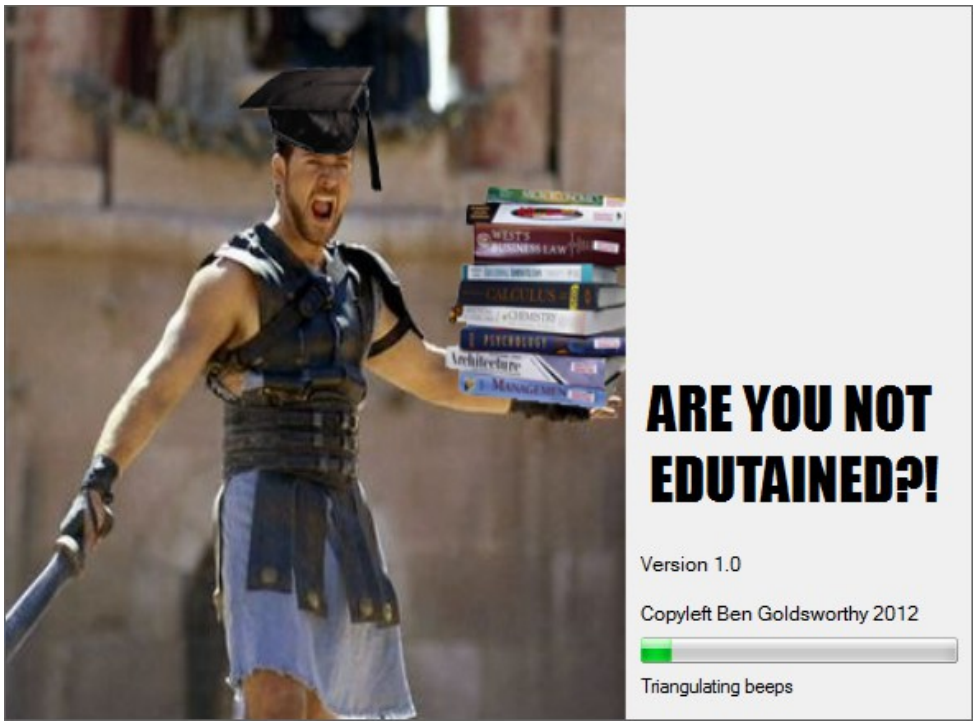
Test that the program runs

Method

1. Double click the Edutainment_Suite.exe application

Expected Result

The program loads up in no more than ten seconds



Actual Result

PASS/FAIL

1.2

Scope

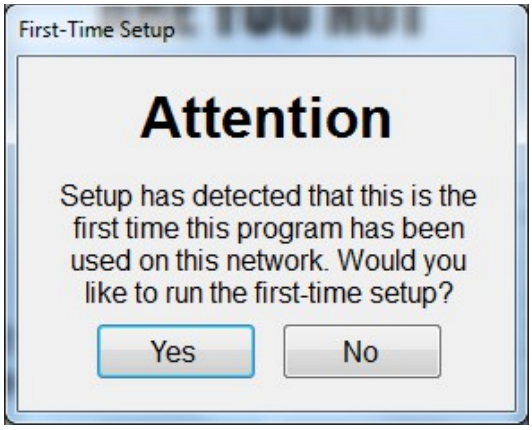
Test that, if the first-time setup had not yet been run, the alert message is displayed

Method

1. Check that the file FTS.txt in *G:\Computing Group\Y10 2009-10\Ben's\Edutainment Suite\Connect4\bin\Debug* contains the text “0”
2. Run the program

Expected Result

The first-time setup alert message is displayed



Actual Result

PASS/FAIL

1.3

Scope

Test that the first-time setup alert message is always on top of any other programs

Method

1. Check that the file FTS.txt in *G:\Computing Group\Y10 2009-10\Ben's\Edutainment Suite\Connect4\bin\Debug* contains the text “0”
2. Run the program
3. Open up another program

Expected Result

The first-time setup alert message is always on top of any other programs

Actual Result

PASS/FAIL

1.4

Scope

Test that the no button on the first time setup alert message works correctly

Method

1. Check that the file FTS.txt in *G:\Computing Group\Y10 2009-10\Ben's\Edutainment Suite\Connect4\bin\Debug* contains the text “0”
2. Run the program
3. Click the ‘No’ button
4. Observe the file FTS.txt in *G:\Computing Group\Y10 2009-10\Ben's\Edutainment Suite\Connect4\bin\Debug*

Expected Result

The message will close and the text file will be amended to “1”

Actual Result

PASS/FAIL

1.5

Scope

Test that, if the first-time setup has been run before, no message will appear

Method

1. Check that the file FTS.txt in *G:\Computing Group\Y10 2009-10\Ben's\Edutainment Suite\Connect4\bin\Debug* contains the text “1”
2. Run the program

Expected Result

No message will appear

Actual Result

PASS/FAIL

1.6

Scope

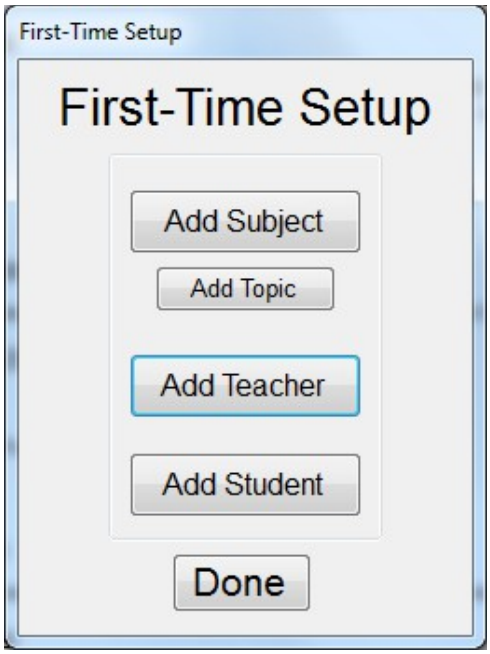
Test that the yes button on the first time setup alert message works correctly

Method

1. Check that the file FTS.txt in *G:\Computing Group\Y10 2009-10\Ben's\Edutainment Suite\Connect4\bin\Debug* contains the text “0”
2. Run the program
3. Click the ‘Yes’ button

Expected Result

The first-time setup form is displayed



Actual Result

PASS/FAIL

1.7

Scope

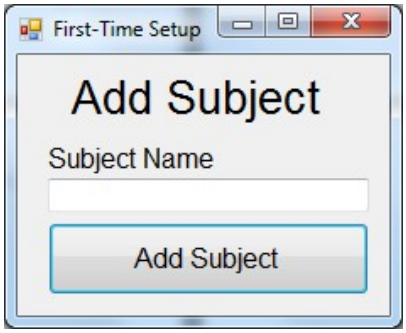
Test that the add subject button on the first time setup form works correctly

Method

1. Check that the file FTS.txt in *G:\Computing Group\Y10 2009-10\Ben's\Edutainment Suite\Connect4\bin\Debug* contains the text “0”
2. Run the program
3. Click the ‘Yes’ button
4. Click the ‘Add Subject’ button

Expected Result

The add subject form is displayed



Actual Result

PASS/FAIL

1.8

Scope

Test that the add subject button on the add subject form works correctly

a. Test that the add subject form, upon no data being input, will reject the input and blank the data entry controls
b. Test that the add subject form, upon valid data entry but not confirming the input, will reject the input
c. Test that the add subject form, upon inputting of a valid subject and user confirmation, will accept the input and add the record to the database
d. Test that the add subject form, upon inputting of a valid subject and user confirmation, will create a text file for the subject in *G:\Computing Group\Y10 2009-10\Ben's\Edutainment Suite\Connect4\bin\Debug\Topics*

Method

1. Check that the file FTS.txt in *G:\Computing Group\Y10 2009-10\Ben's\Edutainment Suite\Connect4\bin\Debug* contains the text “0”
2. Run the program
3. Click the ‘Yes’ button
4. Click the ‘Add Subject’ button
5. Click the ‘Add Subject’ button
6. Enter “Psychology” as a subject
7. Click the ‘Add Subject’ button
8. Click the ‘No’ button
9. Click the ‘Add Subject’ button

10. Click the ‘Yes’ button
11. Observe ‘tblSubjects’ in the database
12. Observe ‘PsychologyTopics.txt’ in *G:\Computing Group\Y10 2009-10\Ben's\Edutainment Suite\Connect4\bin\Debug\Topics*

Expected Result

a. The form will reject the data and blank the entry controls
b. The form will reject the data
c. The form will accept the data and add a record to the database
d. The form will create the topics text file

Actual Result

PASS/FAIL

1.9

Scope

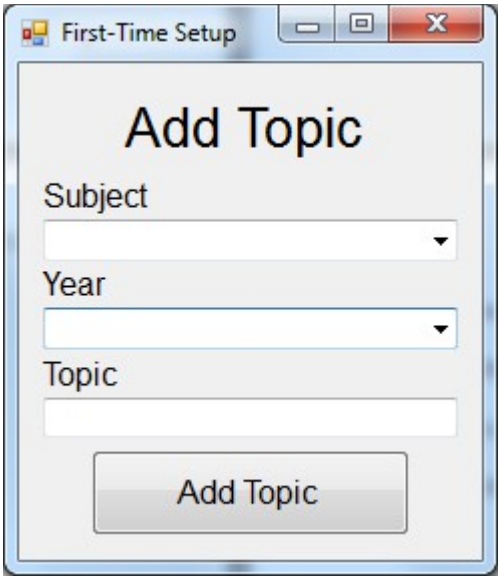
Test that the add topic button on the first time setup form works correctly

Method

1. Check that the file FTS.txt in *G:\Computing Group\Y10 2009-10\Ben's\Edutainment Suite\Connect4\bin\Debug* contains the text “0”
2. Run the program
3. Click the ‘Yes’ button
4. Click the ‘Add Topic’ button

Expected Result

The add topic form is displayed



Actual Result

PASS/FAIL

1.10

Scope

Test that the subject combobox on the add topic form populates correctly

Method

1. Check that the file FTS.txt in *G:\Computing Group\Y10 2009-10\Ben's\Edutainment Suite\Connect4\bin\Debug* contains the text “0”
2. Run the program
3. Click the ‘Yes’ button
4. Click the ‘Add Topic’ button
5. Observe the combobox labelled ‘Subject’

Expected Result

The combobox will be populated with the subjects from the database correctly

Actual Result

PASS/FAIL

1.11

Scope

Test that the add topic button on the add topic form works correctly

a. Test that the add topic form, upon no data being input, will reject the input and blank the data entry controls
b. Test that the add topic form, upon partial data entry, will reject the input and blank the data entry controls
c. Test that the add topic form, upon valid data entry but not confirming the input, will reject the input
d. Test that the add topic form, upon valid data entry and confirming the input, will accept the input and amend the chosen subject's topics text file in *G:\Computing Group\Y10 2009-10\Ben's\Edutainment Suite\Connect4\bin\Debug*

Method

1. Check that the file FTS.txt in *G:\Computing Group\Y10 2009-10\Ben's\Edutainment Suite\Connect4\bin\Debug* contains the text "0"
2. Run the program
3. Click the 'Yes' button
4. Click the 'Add Topic' button
5. Click the 'Add Topic' button
6. Select "Psychology" as a subject
7. Click the 'Add Subject' button
8. Select "Year 12" as a year
9. Click the 'Add Subject' button
10. Enter "Stress" as a topic
11. Click the 'Add Subject' button
12. Click the 'No' button
13. Observe 'PsychologyTopics.txt' in *G:\Computing Group\Y10 2009-10\Ben's\Edutainment Suite\Connect4\bin\Debug\Topics*
14. Click the 'Add Subject' button
15. Click the 'Yes' button
16. Observe 'PsychologyTopics.txt' in *G:\Computing Group\Y10 2009-10\Ben's\Edutainment Suite\Connect4\bin\Debug\Topics*

Expected Result

a. The form will reject the data and blank the entry controls
b. The form will reject the data and blank the entry controls
c. The form will reject the data
d. The form will add the inputted topic into the selected subject's topics text file

Actual Result

PASS/FAIL

1.12

Scope

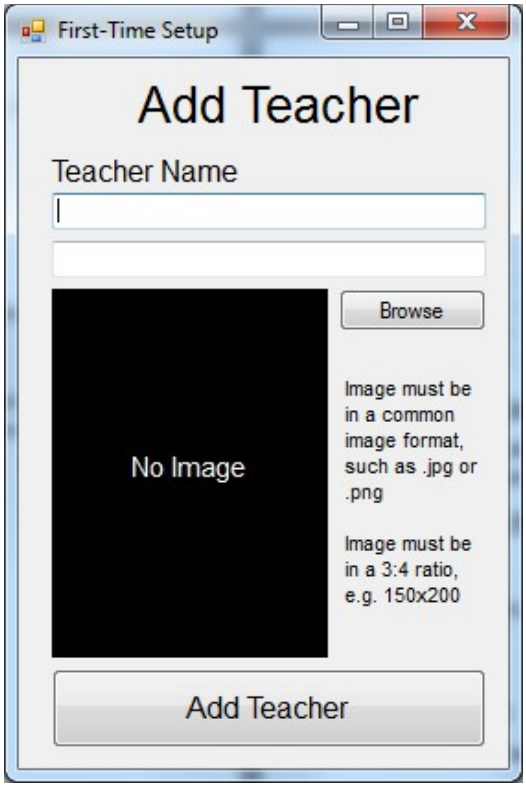
Test that the add teacher button on the first time setup form works correctly

Method

1. Check that the file FTS.txt in *G:\Computing Group\Y10 2009-10\Ben's\Edutainment Suite\Connect4\bin\Debug* contains the text “0”
2. Run the program
3. Click the ‘Yes’ button
4. Click the ‘Add Teacher’ button

Expected Result

The add teacher form is displayed



Actual Result

PASS/FAIL

1.13

Scope

Test that the browse image button on the add teacher form works correctly

Method

1. Check that the file FTS.txt in *G:\Computing Group\Y10 2009-10\Ben's\Edutainment Suite\Connect4\bin\Debug* contains the text “0”
2. Run the program
3. Click the ‘Yes’ button
4. Click the ‘Add Teacher’ button
5. Click the ‘Browse’ button
6. Select an image file (.jpg, .png, etc.)
7. Click the ‘Open’ button

Expected Result

The button displays the file viewer window, from which an image file can be selected. After being selected, the image will be displayed on the form

Actual Result

PASS/FAIL

1.14

Scope

Test that the add teacher button on the add teacher form works correctly

a. Test that the add teacher form, upon no data being input, will reject the input and blank the data entry controls
b. Test that the add teacher form, upon partial data entry, will reject the input and blank the data entry controls
c. Test that the add teacher form, upon inputting of valid and complete details and user confirmation, will accept the input and add the record to the database
d. Test that the add teacher form, upon inputting of valid and complete details and user confirmation, will create a text file for the teacher in *G:\Computing Group\Y10 2009-10\Ben's\Edutainment Suite\Connect4\bin\Debug\QuestionCreation*

Method

1. Check that the file FTS.txt in *G:\Computing Group\Y10 2009-10\Ben's\Edutainment Suite\Connect4\bin\Debug* contains the text “0”
2. Run the program
3. Click the ‘Yes’ button
4. Click the ‘Add Teacher’ button
5. Click the ‘Add Teacher’ button
6. Enter “Stephen” as a first name

7. Click the ‘Add Teacher’ button
8. Enter “Topham” as a last name
9. Click the ‘Add Teacher’ button
10. Click the ‘Browse’ button
11. Select an image file
12. Click the ‘Open’ button
13. Click the ‘Add Teacher’ button
14. Click the ‘No’ button
15. Observe ‘stst11.txt’ in *G:\Computing Group\Y10 2009-10\Ben's\Edutainment Suite\Connect4\bin\Debug\QuestionCreation*
16. Click the ‘Add Teacher’ button
13. Click the ‘Yes’ button
14. Observe ‘tblTeachers’ in the database
17. Observe ‘stst11.txt’ in *G:\Computing Group\Y10 2009-10\Ben's\Edutainment Suite\Connect4\bin\Debug\QuestionCreation*

Expected Result

a. The form will reject the data and blank the entry controls
b. The form will reject the data and blank the entry controls
c. The form will reject the data
d. The form will add the a record to the database and create the question creation text file

Actual Result

PASS/FAIL

1.15

Scope

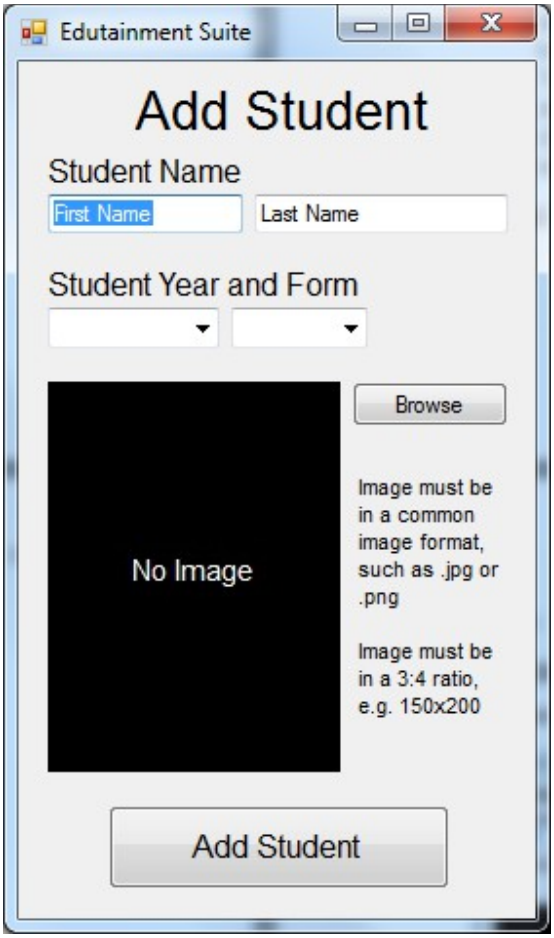
Test that the add student button on the first time setup form works correctly

Method

1. Check that the file FTS.txt in *G:\Computing Group\Y10 2009-10\Ben's\Edutainment Suite\Connect4\bin\Debug* contains the text “0”
2. Run the program
3. Click the ‘Yes’ button
4. Click the ‘Add Student’ button

Expected Result

The add student form is displayed



Actual Result

PASS/FAIL

1.16

Scope

Test that the browse image button on the add student form works correctly

Method

1. Check that the file FTS.txt in *G:\Computing Group\Y10 2009-10\Ben's\Edutainment Suite\Connect4\bin\Debug* contains the text “0”
2. Run the program
3. Click the ‘Yes’ button
4. Click the ‘Add Student’ button
5. Click the ‘Browse’ button
6. Select an image file (.jpg, .png, etc.)
7. Click the ‘Open’ button

Expected Result

The button displays the file viewer window, from which an image file can be selected. After being selected, the image will be displayed on the form

Actual Result

PASS/FAIL

1.17

Scope

Test that the add student button on the add student form works correctly

a. Test that the add student form, upon no data being input, will reject the input and blank the data entry controls
b. Test that the add student form, upon partial data entry, will reject the input and blank the data entry controls
c. Test that the add student form, upon inputting of valid and complete details and user confirmation, will accept the input and add the record to the database
d. Test that the add student form, upon inputting of valid and complete details and user confirmation, will create a text file for the student in *G:\Computing Group\Y10 2009-10\Ben's\Edutainment Suite\Connect4\bin\Debug\WinsLosses*

Method

1. Check that the file FTS.txt in *G:\Computing Group\Y10 2009-10\Ben's\Edutainment Suite\Connect4\bin\Debug* contains the text “0”
2. Run the program
3. Click the ‘Yes’ button
4. Click the ‘Add Student’ button
5. Click the ‘Add Student’ button
6. Enter “Thomas” as a first name
7. Click the ‘Add Student’ button
8. Enter “Shawley” as a last name
9. Click the ‘Add Student’ button
10. Select “12” and “E” from the form comboboxes
11. Click the ‘Add Student’ button
12. Click the ‘Browse’ button
13. Select an image file
14. Click the ‘Open’ button
15. Click the ‘Add Student’ button
16. Click the ‘No’ button
17. Observe ‘95ts11.txt’ in *G:\Computing Group\Y10 2009-10\Ben's\Edutainment Suite\Connect4\bin\Debug\WinsLosses*
18. Click the ‘Add Student’ button

15. Click the ‘Yes’ button
16. Observe ‘tblStudents’ in the database
19. Observe ‘95ts11.txt’ in *G:\Computing Group\Y10 2009-10\Ben's\Edutainment Suite\Connect4\bin\Debug\WinsLosses*

Expected Result

a. The form will reject the data and blank the entry controls
b. The form will reject the data and blank the entry controls
c. The form will reject the data
d. The form will add the a record to the database and create the win-loss text file

Actual Result

PASS/FAIL

1.18

Scope

Test that the done button on the first time setup form works correctly

Method

1. Check that the file FTS.txt in *G:\Computing Group\Y10 2009-10\Ben's\Edutainment Suite\Connect4\bin\Debug* contains the text “0”
2. Run the program
3. Click the ‘Yes’ button
4. Click the ‘Done’ button
5. Observe the file FTS.txt in *G:\Computing Group\Y10 2009-10\Ben's\Edutainment Suite\Connect4\bin\Debug*

Expected Result

The login form is displayed, and the text file is changed to read “1”

Actual Result

PASS/FAIL

1.19

Scope

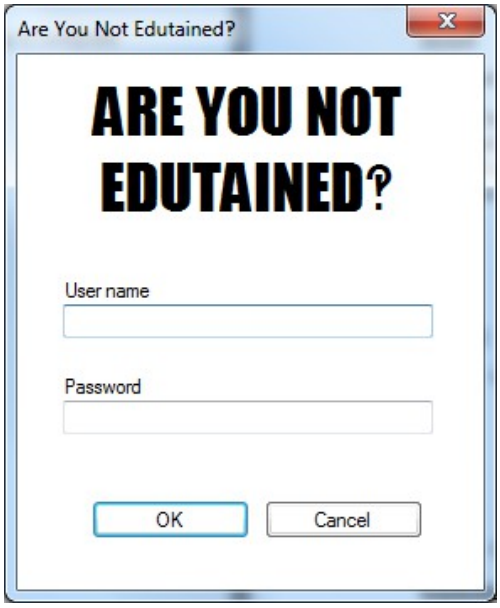
Test that the splash screen loads the login form once the loading is complete

Method

3. Double click the Edutainment_Suite.exe application
4. Wait for the loading to complete

Expected Result

The login form is displayed



Actual Result

PASS/FAIL

1.20

Scope

Test that the cancel button on the login form works correctly

Method

1. Load up the login form
2. Click the ‘cancel’ button

Expected Result

The program closes

Actual Result

PASS/FAIL

1.21

Scope

Test that the OK button on the login form works correctly

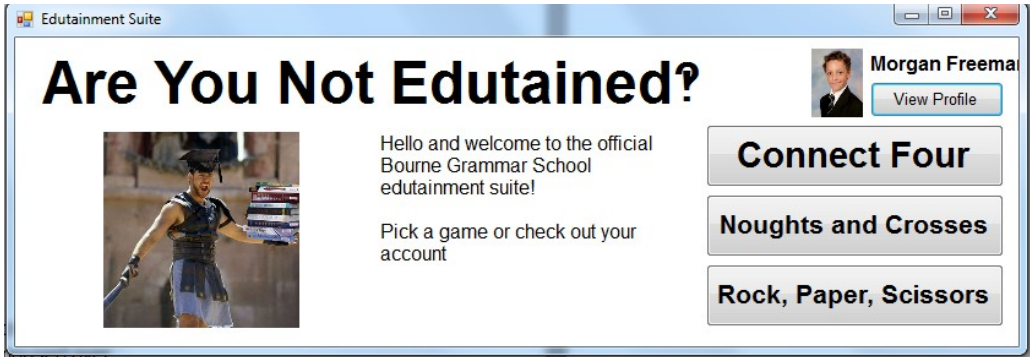
a. Test that the login form, upon inputting of incorrect login credentials, will refuse the login and blank the data entry controls
b. Test that the login form, upon inputting of correct login credentials, will accept the login and load the main menu form
    i. Test that login form, upon receiving valid student credentials, will display the student main menu form
    ai. Test that the login form, upon receiving valid teacher credentials, will display the teacher main menu form

Method

1. Load up the login form
    a. Enter “abcdef” as a username and password
    b. Enter valid login credentials
        i. Enter “95mh11” as a username and “password” as a password
        ii. Enter “stst11” as a username and “password” as a password
2. Click the ‘OK’ button in the bottom left

Expected Result

a. The form will reject the login and blank the entry controls
b. The form will accept the login and proceed onto the main menu form
    i. The next form will be the student main menu form



    ii. The next form will be the teacher main menu form

Actual Result

PASS/FAIL

Section 2: Main Menu & Account View

Sub-Section 1: Student Experience

2.1.1

Scope

Test that the image and description on the student main menu form change when the game buttons or the form are moused over

Method

1. Enter “95mh11” as a username and “password” as a password
2. Move the mouse over the game buttons and form
    a. Mouse over the Connect Four Button
    b. Mouse over the Noughts and Crosses button
    c. Mouse over the Rock, Paper, Scissors button
    d. Mouse over the form
3. Observe the picture and game description changing

Expected Result

a. The picture and description will be Connect Four
b. The picture and description will be Noughts and Crosses
c. The picture and description will be Rock, Paper, Scissors
d. The picture and description will be the defaults

Actual Result

PASS/FAIL

2.1.2

Scope

Test that the account details section of the student main menu form populates correctly

Method

1. Enter “95mh11” as a username and “password” as a password
2. Observe the account details section

Expected Result

The correct name and picture are displayed on the form

Actual Result

PASS/FAIL

2.1.3

Scope

Test that the view profile button on the student main menu form works correctly

Method

1. Enter “95mh11” as a username and “password” as a password
2. Click the ‘View Profile’ button in the top right

Expected Result

The view student profile form is displayed

Actual Result

PASS/FAIL

2.1.4

Scope

Test that the details the view profile form is populated with are correct

Method

1. Enter “95mh11” as a username and “password” as a password
2. Click the ‘View Profile’ button in the top right
3. Observe the view profile form

Expected Result

The view student profile form displays the logged-in student’s details correctly

Actual Result

PASS/FAIL

2.1.5

Scope

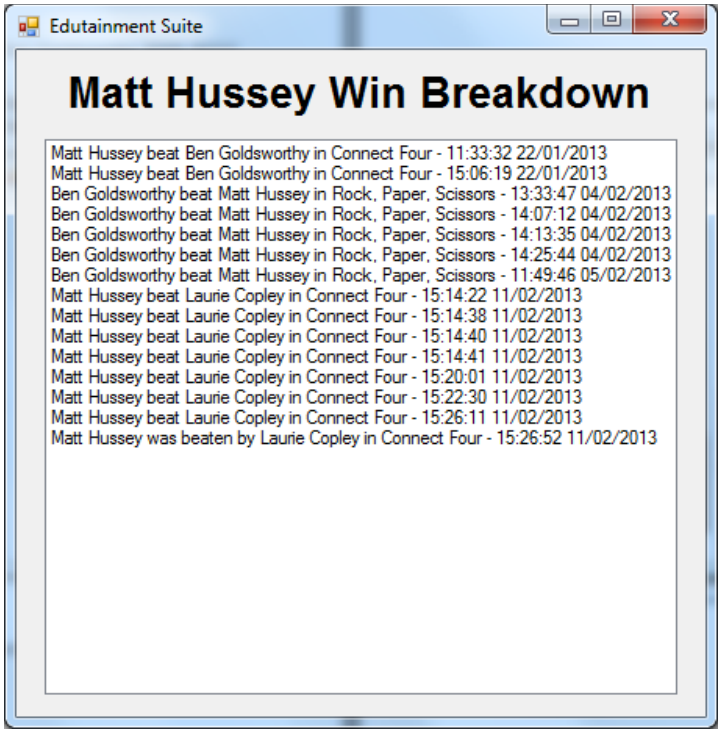
Test that the view details button on the view student profile form works correctly

Method

1. Enter “95mh11” as a username and “password” as a password
2. Click the ‘View Profile’ button in the top right
3. Click the ‘Details’ button in the centre of the form

Expected Result

The view student profile details form is displayed



Actual Result

PASS/FAIL

2.1.6

Scope

Test that the details the view profile details form is populated with are correct

Method

1. Enter “95mh11” as a username and “password” as a password
2. Click the ‘Details’ button in the top right
3. Observe the view profile details form

Expected Result

The view student profile form displays the logged-in student’s details correctly

Actual Result

PASS/FAIL

2.1.7

Scope

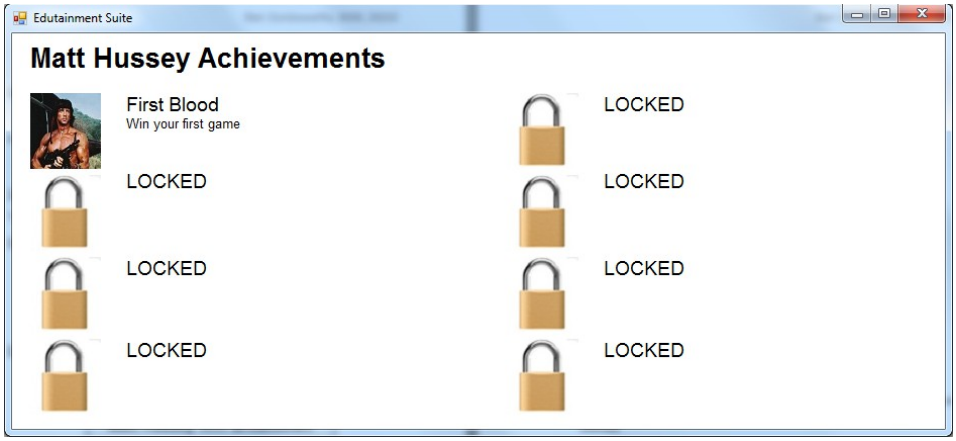
Test that the view achievements button on the view student profile form works correctly

Method

1. Enter “95mh11” as a username and “password” as a password
2. Click the ‘View Profile’ button in the top right
3. Click the ‘Achievement’ button in the bottom-centre of the form

Expected Result

The view student achievements form is displayed



Actual Result

PASS/FAIL

2.1.8

Scope

Test that the details the view achievements form is populated with are correct

Method

1. Enter “95mh11” as a username and “password” as a password
2. Click the ‘Achievements’ button in the top right
3. Observe the view achievements form

Expected Result

The view student achievements form displays the logged-in student's details correctly

Actual Result

PASS/FAIL

2.1.9

Scope

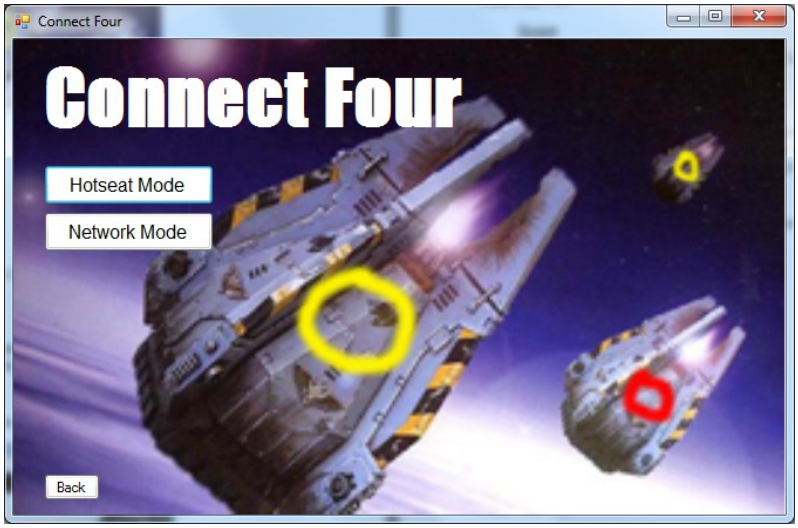
Test that the Connect Four button on the student home form works correctly

Method

1. Enter "95mh11" as a username and "password" as a password
2. Click the 'Connect Four' button in the top right

Expected Result

The Connect Four main menu form is displayed



Actual Result

PASS/FAIL

2.1.10

Scope

Test that the Noughts & Crosses button on the student home form works correctly

Method

1. Enter “95mh11” as a username and “password” as a password
2. Click the ‘Noughts and Crosses’ button in the top right

Expected Result

The Noughts and Crosses main menu form is displayed

Actual Result

PASS/FAIL

2.1.11

Scope

Test that the Rock, Paper, Scissors button on the student home form works correctly

Method

1. Enter “95mh11” as a username and “password” as a password
2. Click the ‘Rock, Paper, Scissors’ button in the top right

Expected Result

The Rock, Paper, Scissors main menu form is displayed

Actual Result

PASS/FAIL

Sub-Section 2: Staff Experience

2.2.1

Scope

Test that the account details section of the staff main menu form populates correctly

Method

3. Enter “stst11” as a username and “password” as a password
4. Observe the account details section

Expected Result

The correct name and picture are displayed on the form

Actual Result

PASS/FAIL

2.2.2

Scope

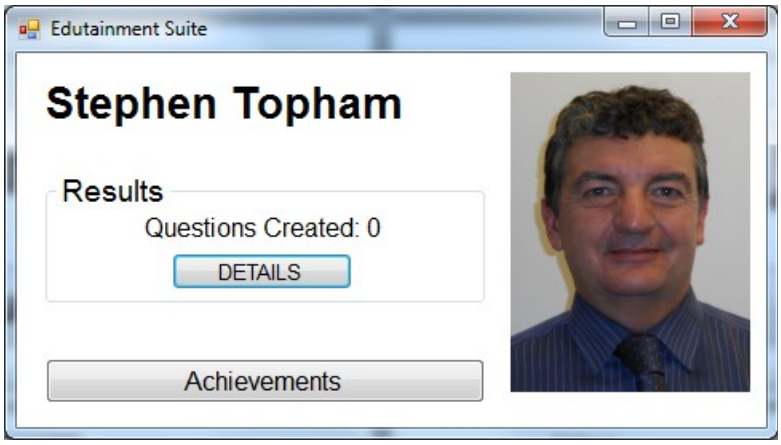
Test that the view profile button on the staff main menu form works correctly

Method

3. Enter “stst11” as a username and “password” as a password
4. Click the ‘View Profile’ button in the top right

Expected Result

The view staff profile form is displayed



Actual Result

PASS/FAIL

2.2.3

Scope

Test that the details the view staff profile form is populated with are correct

Method

4. Enter “stst11” as a username and “password” as a password
5. Click the ‘View Profile’ button in the top right
6. Observe the view profile form

Expected Result

The view staff profile form displays the logged-in staff member’s details correctly

Actual Result

PASS/FAIL

2.2.4

Scope

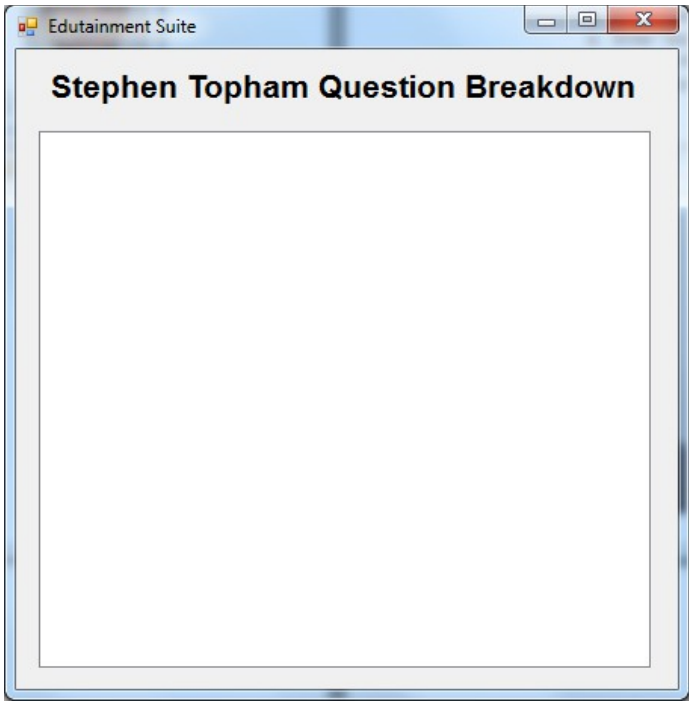
Test that the view details buttons on the view staff profile form works correctly

Method

4. Enter “stst11” as a username and “password” as a password
5. Click the ‘View Profile’ button in the top right
6. Click the ‘Details’ button in the centre of the form

Expected Result

The view staff profile details form is displayed



Actual Result

PASS/FAIL

2.2.5

Scope

Test that the details the view staff profile details form is populated with are correct

Method

4. Enter “stst11” as a username and “password” as a password
5. Click the ‘Details’ button in the top right
6. Observe the view profile details form

Expected Result

The view staff profile form displays the logged-in staff member’s details correctly

Actual Result

PASS/FAIL

2.2.6

Scope

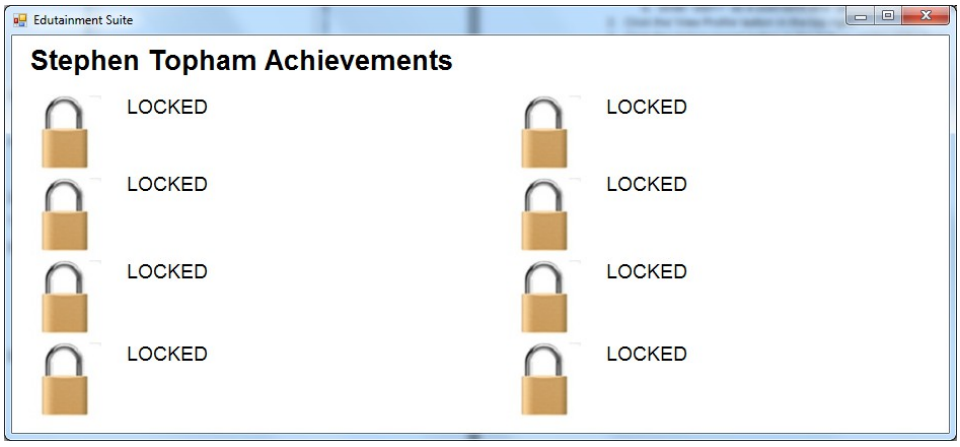
Test that the view achievements button on the view staff profile form works correctly

Method

4. Enter “stst11” as a username and “password” as a password
5. Click the ‘View Profile’ button in the top right
6. Click the ‘Achievement’ button in the bottom-centre of the form

Expected Result

The view staff achievements form is displayed



Actual Result

PASS/FAIL

2.2.7

Scope

Test that the details the view staff achievements form is populated with are correct

Method

4. Enter “stst11” as a username and “password” as a password
5. Click the ‘Achievements’ button in the top right
6. Observe the view achievements form

Expected Result

The view staff achievements form displays the logged-in staff member’s details correctly

Actual Result

PASS/FAIL

2.2.8

Scope

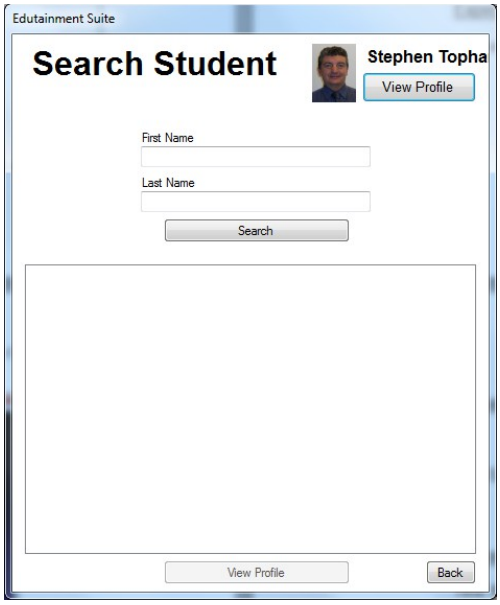
Test that the view student button on the staff home form works correctly

Method

1. Enter “stst11” as a username and “password” as a password
2. Click the ‘View Student’ button in the right

Expected Result

The view student form is displayed



Actual Result

PASS/FAIL

2.2.9

Scope

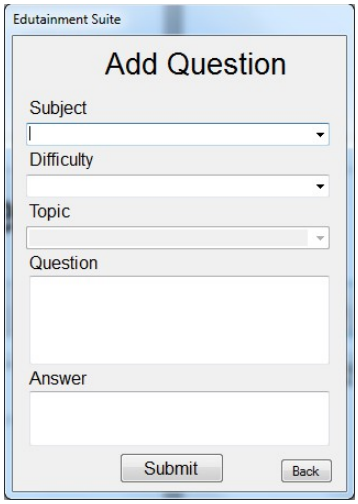
Test that the create question button on the staff home form works correctly

Method

1. Enter “stst11” as a username and “password” as a password
2. Click the ‘Create Question’ button in the right

Expected Result

The create question form is displayed



Actual Result

PASS/FAIL

2.2.10

Scope

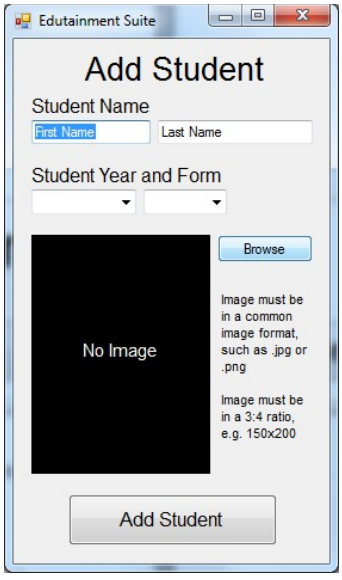
Test that the add student button on the staff home form works correctly

Method

1. Enter “stst11” as a username and “password” as a password
2. Click the ‘Add Student’ button in the right

Expected Result

The add student form is displayed



Actual Result

PASS/FAIL

Section 3: Connect Four

3.1

Scope

Test that the hotseat game button on the Connect Four main menu form works correctly

Method

1. Enter “95mh11” as a username and “password” as a password
2. Click the ‘Connect Four’ button in the top right
3. Click the ‘Hotseat Mode’ button in the top left

Expected Result

The Connect Four hotseat opponent login form is displayed

Actual Result

PASS/FAIL

3.2

Scope

a. Test that the network game button, if there are no games available, will display a message

b. Test that the network game button works correctly

Method

1. Enter “95mh11” as a username and “password” as a password
2. Click the ‘Connect Four’ button in the top right
3. Click the ‘Network Mode’ button in the top left
4. Click the ‘OK’ button

Expected Result

a. The message is displayed
b. The Connect Four network lobby form is displayed

Actual Result

PASS/FAIL

Sub-Section 1: Hotseat Game

3.1.1

Scope

Test that the cancel button on the Connect Four hotseat login form works correctly

Method

1. Load up the Connect Four hotseat login form
2. Click the ‘cancel’ button

Expected Result

The program closes

Actual Result

PASS/FAIL

3.1.2

Scope

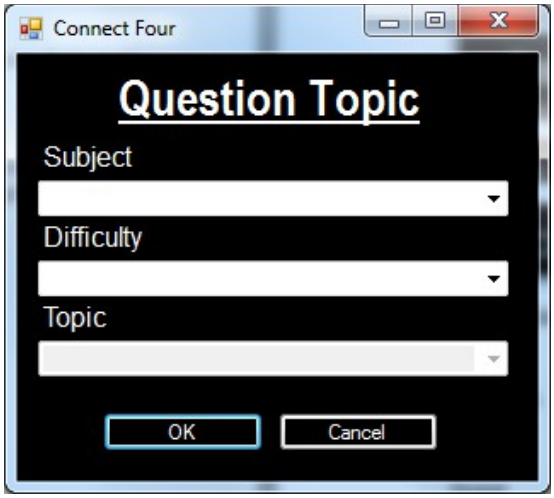
a. Test that the Connect Four hotseat opponent login form, upon receiving invalid credentials, will refuse the login and blank the data entry controls
b. Test that the Connect Four hotseat opponent login form, upon receiving valid teacher credentials, will refuse the login and blank the data entry controls
c. Test that the Connect Four hotseat opponent login form, upon receiving valid student credentials, will display the Connect Four hotseat question selection form

Method

1. Enter “95mh11” as a username and “password” as a password
2. Click the ‘Connect Four’ button in the top right
3. Click the ‘Hotseat Mode’ button in the top left
4. Enter login credentials
    a. Enter invalid login credentials
        i. Enter “abcdef” as a username and password
        ii. Enter “stap11” as a username and “password” as a password
    b. Enter “95lc11” as a username and “password” as a password
5. Click the ‘OK’ button in the bottom left

Expected Result

a. The form will reject the login and blank the entry controls
b. The form will reject the login and blank the entry controls
c. The form will accept the login and proceed onto the Connect Four hotseat question selection form



Actual Result

PASS/FAIL

3.1.3

Scope

Test that the cancel button on the Connect Four hotseat question selection form works correctly

Method

1. Load up the Connect Four hotseat question selection form
2. Click the ‘cancel’ button

Expected Result

The program closes

Actual Result

PASS/FAIL

3.1.4

Scope

a. Test that the topic combobox on the Connect Four hotseat question selection form is disabled as long as there is no subject selected
b. Test that the topic combobox on the Connect Four hotseat question selection form enables when a subject is selected

Method

1. Load up the Connect Four hotseat question selection form
2. Check that the combobox labelled ‘Topic’ is disabled
3. Select ‘History’ from the combobox labelled ‘Subject’
4. Check that the combobox labelled ‘Topic’ is enabled

Expected Result

a. The combobox will be disabled
b. The combobox will be enabled

Actual Result

PASS/FAIL

3.1.5

Scope

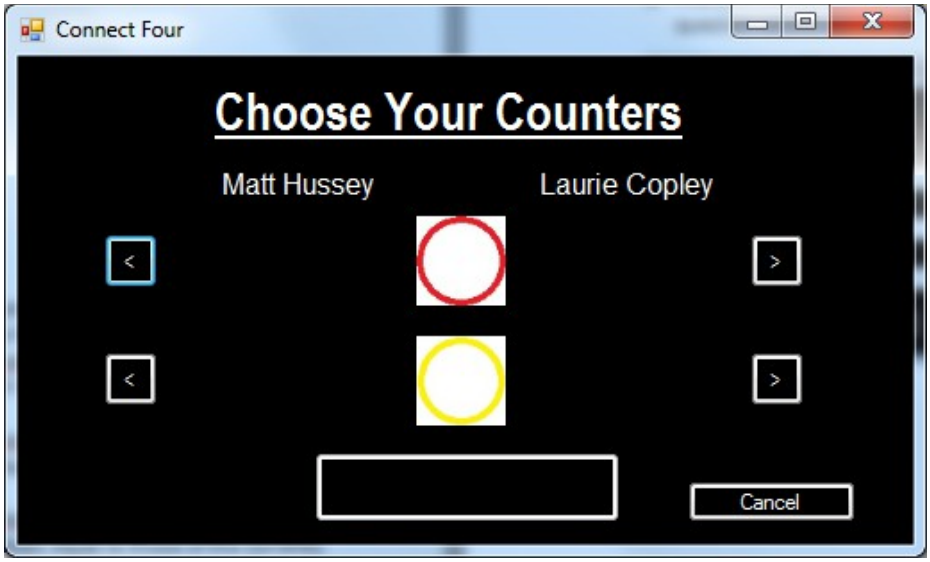
a. Test that the OK button on the Connect Four hotseat question selection form displays an error message if no data is selected
b. Test that the OK button on the Connect Four hotseat question selection form displays an error message if partial data is selected
c. Test that the OK button on the Connect Four hotseat question selection form displays the correct form if data is selected

Method

1. Load up the Connect Four hotseat question selection form
2. Click the ‘OK’ button
3. Select ‘History’ from the combobox labelled ‘Subject’
4. Click the ‘OK’ button
5. Select ‘Year 12’ from the combobox labelled ‘Difficulty’
6. Click the ‘OK’ button
7. Select ‘Russia 1881-1924’ from the combobox labelled ‘Topic’
8. Click the ‘OK’ button

Expected Result

a. The error message will be displayed
b. The error message will be displayed
c. The Connect Four hotseat counter selection form is displayed



Actual Result

PASS/FAIL

3.1.6

Scope

Test that the cancel button on the Connect Four hotseat counter selection form works correctly

Method

1. Load up the Connect Four hotseat counter selection form
2. Click the ‘cancel’ button

Expected Result

The program closes

Actual Result

PASS/FAIL

3.1.7

Scope

a. Test that continue button is disabled when both counters have not been chosen
b. Test that the four counter selection buttons work correctly

   i. Test that the top-left button chooses red for the left player and disables all but the bottom-right button
   ii. Test that the top-right button chooses red for the right player and disables all but the bottom-left button
   iii. Test that the bottom-left button chooses yellow for the left player and disables all but the top-right button
   iv. Test that the bottom-right button chooses yellow for the right player and disables all but the top-left button
   v. Test that the display updates accordingly
c. Test that the continue button is enabled once both counters have been chosen

Method

1. Load up the Connect Four hotseat counter selection form
2. Click the OK button in the bottom centre
3. Click the top-left arrow button
4. Click the top-right arrow button
5. Click the bottom-left arrow button
6. Click the bottom-right arrow button
7. Exit the program, load up the Connect Four hotseat counter selection form
8. Click the top-right arrow button
9. Click the top-left arrow button
10. Click the bottom-right arrow button
11. Click the bottom-left arrow button
12. Exit the program, load up the Connect Four hotseat counter selection form
13. Click the bottom-left arrow button
14. Click the bottom-right arrow button
15. Click the top-left arrow button
16. Click the top-right arrow button
17. Exit the program, load up the Connect Four hotseat counter selection form
18. Click the bottom-right arrow button
19. Click the bottom-left arrow button
20. Click the top-right arrow button
21. Click the top-left arrow button
22. Click the ‘Continue’ button

Expected Result

a. The button will be disabled
b. The counter selection will work as expected, with the display updating accordingly
c. The button will be enabled

Actual Result

PASS/FAIL

3.1.8

Scope

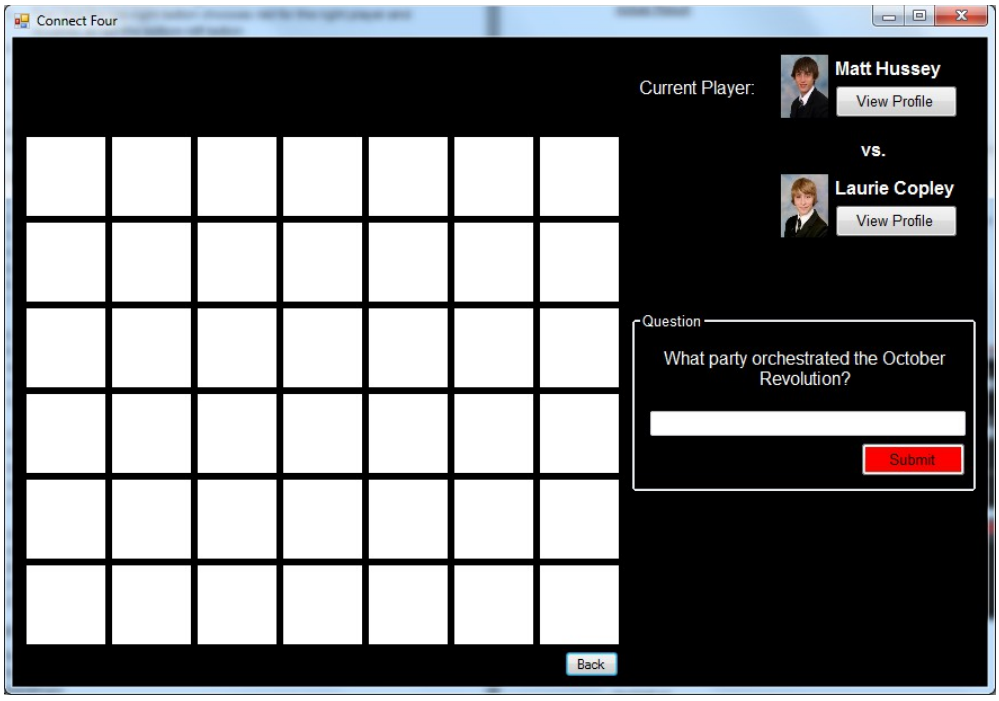
Test that the continue button on the Connect Four hotseat counter selection form works correctly

Method

1. Load up the Connect Four hotseat counter selection form
2. Click the top-left arrow button
3. Click the bottom-right arrow button
4. Click the ‘Continue’ button in the bottom centre

Expected Result

The Connect Four hotseat game form is displayed



Actual Result

PASS/FAIL

3.1.9

Scope

Test that the back button on the Connect Four hotseat game form works correctly

Method

1. Load up the Connect Four hotseat game form
2. Click the ‘back’ button

Expected Result

The Connect Four main menu form loads

Actual Result

PASS/FAIL

3.1.10

Scope

Test that the view player one profile button on the Connect Four hotseat game form works correctly

Method

1. Load up the Connect Four hotseat game form
2. Click the ‘view profile’ button under player one’s name

Expected Result

The view student profile form loads

Actual Result

PASS/FAIL

3.1.11

Scope

Test that the details the view player one profile form is populated with are correct

Method

1. Load up the Connect Four hotseat game form
2. Click the ‘view profile’ button under player one’s name

3. Observe the view profile form

Expected Result

The view player one profile form displays player one's details correctly

Actual Result

PASS/FAIL

3.1.12

Scope

Test that the view player two profile button on the Connect Four hotseat game form works correctly

Method

3. Load up the Connect Four hotseat game form
4. Click the 'view profile' button under player two's name

Expected Result

The view student profile form loads

Actual Result

PASS/FAIL

3.1.13

Scope

Test that the details the view player two profile form is populated with are correct

Method

1. Load up the Connect Four hotseat game form
2. Click the 'view profile' button under player two's name
3. Observe the view profile form

Expected Result

The view player two profile form displays player two's details correctly

Actual Result

PASS/FAIL

3.1.14

Scope

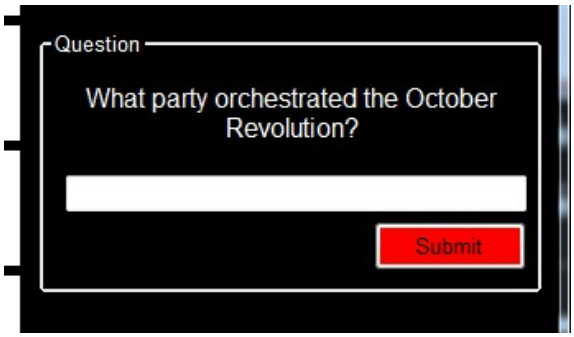
Test that the question section of the Connect Four hotseat game form populates correctly

Method

1. Load up the Connect Four hotseat game form
2. Observe the question section in the bottom right

Expected Result

The question section populates correctly, with a question relevant to the selected subject



Actual Result

PASS/FAIL

3.1.15

Scope

a. Test that, if no answer is given, clicking the submit button works correctly
b. Test that, if the answer given is incorrect, clicking the submit button works correctly
c. Test that, if the answer given is correct, clicking the submit button works correctly

Method

1. Load up the Connect Four hotseat game form

2. Click the ‘Submit’ button in the question section
3. Type “asdf” as an answer
4. Click the ‘Submit button in the question section
5. Type the correct answer to the given question
6. Click the ‘Submit’ button in the question section

Expected Result

a. The answer is flagged as incorrect
b. The answer is flagged as incorrect
c. The answer is flagged as correct

Actual Result

PASS/FAIL

3.1.16

Scope

Test that the player can, after answering a question correctly, choose to place a counter

Method

1. Load up the Connect Four hotseat game form
2. Answer the question correctly
3. Click the ‘Button 1’ button in the top-left
4. Observe the game grid

Expected Result

A red counter will be placed at the bottom of the leftmost column

Actual Result

PASS/FAIL

3.1.17

Scope

a. Test that the current player label moves according to which player is currently up
b. Test that, after one player drops a counter, the player is changed

Method

1. Load up the Connect Four hotseat game form
2. Answer the question correctly
3. Click the ‘Button 1’ button in the top-left
4. Observe the ‘current player’ label in the top-right

Expected Result

a. The label moves correctly
b. The player is changed

Actual Result

PASS/FAIL

3.1.18

Scope

Test that a column, when full, will cap itself off to stop further counters being dropped on it

Method

1. Load up the Connect Four hotseat game form
2. Answer the question correctly
3. Click the ‘Button 1’ button in the top-left
4. Repeat 2-4 until leftmost column full

Expected Result

The counter drop button for the leftmost column will be disabled

Actual Result

PASS/FAIL

3.1.19

Scope

Test that all counter drop buttons work correctly

Method

1. Load up the Connect Four hotseat game form
2. Answer the question correctly
3. Click the ‘Button 1’ button in the top-left

4. Repeat 2-4, but on each button moving along the columns

Expected Result

A counter will be placed in each column, in alternating colours

Actual Result

PASS/FAIL

3.1.20

Scope

a. Test that a player can win by getting four counters of their colour in a row horizontally
b. Test that a player can win by getting four counters of their colour in a row vertically
c. Test that a player can win by getting four counters of their colour in a row diagonally top-left to bottom-right
d. Test that a player can win by getting four counters of their colour in a row diagonally top-right to bottom-left
e. Test that a game can draw if all 42 squares are filled with no wins

Method

1. Load up the Connect Four hotseat game form
2. Get four counters of the same colour in a row
    a. horizontally
    b. vertically
    c. diagonally top-left to bottom-right
    d. diagonally top-right to bottom-left
3. Fill the grid with no wins

Expected Result

All win conditions will register as wins, and the draw a draw, a winner will be declared (if applicable) and the game will end, with the reset button appearing

Actual Result

PASS/FAIL

3.1.21

Scope

Test that both players can win

Method

1. Load up the Connect Four hotseat game form
2. Win a game for red
3. Exit the program, load up the Connect Four hotseat game form
4. Win a game for yellow

Expected Result

Both wins work as expected

Actual Result

PASS/FAIL

3.1.22

Scope

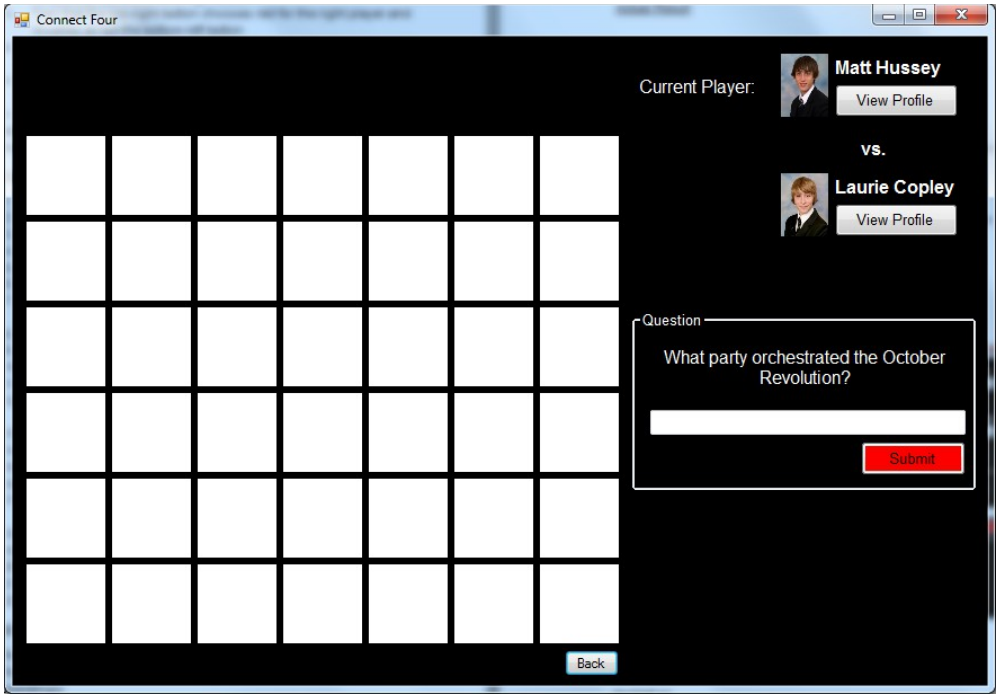
Test that the reset button works correctly

Method

1. Load up the Connect Four hotseat game form
2. Win a game
3. Click the ‘Reset’ button that appears
4. Observe the form

Expected Result

The form is returned to its default state



Actual Result

PASS/FAIL

3.1.23

Scope

Test that a player can win after resetting a previously finished game

Method

1. Load up the Connect Four hotseat game form
2. Win a game
3. Click the ‘Reset button that appears
4. Win another game

Expected Result

All win conditions will register as wins, a winner will be declared and the game will end, with the reset button appearing again

Actual Result

PASS/FAIL

3.1.24

Scope

a. Test that the database is correctly updated when a question is answered
b. Test that the game breakdown text files for both players are correctly updated after a game
c. Test that the database is correctly updated after a game

Method

1. Load up the Connect Four hotseat game form
2. Answer a question correctly
3. Observe 'tblAttempted' in the database
4. Win a game for red
5. Observe the 96mh11.txt and 95lc11.txt text files at G:\Computing Group\Y10\2009-10\Ben's\Edutainment Suite\Connect4\bin\Debug\ WinsLosses
6. Observe the records for 'Matthew Hussey' and 'Laurie Copley' in 'tblStudents' in the database

Expected Result

a. A record is added to tblAttempted when a question is answered
b. The text files for both players are correctly updated
c. The records for both players in tblStudents are changed accordingly

Actual Result

PASS/FAIL

Sub-Section 2: Network Game

3.2.1

Scope

a. Test that the game list populates correctly on the Connect Four network lobby form
b. Test that, if the logged-in user is currently playing a game, it will be displayed in that game's entry in the listbox

Method

1. Load up the Connect Four network lobby form
2. Observe the game list listbox

Expected Result

a. The listbox is populated with all the currently ongoing games that have a free space or that the logged-in user is currently playing in
b. Games the logged-in user is currently a part of have a note displayed in the listbox

Actual Result

PASS/FAIL

3.2.2

Scope

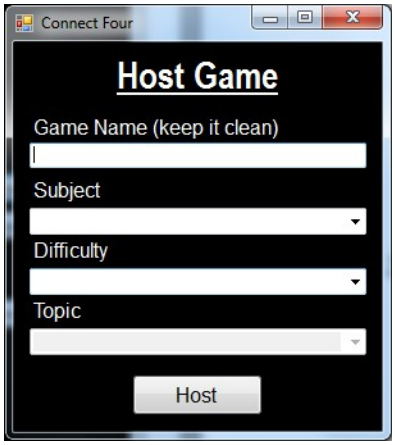
Test that the host game button on the Connect Four network lobby form works correctly

Method

1. Load up the Connect Four network lobby form
2. Click the 'Host Game' button

Expected Result

The Connect Four network host game form is displayed



Actual Result

PASS/FAIL

3.2.3

Scope

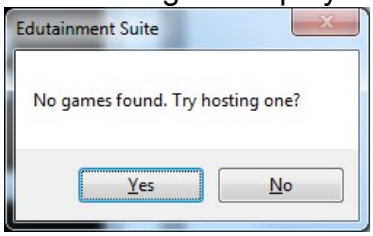
a. Test that, if the logged-in user has made no games, the your games button will display a message
    i. Test that the no button works correctly
    ii. Test that the yes button works correctly
b. Test that , if the logged-in user has made a game, the your games button works correctly

Method

1. Load up the Connect Four network lobby form
2. Before making a game
    a. Click the ‘Your Games’ button
    b. Click the ‘No’ button
    c. Click the ‘Your Games’ button
    d. Click the ‘Yes’ button
3. After making a game (see 3.2.6)
    a. Click the ‘Your Games’ button

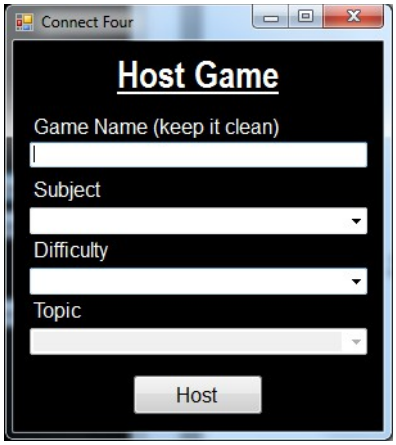
Expected Result

a. The message is displayed



    i. The message closes and the Connect Four network your games form is displayed

    ii. The message closes and the Connect Four network your games and Connect Four network host game forms are displayed



b. The Connect Four network your games form is displayed

Actual Result

PASS/FAIL

3.2.4

Scope

Test that the Connect Four network host game form is always on top of any other programs

Method

1. Load up the Connect Four network host game form
2. Open up another program

Expected Result

The Connect Four network host game form is always on top of any other programs

Actual Result

PASS/FAIL

3.2.4

Scope

Test that the subject combobox on the Connect Four network host game form populates correctly

Method

1. Load up the Connect Four network host game form
2. Observe the combobox labelled ‘Subject’

Expected Result

The combobox will be populated with the subjects from the database correctly

Actual Result

PASS/FAIL

3.2.5

Scope

a. Test that the topic combobox on the Connect Four network host game form is disabled as long as there is no subject selected
b. Test that the topic combobox on the Connect Four network host game form enables when a subject is selected

Method

1. Load up the Connect Four network host game form
2. Check that the combobox labelled ‘Topic’ is disabled
3. Select ‘History’ from the combobox labelled ‘Subject’
4. Check that the combobox labelled ‘Topic’ is enabled

Expected Result

a. The combobox will be disabled
b. The combobox will be enabled

Actual Result

PASS/FAIL

3.2.6

Scope

a. Test that the host button on the Connect Four network host game form displays an error message if no data is inputted
b. Test that the host button on the Connect Four network host game form displays an error message if partial data is inputted

c. Test that the host button on the Connect Four network host game form creates the game if valid data is inputted
d. Test that the lobby form updates to reflect the new game

Method

1. Load up the Connect Four network host game form
2. Click the ‘Host’ button
3. Enter “TestGame” as a Game Name
4. Click the ‘Host’ button
5. Select ‘History’ from the combobox labelled ‘Subject’
6. Click the ‘Host’ button
7. Select ‘Year 12’ from the combobox labelled ‘Difficulty’
8. Click the ‘Host’ button
9. Select ‘Russia 1881-1924’ from the combobox labelled ‘Topic’
10. Click the ‘Host’ button
11. Click the ‘Yes’ button
12. Observe ‘tblConnect4’ in the database
13. Repeat with Game Name “TestGame2”
14. Observe the lobby form

Expected Result

a. The error message will be displayed
b. The error message will be displayed
c. The game will be created
d. The form updates

Actual Result

PASS/FAIL

3.2.7

Scope

a. Test that the game list populates correctly on the Connect Four network your games form
b. Test that, if a game has an opponent, it will be displayed in that game’s entry in the listbox
c. Test that the lobby updates when a new game is created

Method

1. Load up the Connect Four network your games form
2. Observe the game list listbox
3. Create a game
4. Observe the listbox

Expected Result

a. The listbox is populated with all the currently ongoing games that have a free space or that the logged-in user is currently playing in
b. Games that have an opponent in have a note displayed in the listbox
c. The lobby updates to reflect newly-created games

Actual Result

PASS/FAIL

3.2.8

Scope

Test that the refresh button on the Connect Four network lobby form works correctly

Method

1. Load up the Connect Four network lobby form
2. Create a new game (see 3.2.6)
3. Click the ‘Refresh’ button
4. Observe the game list listbox

Expected Result

The game list listbox is updated

Actual Result

PASS/FAIL

3.2.9

Scope

Test that the refresh button on the Connect Four network your games form works correctly

Method

1. Load up the Connect Four network your games form
2. Create a new game (see 3.2.6)
3. Click the ‘Refresh’ button
4. Observe the game list listbox

Expected Result

The game list listbox is updated

Actual Result

PASS/FAIL

3.2.10

Scope

Test that the host game button on the Connect Four network your games form works correctly

Method

1. Load up the Connect Four network your games form
2. Click the ‘Host Game’ button

Expected Result

The Connect Four network host game form is displayed

Actual Result

PASS/FAIL

3.2.11

Scope

a. Test that, if there is no opponent in a game the logged-in user is host of, the join game button on the Connect Four network lobby form will display an error message

b. Test that, if the logged-in user isn't the host of a game, the join game button on the Connect Four network lobby form works correctly
c. Test that, if there is an opponent in a game the logged-in user is host of, the join game button on the Connect Four network lobby form works correctly

Method

1. Log in with "95mh11" and "password"
2. Load up the Connect Four network lobby form
3. Click on the TestGame game in the game list listbox
4. Click the 'Join Game' button
5. Click the 'OK' button
6. Close the program
7. Log in with "95lc11" and "password"
8. Load up the Connect Four network lobby form
9. Click on the TestGame game in the game list listbox
10. Click the 'Join Game' button
11. Close the program
12. Log in with "95mh11" and "password"
13. Load up the Connect Four network lobby form
14. Click on the TestGame game in the game list listbox
15. Click the 'Join Game' button

Expected Result

a. The error message will be displayed
b. The database is updated about the new player in the game and the Connect Four network game form is displayed
c. The Connect Four network game form is displayed

Actual Result

PASS/FAIL

3.2.12

Scope

a. Test that, if there is no opponent in a game the logged-in user is host of, the join game button on the Connect Four network your games form will display an error message
b. Test that, if there is an opponent in a game the logged-in user is host of, the join game button on the Connect Four network your games form works correctly

Method

1. Log in with “95mh11” and “password”
2. Load up the Connect Four network your games form
3. Click on the TestGame2 game in the game list listbox
4. Click the ‘Join Game’ button
5. Click the ‘OK’ button
6. Close the program
7. Log in with “95lc11” and “password”
8. Load up the Connect Four network lobby form
9. Click on the TestGame2 game in the game list listbox
10. Click the ‘Join Game’ button
11. Close the program
12. Log in with “95mh11” and “password”
13. Load up the Connect Four network your games form
14. Click on the TestGame2 game in the game list listbox
15. Click the ‘Join Game’ button

Expected Result

a. The error message will be displayed
b. The Connect Four network game form is displayed

Actual Result

PASS/FAIL

3.2.13

Scope

Test that the back button on the Connect Four network game form works correctly

Method

1. Load up the Connect Four network game form
2. Click the ‘back’ button

Expected Result

The Connect Four main menu form loads

Actual Result

PASS/FAIL

3.2.14

Scope

Test that the view player one profile button on the Connect Four network game form works correctly

Method

1. Load up the Connect Four network game form
2. Click the ‘view profile’ button under player one’s name

Expected Result

The view student profile form loads

Actual Result

PASS/FAIL

3.2.15

Scope

Test that the details the view player one profile form is populated with are correct

Method

1. Load up the Connect Four network game form
2. Click the ‘view profile’ button under player one’s name
3. Observe the view profile form

Expected Result

The view player one profile form displays player one’s details correctly

Actual Result

PASS/FAIL

3.2.16

Scope

Test that the view player two profile button on the Connect Four network game form works correctly

Method

1. Load up the Connect Four network game form
2. Click the ‘view profile’ button under player two’s name

Expected Result

The view student profile form loads

Actual Result

PASS/FAIL

3.2.17

Scope

Test that the details the view player two profile form is populated with are correct

Method

1. Load up the Connect Four network game form
2. Click the ‘view profile’ button under player two’s name
3. Observe the view profile form

Expected Result

The view player two profile form displays player two’s details correctly

Actual Result

PASS/FAIL

3.2.18

Scope

Test that the question section of the Connect Four network game form populates correctly

Method

1. Load up the Connect Four network game form
2. Observe the question section in the bottom right

Expected Result

The question section populates correctly, with a question relevant to the selected subject

Actual Result

PASS/FAIL

3.2.19

Scope

Test that the question section of the Connect Four network game form is only visible to the current player

Method

1. Load up the Connect Four network game form twice, one as “95mh11” and one as “95lc11”
2. Load up the Connect Four network game form
3. Observe the question section in the bottom right

Expected Result

The question section is only visible to the current player

Actual Result

PASS/FAIL

3.2.20

Scope

Test that the question submit button works correctly

Method

1. Load up the Connect Four network game form twice, one as “95mh11” and one as “95lc11”
2. Click the ‘Submit’ button in the question section
3. Type “asdf” as an answer
4. Click the ‘Submit button in the question section
5. Type the correct answer to the given question
6. Click the ‘Submit’ button in the question section

Expected Result

If the answer given is incorrect, or no answer is given, the entry will be marked incorrect and the next player will be up. If the answer given is correct, the entry will be marked correct and the current player may drop a counter

Actual Result

PASS/FAIL

3.2.21

Scope

Test that the player can, after answering a question correctly, choose to place a counter

Method

1. Load up the Connect Four network game form
2. Answer the question correctly
3. Click the ‘Button 1’ button in the top-left
4. Observe the game grid

Expected Result

A red counter will be placed at the bottom of the leftmost column

Actual Result

PASS/FAIL

3.2.22

Scope

a. Test that the current player label moves according to which player is currently up
b. Test that, after one player drops a counter, the player is changed

Method

1. Load up the Connect Four network game form
2. Answer the question correctly
3. Click the ‘Button 1’ button in the top-left
4. Observe the ‘current player’ label in the top-right

Expected Result

a. The label moves correctly
b. The player is changed

Actual Result

PASS/FAIL

3.2.23

Scope

Test that a column, when full, will cap itself off to stop further counters being dropped on it

Method

1. Load up the Connect Four network game form twice, one as “95mh11” and one as “95lc11”
2. Answer the question correctly
3. Click the ‘Button 1’ button in the top-left
4. Repeat 2-4 until leftmost column full

Expected Result

The counter drop button for the leftmost column will be disabled

Actual Result

PASS/FAIL

3.2.24

Scope

Test that the game updates correctly to display the other user’s moves

Method

1. Load up the Connect Four network game form twice, one as “95mh11” and one as “95lc11”
2. Answer the question correctly
3. Click the ‘Button 1’ button in the top-left
4. Observe the grid as 95lc11

Expected Result

The dropping of the counter will be reflected on 95lc11's screen almost instantaneously

Actual Result

PASS/FAIL

3.2.25

Scope

Test that all counter drop buttons work correctly

Method

1. Load up the Connect Four network game form twice, one as "95mh11" and one as "95lc11"
2. Answer the question correctly
3. Click the 'Button 1' button in the top-left
4. Repeat 2-4, but on each button moving along the columns

Expected Result

A counter will be placed in each column, in alternating colours

Actual Result

PASS/FAIL

3.2.26

Scope

a. Test that a player can win by getting four counters of their colour in a row horizontally
b. Test that a player can win by getting four counters of their colour in a row vertically
c. Test that a player can win by getting four counters of their colour in a row diagonally top-left to bottom-right
d. Test that a player can win by getting four counters of their colour in a row diagonally top-right to bottom-left
e. Test that a game can draw if all 42 squares are filled with no wins

Method

1. Load up the Connect Four network game form twice, one as “95mh11” and one as “95lc11”
2. Get four counters of the same colour in a row
    a. horizontally
    b. vertically
    c. diagonally top-left to bottom-right
    d. diagonally top-right to bottom-left
3. Fill the grid with no wins

Expected Result

All win conditions will register as wins, and the draw a draw, a winner will be declared (if applicable) and the game will end, with the reset button appearing

Actual Result

PASS/FAIL

3.2.27

Scope

Test that both players can win

Method

1. Load up the Connect Four network game form twice, one as “95mh11” and one as “95lc11”
2. Win a game for red
4. Exit the program, load up the Connect Four network game form twice, one as “95mh11” and one as “95lc11”
3. Win a game for yellow

Expected Result

Both wins work as expected

Actual Result

PASS/FAIL

3.2.28

Scope

Test that the quit button works correctly

Method

1. Load up the Connect Four network game form twice, one as “95mh11” and one as “95lc11”
2. Win a game
3. Click the ‘Quit’ button that appears
4. Observe the database

Expected Result

The Connect Four network lobby form is displayed and the record for the completed game is deleted from the database

Actual Result

PASS/FAIL

3.2.29

Scope

a. Test that the database is correctly updated when a question is answered
b. Test that the game breakdown text files for both players are correctly updated after a game
c. Test that the database is correctly updated after a game

Method

1. Load up the Connect Four network game form twice, one as “95mh11” and one as “95lc11”
2. Answer a question correctly
3. Observe ‘tblAttempted’ in the database
4. Win a game for red
5. Observe the 96mh11.txt and 95lc11.txt text files at G:\Computing Group\Y10\2009-10\Ben's\Edutainment Suite\Connect4\bin\Debug\ WinsLosses
6. Observe the records for ‘Matthew Hussey’ and ‘Laurie Copley’ in ‘tblStudents’ in the database

Expected Result

a. A record is added to tblAttempted when a question is answered
b. The text files for both players are correctly updated
c. The records for both players in tblStudents are changed accordingly

Actual Result

PASS/FAIL

Section 4: Noughts and Crosses

4.1

Scope

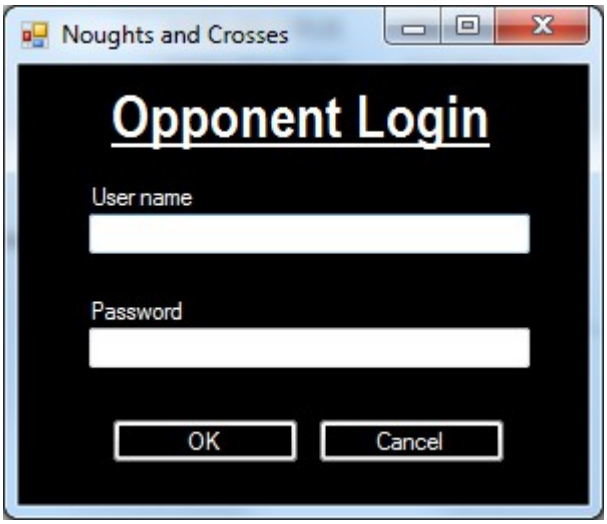
Test that the hotseat game button on the Noughts and Crosses main menu form works correctly

Method

1. Enter “95mh11” as a username and “password” as a password
2. Click the ‘Noughts and Crosses’ button in the top right
3. Click the ‘Hotseat Mode’ button in the top left

Expected Result

The Noughts and Crosses hotseat opponent login form is displayed



Actual Result

PASS/FAIL

4.2

Scope

Test that the cancel button on the Noughts and Crosses hotseat login form works correctly

Method

1. Load up the Noughts and Crosses hotseat login form
2. Click the ‘cancel’ button

Expected Result

The program closes

Actual Result

PASS/FAIL

4.3

Scope

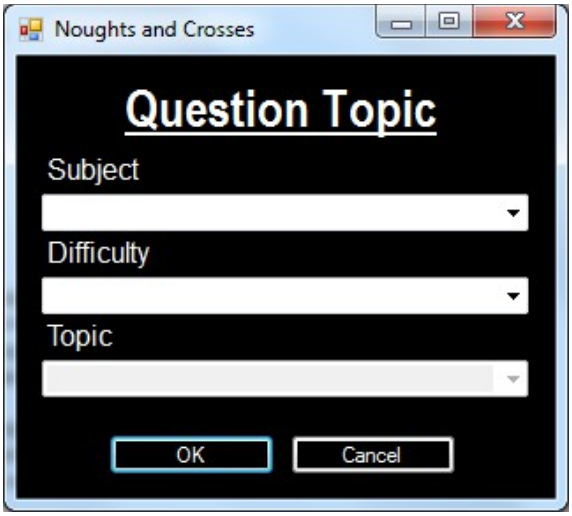
a. Test that the Noughts and Crosses hotseat opponent login form, upon inputting of incorrect login credentials, will refuse the login
b. Test that the Noughts and Crosses hotseat opponent login form, upon receiving valid teacher credentials, will refuse the login and blank the data entry controls
c. Test that the Noughts and Crosses hotseat opponent login form, upon receiving valid student credentials, will display the Noughts and Crosses hotseat question selection form

Method

1. Enter “95mh11” as a username and “password” as a password
2. Click the ‘Noughts and Crosses’ button in the top right
3. Click the ‘Hotseat Mode’ button in the top left
4. Enter login credentials
    a. Enter invalid login credentials
        i. Enter “abcdef” as a username and password
        ii. Enter “stap11” as a username and “password” as a password
    b. Enter “95lc11” as a username and “password” as a password
5. Click the ‘OK’ button in the bottom left

Expected Result

a. The form will reject the login and blank the entry controls
b. The form will reject the login and blank the entry controls
c. The form will accept the login and proceed onto the Noughts and Crosses hotseat question selection form



Actual Result

PASS/FAIL

4.4

Scope

Test that the cancel button on the Noughts and Crosses hotseat question selection form works correctly

Method

1. Load up the Noughts and Crosses hotseat question selection form
2. Click the ‘cancel’ button

Expected Result

The program closes

Actual Result

PASS/FAIL

4.5

Scope

a. Test that the topic combobox on the Noughts and Crosses hotseat question selection form is disabled as long as there is no subject selected
b. Test that the topic combobox on the Noughts and Crosses hotseat question selection form enables when a subject is selected

Method

1. Load up the Noughts and Crosses hotseat question selection form
2. Check that the combobox labelled ‘Topic’ is disabled
3. Select ‘History’ from the combobox labelled ‘Subject’
4. Check that the combobox labelled ‘Topic’ is enabled

Expected Result

a. The combobox will be disabled
b. The combobox will be enabled

Actual Result

PASS/FAIL

4.6

Scope

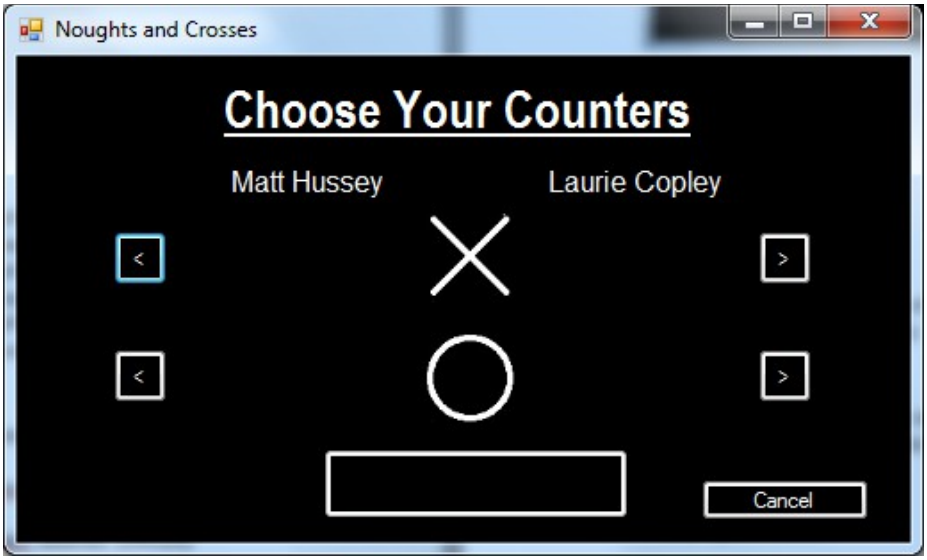
a. Test that the OK button on the Noughts and Crosses hotseat question selection form displays an error message if no data is selected
b. Test that the OK button on the Noughts and Crosses hotseat question selection form displays an error message if partial data is selected
c. Test that the OK button on the Noughts and Crosses hotseat question selection form displays the correct form if data is selected

Method

1. Load up the Noughts and Crosses hotseat question selection form
2. Click the ‘OK’ button
3. Select ‘History’ from the combobox labelled ‘Subject’
4. Click the ‘OK’ button
5. Select ‘Year 12’ from the combobox labelled ‘Difficulty’
6. Click the ‘OK’ button
7. Select ‘Russia 1881-1924’ from the combobox labelled ‘Topic’
8. Click the ‘OK’ button

Expected Result

a. The error message will be displayed
b. The error message will be displayed
c. The Noughts and Crosses hotseat counter selection form is displayed



Actual Result

PASS/FAIL

4.7

Scope

Test that the cancel button on the Noughts and Crosses hotseat counter selection form works correctly

Method

1. Load up the Noughts and Crosses hotseat counter selection form
2. Click the ‘cancel’ button

Expected Result

The program closes

Actual Result

PASS/FAIL

4.8

Scope

a. Test that continue button is disabled when both counters have not been chosen
b. Test that the four counter selection buttons work correctly

   i. Test that the top-left button chooses X for the left player and disables all but the bottom-right button
   ii. Test that the top-right button chooses X for the right player and disables all but the bottom-left button
   iii. Test that the bottom-left button chooses O for the left player and disables all but the top-right button
   iv. Test that the bottom-right button chooses O for the right player and disables all but the top-left button
   v. Test that the display updates accordingly
c. Test that the continue button is enabled once both counters have been chosen

Method

1. Load up the Noughts and Crosses hotseat counter selection form
2. Click the OK button in the bottom centre
3. Click the top-left arrow button
4. Click the top-right arrow button
5. Click the bottom-left arrow button
6. Click the bottom-right arrow button
7. Exit the program, load up the Noughts and Crosses hotseat counter selection form
8. Click the top-right arrow button
9. Click the top-left arrow button
10. Click the bottom-right arrow button
11. Click the bottom-left arrow button
12. Exit the program, load up the Noughts and Crosses hotseat counter selection form
13. Click the bottom-left arrow button
14. Click the bottom-right arrow button
15. Click the top-left arrow button
16. Click the top-right arrow button
17. Exit the program, load up the Noughts and Crosses hotseat counter selection form
18. Click the bottom-right arrow button
19. Click the bottom-left arrow button
20. Click the top-right arrow button
21. Click the top-left arrow button
22. Click the ‘Continue’ button

Expected Result

a. The button will be disabled
b. The counter selection will work as expected, with the display updating accordingly
c. The button will be enabled

Actual Result

PASS/FAIL

4.9

Scope

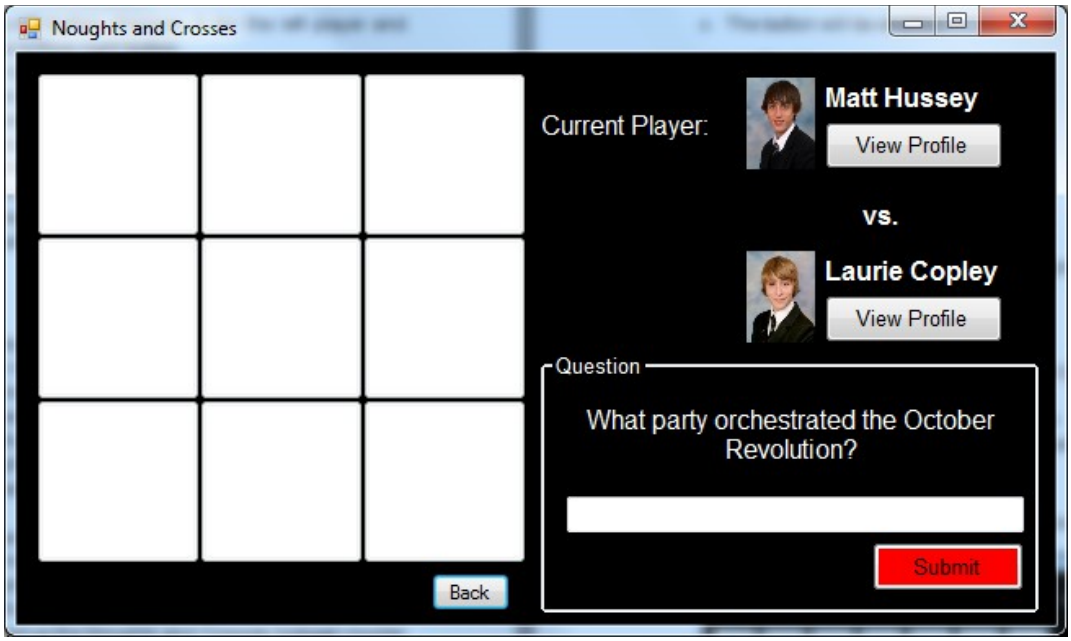
Test that the continue button on the Noughts and Crosses hotseat counter selection form works correctly

Method

1. Load up the Noughts and Crosses hotseat counter selection form
2. Click the top-left arrow button
3. Click the bottom-right arrow button
4. Click the ‘Continue’ button in the bottom centre

Expected Result

The Noughts and Crosses hotseat game form is displayed



Actual Result

PASS/FAIL

4.10

Scope

Test that the back button on the Noughts and Crosses hotseat game form works correctly

Method

1. Load up the Noughts and Crosses hotseat game form
2. Click the ‘back’ button

Expected Result

The Noughts and Crosses main menu form loads

Actual Result

PASS/FAIL

4.11

Scope

Test that the view player one profile button on the Noughts and Crosses hotseat game form works correctly

Method

1. Load up the Noughts and Crosses hotseat game form
2. Click the ‘view profile’ button under player one’s name

Expected Result

The view student profile form loads

Actual Result

PASS/FAIL

4.12

Scope

Test that the details the view player one profile form is populated with are correct

Method

1. Load up the Noughts and Crosses hotseat game form
2. Click the ‘view profile’ button under player one’s name

3. Observe the view profile form

Expected Result

The view player one profile form displays player one's details correctly

Actual Result

PASS/FAIL

4.13

Scope

Test that the view player two profile button on the Noughts and Crosses hotseat game form works correctly

Method

1. Load up the Noughts and Crosses hotseat game form
2. Click the 'view profile' button under player two's name

Expected Result

The view student profile form loads

Actual Result

PASS/FAIL

4.14

Scope

Test that the details the view player two profile form is populated with are correct

Method

1. Load up the Noughts and Crosses hotseat game form
2. Click the 'view profile' button under player two's name
3. Observe the view profile form

Expected Result

The view player two profile form displays player two's details correctly

Actual Result

PASS/FAIL

4.15

Scope

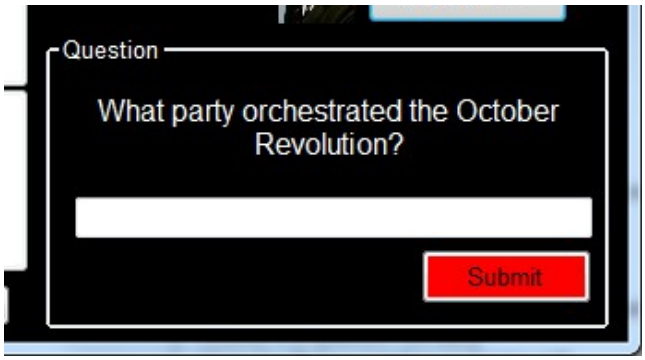
Test that the question section of the Noughts and Crosses hotseat game form populates correctly

Method

1. Load up the Noughts and Crosses hotseat game form
2. Observe the question section in the bottom right

Expected Result

The question section populates correctly, with a question relevant to the selected subject



Actual Result

PASS/FAIL

4.16

Scope

a. Test that, if no answer is given, clicking the submit button works correctly
b. Test that, if the answer given is incorrect, clicking the submit button works correctly
c. Test that, if the answer given is correct, clicking the submit button works correctly

Method

1. Load up the Noughts and Crosses hotseat game form

2. Click the ‘Submit’ button in the question section
3. Type “asdf” as an answer
4. Click the ‘Submit button in the question section
5. Type the correct answer to the given question
6. Click the ‘Submit’ button in the question section

Expected Result

a. The answer is flagged as incorrect
b. The answer is flagged as incorrect
c. The answer is flagged as correct

Actual Result

PASS/FAIL

4.17

Scope

Test that the player can, after answering a question correctly, choose to place a counter

Method

1. Load up the Noughts and Crosses hotseat game form
2. Answer the question correctly
3. Click the centre button on the game grid
4. Observe the game grid

Expected Result

An X counter will be placed on the centre square of the grid

Actual Result

PASS/FAIL

4.18

Scope

a. Test that the current player label moves according to which player is currently up
b. Test that, after one player drops a counter, the player is changed

Method

1. Load up the Noughts and Crosses hotseat game form
2. Answer the question correctly
3. Click the centre button on the game grid
4. Observe the ‘current player’ label in the top-right

Expected Result

a. The label moves correctly
b. The player is changed

Actual Result

PASS/FAIL

4.19

Scope

Test that, when a counter has been placed on it already, a square cannot be clicked again

Method

1. Load up the Noughts and Crosses hotseat game form
2. Answer the question correctly
3. Click the centre button on the game grid
4. Answer the question correctly
5. Click the centre button on the game grid

Expected Result

The buttons will be disabled

Actual Result

PASS/FAIL

4.20

Scope

Test that all counter drop buttons work correctly

Method

1. Load up the Noughts and Crosses hotseat game form
2. Answer the question correctly

3. Click the centre button on the game grid
4. Repeat 2-4, but on each button

Expected Result

A counter will be placed on each square clicked

Actual Result

PASS/FAIL

4.21

Scope

Test that counters cannot be placed whilst a question is being asked

Method

1. Load up the Noughts and Crosses hotseat game form
2. Click the centre button on the game grid

Expected Result

A counter will not be placed

Actual Result

PASS/FAIL

4.22

Scope

a. Test that a player can win by getting three of their counters in a row horizontally
b. Test that a player can win by getting three of their counters in a row vertically
c. Test that a player can win by getting three of their counters in a row diagonally top-left to bottom-right
d. Test that a player can win by getting three of their counters in a row diagonally top-right to bottom-left
e. Test that a game can draw if all 9 squares are filled with no wins

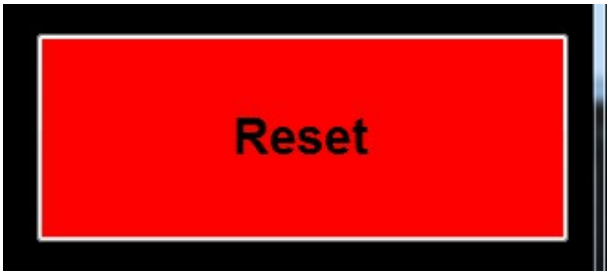
Method

1. Load up the Noughts and Crosses hotseat game form

2. Get three of the same counter in a row
    a. horizontally
    b. vertically
    c. diagonally top-left to bottom-right
    d. diagonally top-right to bottom-left
3. Fill the grid with no wins

Expected Result

All win conditions will register as wins, and the draw a draw, a winner will be declared (if applicable) and the game will end, with the reset button appearing



Actual Result

PASS/FAIL

4.23

Scope

Test that both players can win

Method

1. Load up the Noughts and Crosses hotseat game form
2. Win a game for X
3. Exit the program, load up the Noughts and Crosses hotseat game form
4. Win a game for O

Expected Result

Both wins work as expected

Actual Result

PASS/FAIL

4.24

Scope

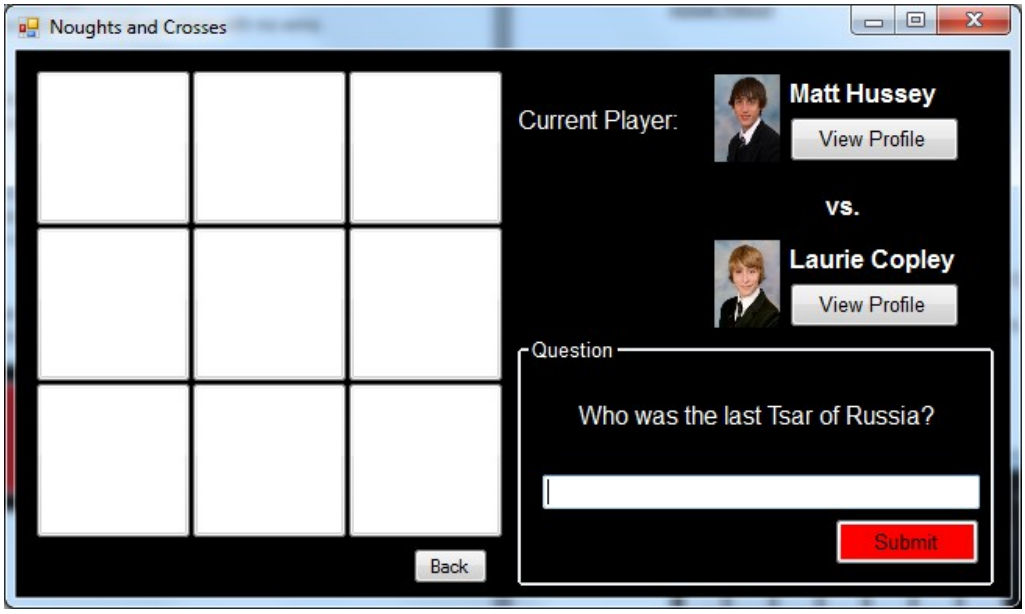
Test that the reset button works correctly

Method

1. Load up the Noughts and Crosses hotseat game form
2. Win a game
3. Click the ‘Reset’ button that appears
4. Observe the form

Expected Result

The form is returned to its default state



Actual Result

PASS/FAIL

4.25

Scope

Test that a player can win after resetting a previously finished game

Method

1. Load up the Noughts and Crosses hotseat game form
2. Win a game
3. Click the ‘Reset button that appears
4. Win another game

Expected Result

All win conditions will register as wins, a winner will be declared and the game will end, with the reset button appearing again

Actual Result

PASS/FAIL

4.26

Scope

a. Test that the database is correctly updated when a question is answered
b. Test that the game breakdown text files for both players are correctly updated after a game
c. Test that the database is correctly updated after a game

Method

1. Load up the Noughts and Crosses hotseat game form
2. Answer a question correctly
3. Observe ‘tblAttempted’ in the database
4. Win a game for red
5. Observe the 96mh11.txt and 95lc11.txt text files at G:\Computing Group\Y10\2009-10\Ben's\Edutainment Suite\Connect4\bin\Debug\WinsLosses
6. Observe the records for ‘Matthew Hussey’ and ‘Laurie Copley’ in ‘tblStudents’ in the database

Expected Result

a. A record is added to tblAttempted when a question is answered
b. The text files for both players are correctly updated
c. The records for both players in tblStudents are changed accordingly

Actual Result

PASS/FAIL

Section 5: Rock, Paper, Scissors

5.1

Scope

Test that the hotseat game button on the Rock, Paper, Scissors main menu form works correctly

Method

1. Enter “95mh11” as a username and “password” as a password
2. Click the ‘Rock, Paper, Scissors’ button in the top right
3. Click the ‘Hotseat Mode’ button in the top left

Expected Result

The Rock, Paper, Scissors hotseat opponent login form is displayed

Actual Result

PASS/FAIL

5.2

Scope

Test that the cancel button on the Rock, Paper, Scissors hotseat login form works correctly

Method

1. Load up the Rock, Paper, Scissors hotseat login form
2. Click the ‘cancel’ button

Expected Result

The program closes

Actual Result

PASS/FAIL

5.3

Scope

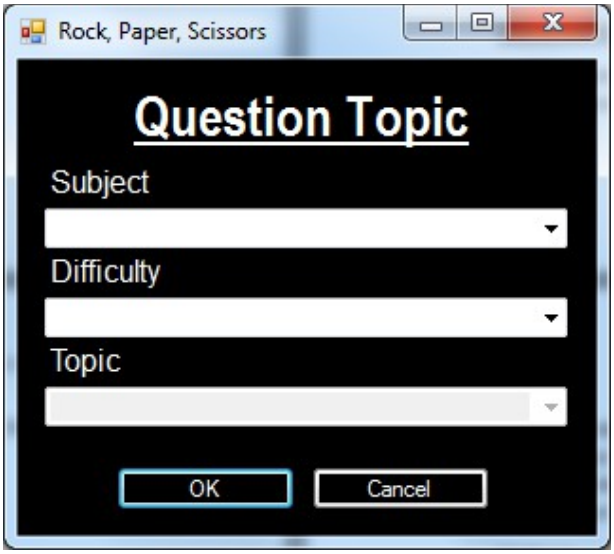
a. Test that the Rock, Paper, Scissors hotseat opponent login form, upon inputting of incorrect login credentials, will refuse the login
b. Test that the Rock, Paper, Scissors hotseat opponent login form, upon receiving valid teacher credentials, will refuse the login and blank the data entry controls
c. Test that the Rock, Paper, Scissors hotseat opponent login form, upon receiving valid student credentials, will display the Rock, Paper, Scissors hotseat question selection form

Method

1. Enter “95mh11” as a username and “password” as a password
2. Click the ‘Rock, Paper, Scissors’ button in the top right
3. Click the ‘Hotseat Mode’ button in the top left
4. Enter login credentials
    a. Enter invalid login credentials
        i. Enter “abcdef” as a username and password
        ii. Enter “stap11” as a username and “password” as a password
    b. Enter “95lc11” as a username and “password” as a password
5. Click the ‘OK’ button in the bottom left

Expected Result

a. The form will reject the login and blank the entry controls
b. The form will reject the login and blank the entry controls
c. The form will accept the login and proceed onto the Rock, Paper, Scissors hotseat question selection form



Actual Result

PASS/FAIL

5.4

Scope

Test that the cancel button on the Rock, Paper, Scissors hotseat question selection form works correctly

Method

1. Load up the Rock, Paper, Scissors hotseat question selection form
2. Click the ‘cancel’ button

Expected Result

The program closes

Actual Result

PASS/FAIL

5.5

Scope

a. Test that the topic combobox on the Rock, Paper, Scissors hotseat question selection form is disabled as long as there is no subject selected

b. Test that the topic combobox on the Rock, Paper, Scissors hotseat question selection form enables when a subject is selected

Method

1. Load up the Rock, Paper, Scissors hotseat question selection form
2. Check that the combobox labelled ‘Topic’ is disabled
3. Select ‘History’ from the combobox labelled ‘Subject’
4. Check that the combobox labelled ‘Topic’ is enabled

Expected Result

a. The combobox will be disabled
b. The combobox will be enabled

Actual Result

PASS/FAIL

5.6

Scope

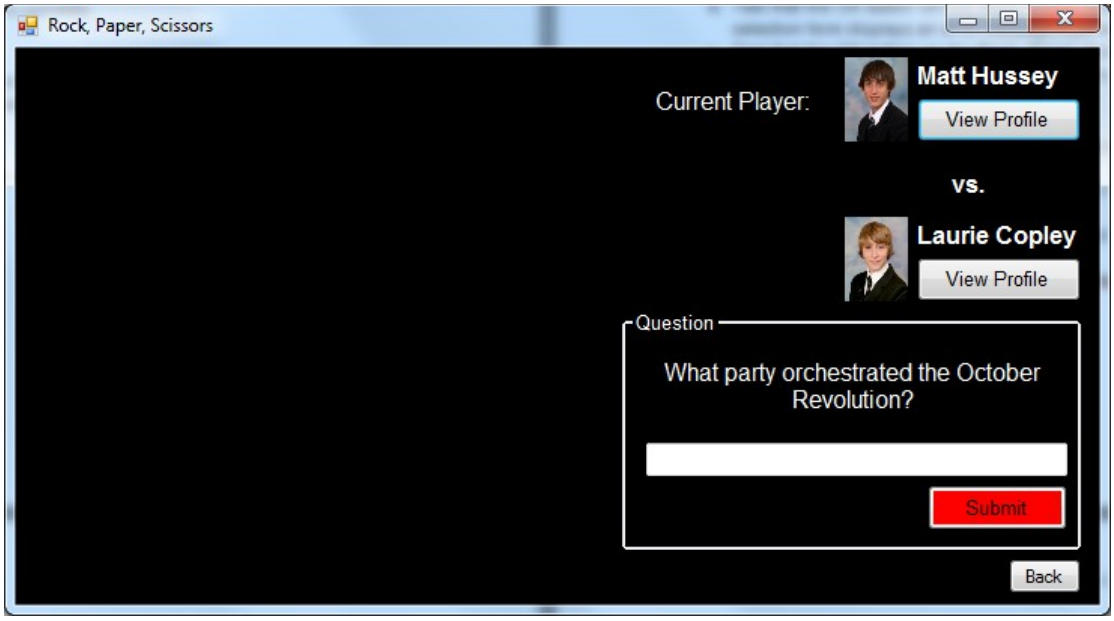
a. Test that the OK button on the Rock, Paper, Scissors hotseat question selection form displays an error message if no data is selected
b. Test that the OK button on the Rock, Paper, Scissors hotseat question selection form displays an error message if partial data is selected
c. Test that the OK button on the Rock, Paper, Scissors hotseat question selection form displays the correct form if data is selected

Method

1. Load up the Rock, Paper, Scissors hotseat question selection form
2. Click the ‘OK’ button
3. Select ‘History’ from the combobox labelled ‘Subject’
4. Click the ‘OK’ button
5. Select ‘Year 12’ from the combobox labelled ‘Difficulty’
6. Click the ‘OK’ button
7. Select ‘Russia 1881-1924’ from the combobox labelled ‘Topic’
8. Click the ‘OK’ button

Expected Result

a. The error message will be displayed
b. The error message will be displayed
c. The Rock, Paper, Scissors hotseat game form is displayed



Actual Result

PASS/FAIL

5.7

Scope

Test that the back button on the Rock, Paper, Scissors hotseat game form works correctly

Method

1. Load up the Rock, Paper, Scissors hotseat game form
2. Click the ‘back’ button

Expected Result

The Rock, Paper, Scissors main menu form loads

Actual Result

PASS/FAIL

5.8

Scope

Test that the view player one profile button on the Rock, Paper, Scissors hotseat game form works correctly

Method

1. Load up the Rock, Paper, Scissors hotseat game form
2. Click the ‘view profile’ button under player one’s name

Expected Result

The view student profile form loads

Actual Result

PASS/FAIL

5.9

Scope

Test that the details the view player one profile form is populated with are correct

Method

1. Load up the Rock, Paper, Scissors hotseat game form
2. Click the ‘view profile’ button under player one’s name
3. Observe the view profile form

Expected Result

The view player one profile form displays player one’s details correctly

Actual Result

PASS/FAIL

5.10

Scope

Test that the view player two profile button on the Rock, Paper, Scissors hotseat game form works correctly

Method

1. Load up the Rock, Paper, Scissors hotseat game form

2. Click the ‘view profile’ button under player two’s name

Expected Result

The view student profile form loads

Actual Result

PASS/FAIL

5.11

Scope

Test that the details the view player two profile form is populated with are correct

Method

1. Load up the Rock, Paper, Scissors hotseat game form
2. Click the ‘view profile’ button under player two’s name
3. Observe the view profile form

Expected Result

The view player two profile form displays player two’s details correctly

Actual Result

PASS/FAIL

5.12

Scope

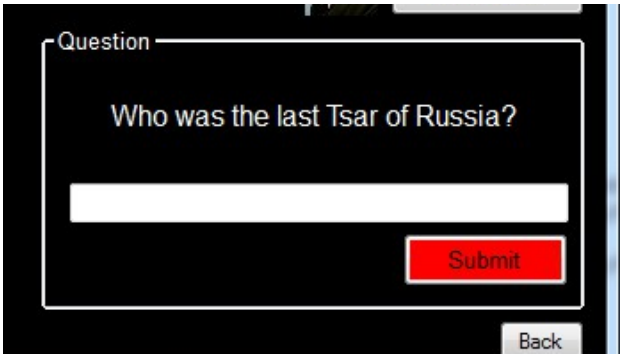
Test that the question section of the Rock, Paper, Scissors hotseat game form populates correctly

Method

1. Load up the Rock, Paper, Scissors hotseat game form
2. Observe the question section in the bottom right

Expected Result

The question section populates correctly, with a question relevant to the selected subject



Actual Result

PASS/FAIL

5.13

Scope

a. Test that, if no answer is given, clicking the submit button works correctly
b. Test that, if the answer given is incorrect, clicking the submit button works correctly
c. Test that, if the answer given is correct, clicking the submit button works correctly

Method

1. Load up the Rock, Paper, Scissors hotseat game form
2. Click the ‘Submit’ button in the question section
3. Type “asdf” as an answer
4. Click the ‘Submit button in the question section
5. Type the correct answer to the given question
6. Click the ‘Submit’ button in the question section

Expected Result

a. The answer is flagged as incorrect
b. The answer is flagged as incorrect
c. The answer is flagged as correct

Actual Result

PASS/FAIL

5.14

Scope

a. Test that the database is correctly updated when a question is answered
b. Test that the game breakdown text files for both players are correctly updated after a game
c. Test that the database is correctly updated after a game

Method

1. Load up the Rock, Paper, Scissors hotseat game form
2. Answer a question correctly
3. Observe 'tblAttempted' in the database
4. Win a game for 95mh11
5. Observe the 96mh11.txt and 95lc11.txt text files at G:\Computing Group\Y10\2009-10\Ben's\Edutainment Suite\Connect4\bin\Debug\WinsLosses
6. Observe the records for 'Matthew Hussey' and 'Laurie Copley' in 'tblStudents' in the database

Expected Result

a. A record is added to tblAttempted when a question is answered
b. The text files for both players are correctly updated
c. The records for both players in tblStudents are changed accordingly

Actual Result

PASS/FAIL

5.15

Scope

Test that the player can, after answering a question correctly, choose a weapon

Method

1. Load up the Rock, Paper, Scissors hotseat game form
2. Answer the question correctly
3. Click the 'Rock' button

Expected Result

The weapon selection controls turn invisible and a new question is asked

Actual Result

PASS/FAIL

5.16

Scope

a. Test that the current player label moves according to which player is currently up
b. Test that, after one player drops a counter, the player is changed

Method

1. Load up the Rock, Paper, Scissors hotseat game form
4. Answer the question correctly_
5. Click the ‘Rock’ button
2. Observe the ‘current player’ label in the top-right

Expected Result

a. The label moves correctly
b. The player is changed

Actual Result

PASS/FAIL

5.17

Scope

Test that, when a weapon has been previously chosen, it can still be chosen by the other player

Method

1. Load up the Rock, Paper, Scissors hotseat game form
2. Answer the question correctly_
3. Click the ‘Rock’ button
4. Answer the question correctly_
5. Click the ‘Rock’ button

Expected Result

The button is selectable both times

Actual Result

PASS/FAIL

5.18

Scope

Test that all weapon selection buttons work correctly

Method

1. Load up the Rock, Paper, Scissors hotseat game form
2. Answer the question correctly
3. Click the ‘Rock’ button
4. Close the program
5. Repeat 1-5, but on each button

Expected Result

Each button either lets the next player answer a question or leads to the fight

Actual Result

PASS/FAIL

5.19

Scope

Test that, when both players have selected their weapons, the next form is displayed

Method

1. Load up the Rock, Paper, Scissors hotseat game form
2. Answer the question correctly
3. Click the ‘Rock’ button
4. Answer the question correctly
5. Click the ‘Paper’ button

Expected Result

The Rock, Paper, Scissors hotseat fight form is displayed

Matt Hussey **Laurie Copley**

3

Actual Result

PASS/FAIL

5.20

Scope

Test that another game can be played after a previous one has ended

Method

1. Load up the Rock, Paper, Scissors hotseat game form
2. Play a game
3. Play another game

Expected Result

Everything is reset correctly after a game and a new one can be played

Actual Result

PASS/FAIL

5.21

Scope

Test that all weapons select correctly

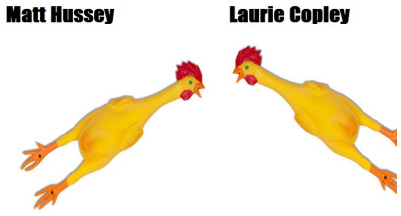
Method

1. Load up the Rock, Paper, Scissors hotseat game form
2. Answer the question correctly_
3. Click the ‘Rock’ button

4. Answer the question correctly
5. Click the ‘Rock’ button
6. Observe the Rock, Paper, Scissors hotseat fight form
7. Go back to the Rock, Paper, Scissors hotseat game form
8. Answer the question correctly_
9. Click the ‘Paper’ button
10. Answer the question correctly
11. Click the ‘Paper’ button
12. Observe the Rock, Paper, Scissors hotseat fight form_
13. Go back to the Rock, Paper, Scissors hotseat game form
14. Answer the question correctly_
15. Click the ‘Scissors’ button
16. Answer the question correctly
17. Click the ‘Scissors’ button
18. Observe the Rock, Paper, Scissors hotseat fight form_
19. Go back to the Rock, Paper, Scissors hotseat game form
20. Answer the question incorrectly
21. Answer the question incorrectly
22. Observe the Rock, Paper, Scissors hotseat fight form

Expected Result

The correct weapon for both players is displayed, and in the case of the scissors of incorrect-answer-rubber-chicken, are facing the correct direction



Actual Result

PASS/FAIL

5.22

Scope

Test that the fight music plays correctly

Method

1. Load up the Rock, Paper, Scissors hotseat fight form
2. Listen

Expected Result

3 Inches of Blood – Deadly Sinners plays correctly

Actual Result

PASS/FAIL

5.23

Scope

Test that all weapons win, lose and draw correctly

Method

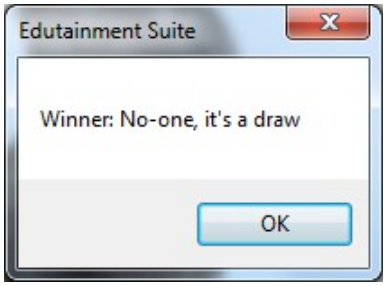
1. Load up the Rock, Paper, Scissors hotseat game form
2. Answer the question correctly_
3. Click the ‘Rock’ button
4. Answer the question correctly
5. Click the ‘Rock’ button
6. Observe the Rock, Paper, Scissors hotseat fight form
7. Go back to the Rock, Paper, Scissors hotseat game form

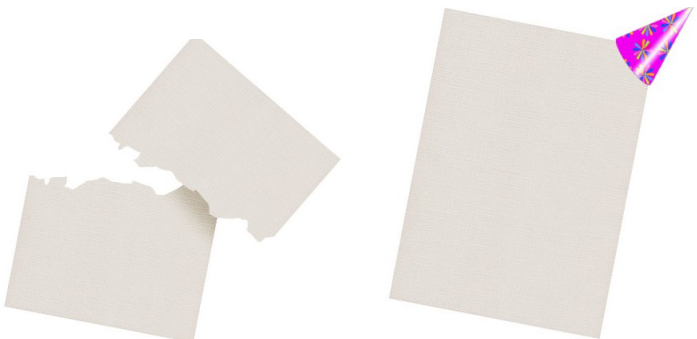
8. Answer the question correctly_
9. Click the ‘Paper’ button
10. Answer the question correctly
11. Click the ‘Paper’ button
12. Observe the Rock, Paper, Scissors hotseat fight form_
13. Go back to the Rock, Paper, Scissors hotseat game form
14. Answer the question correctly_
15. Click the ‘Scissors’ button
16. Answer the question correctly
17. Click the ‘Scissors’ button
18. Observe the Rock, Paper, Scissors hotseat fight form_
19. Go back to the Rock, Paper, Scissors hotseat game form
20. Answer the question incorrectly
21. Answer the question incorrectly
22. Observe the Rock, Paper, Scissors hotseat fight form
23. Go back to the Rock, Paper, Scissors hotseat game form
24. Answer the question correctly_
25. Click the ‘Rock’ button
26. Answer the question correctly
27. Click the ‘Paper’ button
28. Observe the Rock, Paper, Scissors hotseat fight form_
29. Go back to the Rock, Paper, Scissors hotseat game form
30. Answer the question correctly_
31. Click the ‘Rock’ button
32. Answer the question correctly
33. Click the ‘Scissors’ button
34. Observe the Rock, Paper, Scissors hotseat fight form_
35. Go back to the Rock, Paper, Scissors hotseat game form
36. Answer the question correctly_
37. Click the ‘Rock’ button
38. Answer the question incorrectly
39. Observe the Rock, Paper, Scissors hotseat fight form
40. Go back to the Rock, Paper, Scissors hotseat game form
41. Answer the question correctly_
42. Click the ‘Paper’ button
43. Answer the question correctly
44. Click the ‘Rock’ button
45. Observe the Rock, Paper, Scissors hotseat fight form_
46. Go back to the Rock, Paper, Scissors hotseat game form
47. Answer the question correctly_
48. Click the ‘Paper’ button
49. Answer the question correctly
50. Click the ‘Scissors’ button
51. Observe the Rock, Paper, Scissors hotseat fight form_
52. Go back to the Rock, Paper, Scissors hotseat game form
53. Answer the question correctly_
54. Click the ‘Paper’ button
55. Answer the question incorrectly
56. Observe the Rock, Paper, Scissors hotseat fight form

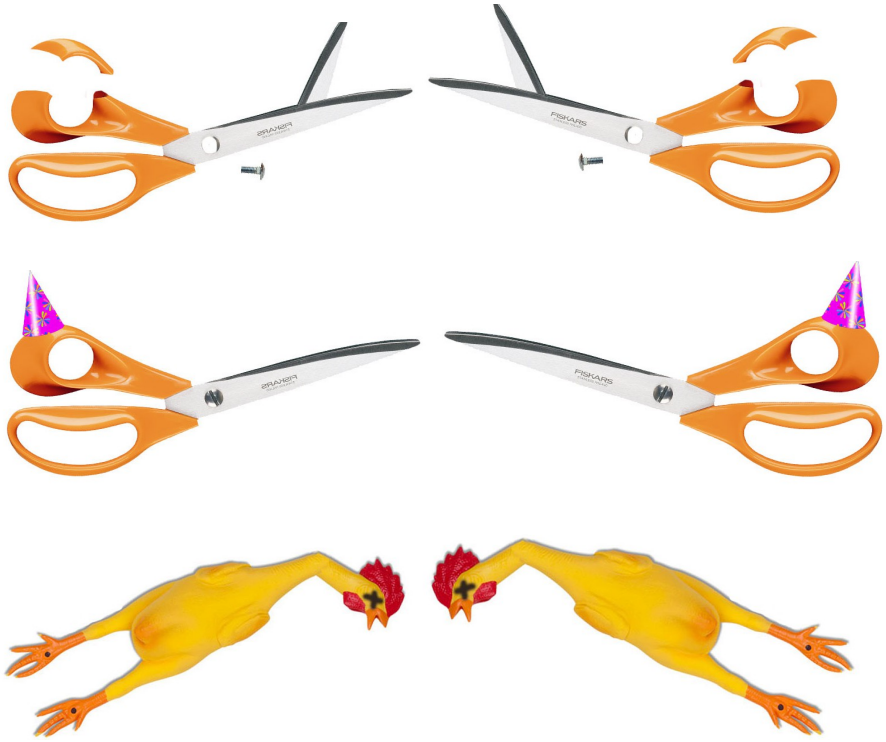
57. Go back to the Rock, Paper, Scissors hotseat game form
58. Answer the question correctly_
59. Click the ‘Scissors’ button
60. Answer the question correctly
61. Click the ‘Rock’ button
62. Observe the Rock, Paper, Scissors hotseat fight form_
63. Go back to the Rock, Paper, Scissors hotseat game form
64. Answer the question correctly_
65. Click the ‘Scissors’ button
66. Answer the question correctly
67. Click the ‘Paper’ button
68. Observe the Rock, Paper, Scissors hotseat fight form_
69. Go back to the Rock, Paper, Scissors hotseat game form
70. Answer the question correctly_
71. Click the ‘Scissors’ button
72. Answer the question incorrectly
73. Observe the Rock, Paper, Scissors hotseat fight form

Expected Result

Rock beats scissors, paper beats rock, scissors beat paper, anything beats rubber chicken. If both are the same, they will draw. Graphics update accordingly and a message box will display.







Actual Result

PASS/FAIL

Section 6: Teacher Forms

Sub-Section 1: View Student

6.1.1

Scope

Test that the account details section of the view student form populates correctly

Method

1. Enter “stst11” as a username and “password” as a password
2. Observe the account details section

Expected Result

The correct name and picture are displayed on the form

Actual Result

PASS/FAIL

6.1.2

Scope

Test that the view staff profile button on the view student form works correctly

Method

1. Load up the view student form
2. Click the ‘view profile’ button in the top right

Expected Result

The view staff profile form loads

Actual Result

PASS/FAIL

6.1.3

Scope

Test that the details the view staff profile form is populated with are correct

Method

1. Load up the view student form
2. Click the ‘view profile’ button in the top right
3. Observe the view profile form

Expected Result

The view staff profile form displays the logged-in member of staff’s details correctly

Actual Result

PASS/FAIL

6.1.4

Scope

Test that the student listbox populates on form load

Method

1. Load up the view student form
2. Observe the listbox

Expected Result

The listbox populates

Actual Result

PASS/FAIL

6.1.5

Scope

Test that the search student button on the view student form works correctly

a. Test that the view student form, upon no data being input, will reject the input
b. Test that the view student form, upon partial data entry, will accept the input and search for the chosen student
c. Test that the view student form, upon the entry of a non-existent student, will reject the input
d. Test that the view student form, upon valid data entry, will accept the input and search for the chosen student

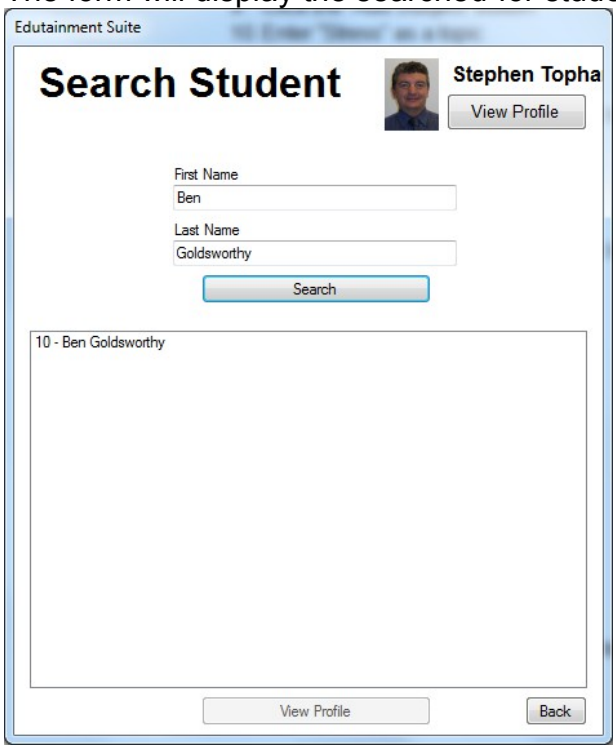
Method

1. Load the view student form
2. Click the ‘Search’ button
3. Enter “asdf” as a first name
4. Click the ‘Search’ button
5. Enter “Ben” as a first name
6. Click the ‘Search’ button
7. Enter “Goldsworthy” as a last name
8. Click the ‘Search’ button
9. Observe the listbox

Expected Result

a. The form will reject the data

b. The form will display students matching the search criteria
c. The form will reject the data
d. The form will display the searched-for student



Actual Result

PASS/FAIL

6.1.6

Scope

Test that view profile button on the view student form is enabled when a record is selected

Method

1. Load up the view student form
2. Enter “Ben” as a first name and “Goldsworthy” as a last name
3. Click the ‘Search’ button
4. Select the record

5. Check that the ‘View Profile’ button is enabled

Expected Result

The combobox will be enabled

Actual Result

PASS/FAIL

6.1.7

Scope

Test that the view student profile button on the view student form works correctly

Method

1. Load up the view student form
2. Enter “Ben” as a first name and “Goldsworthy” as a last name
3. Click the ‘Search’ button
4. Select the record
5. Click the ‘View Profile’ button in the bottom centre

Expected Result

The view student profile form loads

Actual Result

PASS/FAIL

6.1.8

Scope

Test that the details the view student profile form is populated with are correct

Method

1. Load up the view student form
2. Enter “Ben” as a first name and “Goldsworthy” as a last name
3. Click the ‘Search’ button
4. Select the record
5. Click the ‘View Profile’ button in the bottom centre

6. Observe the view profile form

Expected Result

The view student profile form displays the searched-for student's details correctly

Actual Result

PASS/FAIL

6.1.9

Scope

Test that the back button on the view student form works correctly

Method

1. Load up the view student form
2. Click the 'Back' button

Expected Result

The staff home form loads

Actual Result

PASS/FAIL

Sub-Section 2: Create Question

6.2.1

Scope

Test that the subject combobox on the create question form populates correctly

Method

1. Load up the create question form
2. Observe the combobox labelled 'Subject'

Expected Result

The combobox will be populated with the subjects from the database correctly

Actual Result

PASS/FAIL

6.2.2

Scope

a. Test that the topic combobox on the create question form is disabled as long as there is no subject selected
b. Test that the topic combobox on the create question form enables when a subject is selected

Method

1. Load up the create question form
2. Check that the combobox labelled ‘Topic’ is disabled
3. Select ‘History’ from the combobox labelled ‘Subject’
4. Check that the combobox labelled ‘Topic’ is enabled

Expected Result

a. The combobox will be disabled
b. The combobox will be enabled

Actual Result

PASS/FAIL

6.2.3

Scope

Test that the submit button on the create question form works correctly

a. Test that the create question form, upon no data being input, will reject the input
b. Test that the create question form, upon partial data entry, will reject the input
c. Test that the create question form, upon valid data entry but not confirming the input, will reject the input

d. Test that the create question form, upon valid data entry and confirming the input, will accept the input and add the question to the database

Method

1. Load up the create question form
2. Click the ‘Submit’ button
3. Select “Psychology” as a subject
4. Click the ‘Submit’ button
5. Select “Year 12” as a year
6. Click the ‘Submit’ button
7. Select “Stress (Year 12)” as a topic
8. Click the ‘Submit’ button
9. Enter “What does WMM stand for?” as a question
10. Click the ‘Submit’ button
11. Enter “Working Memory Model” as an answer
12. Click the ‘Submit’ button
13. Click the ‘No’ button
14. Observe ‘tblQuestions’ in the database
15. Click the ‘Submit’ button
16. Click the ‘Yes’ button
17. Observe ‘tblQuestions’ in the database

Expected Result

a. The form will reject the data
b. The form will reject the data
c. The form will reject the data
d. The form will add the inputted question into the database

Actual Result

PASS/FAIL

6.2.4

Scope

Test that the back button on the create question form works correctly

Method

1. Load up the create question form
2. Click the ‘Back’ button

Expected Result

The staff home form loads

Actual Result

PASS/FAIL

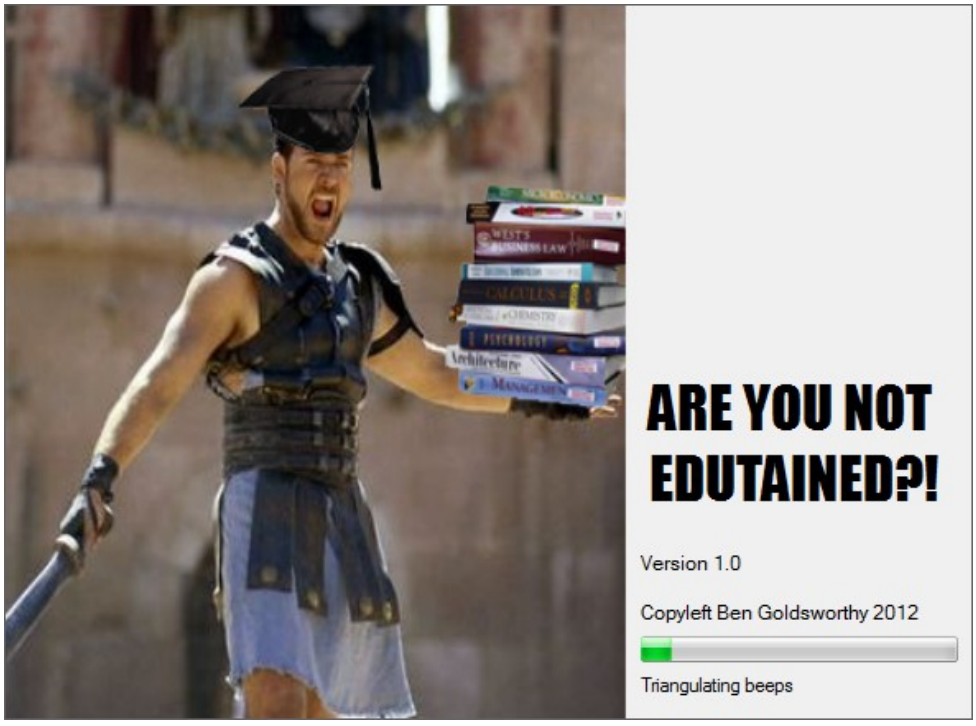
Actual Test Runs

Section 1: Loading & Setup

1.1

Actual Result

As expected

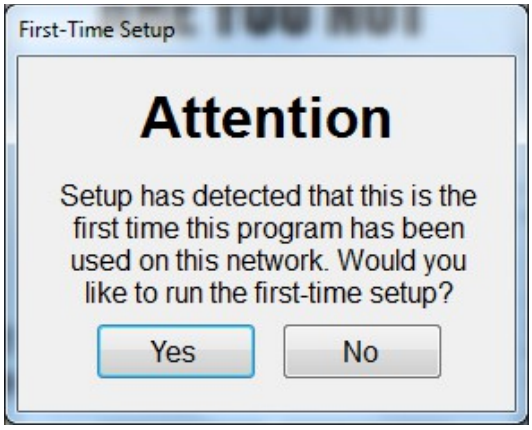


PASS

1.2

Actual Result

As expected



PASS

1.3

Actual Result

As expected

PASS

1.4

Actual Result

As expected

PASS

1.5

Actual Result

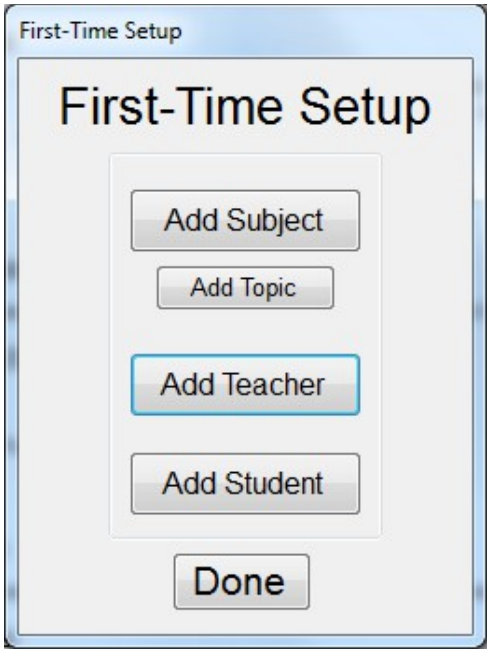
As expected

PASS

1.6

Actual Result

As expected

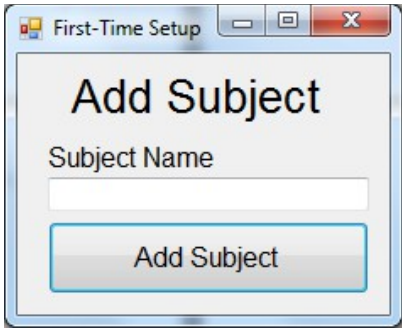


PASS

1.7

Actual Result

As expected



PASS

1.8

Actual Result

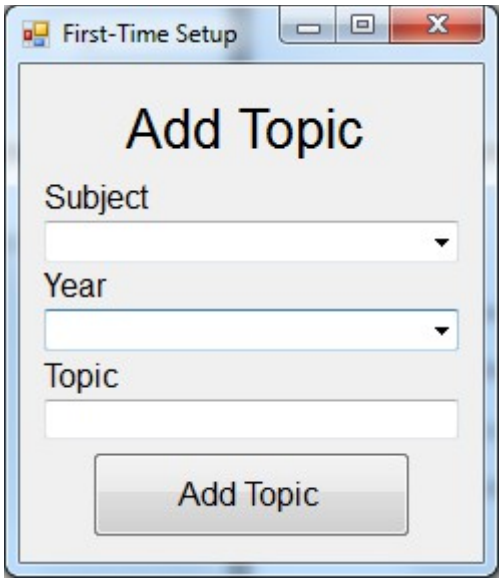
As expected

PASS

1.9

Actual Result

As expected



PASS

1.10

Actual Result

As expected

PASS

1.11

Actual Result

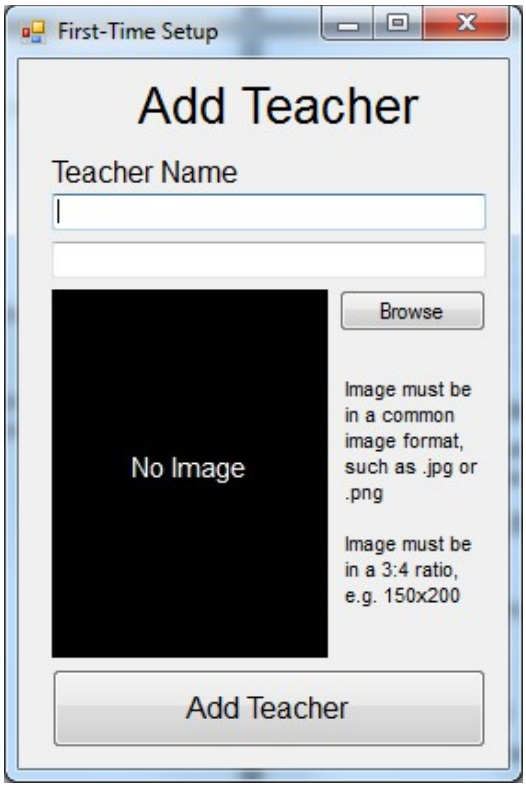
As expected

PASS

1.12

Actual Result

As expected



PASS

1.13

Actual Result

As expected

PASS

1.14

Actual Result

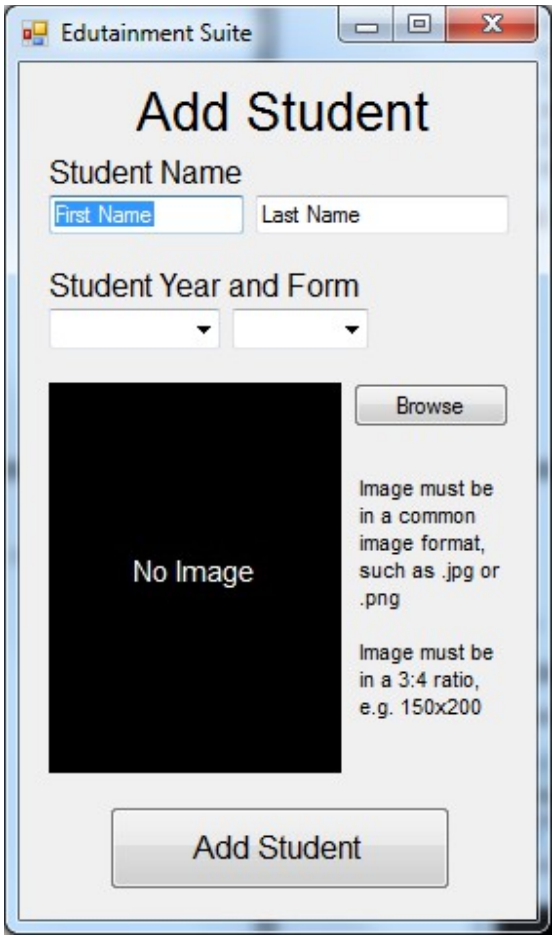
As expected

PASS

1.15

Actual Result

As expected



PASS

1.16

Actual Result

As expected

PASS

1.17

Actual Result

As expected

PASS

1.18

Actual Result

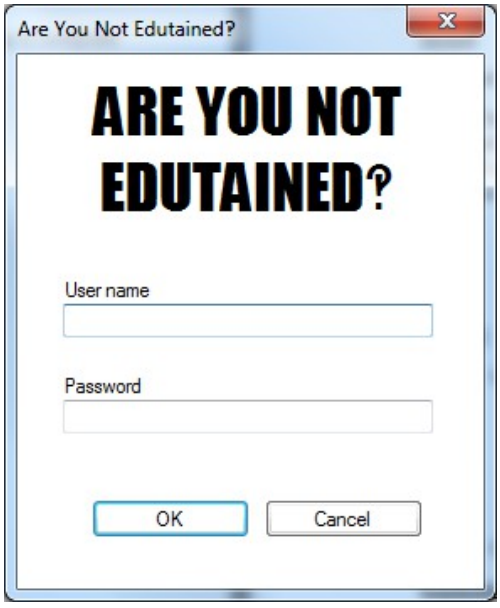
As expected

PASS

1.19

Actual Result

As expected



PASS

1.20

Actual Result

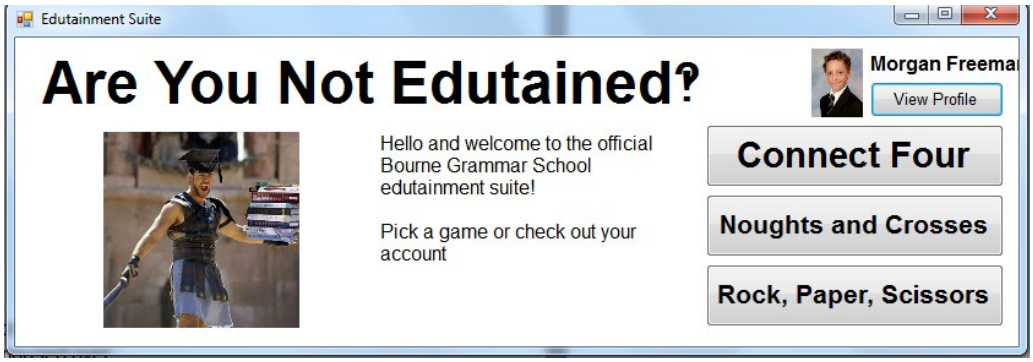
As expected

PASS

1.21

Actual Result

As expected



PASS

Section 2: Main Menu & Account View

Sub-Section 1: Student Experience

2.1.1

Actual Result

As Expected

PASS

2.1.2

Actual Result

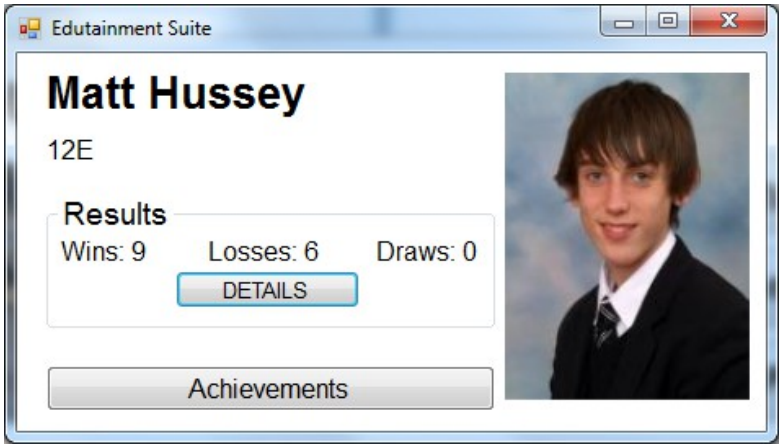
As Expected

PASS

2.1.3

Actual Result

As Expected



PASS

2.1.4

Actual Result

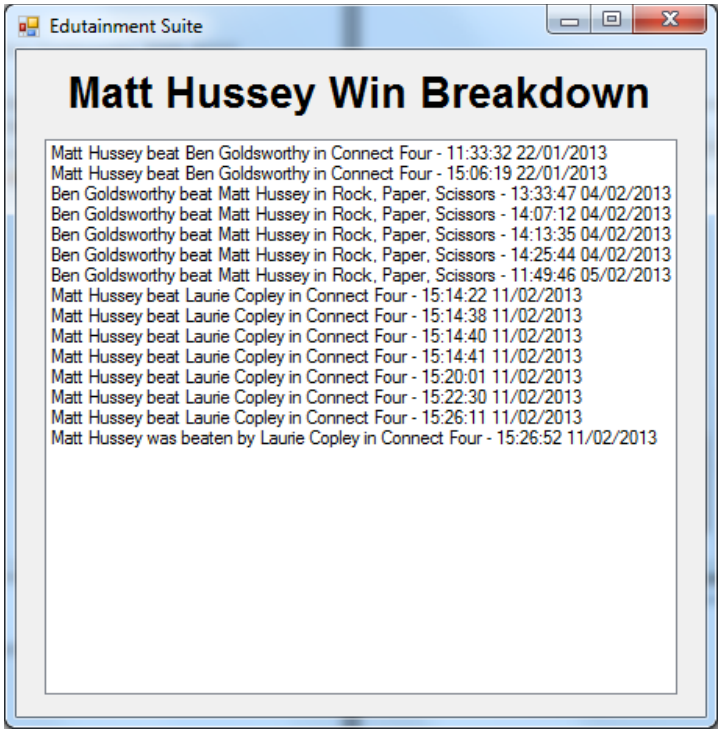
As Expected

PASS

2.1.5

Actual Result

As Expected



PASS

2.1.6

Actual Result

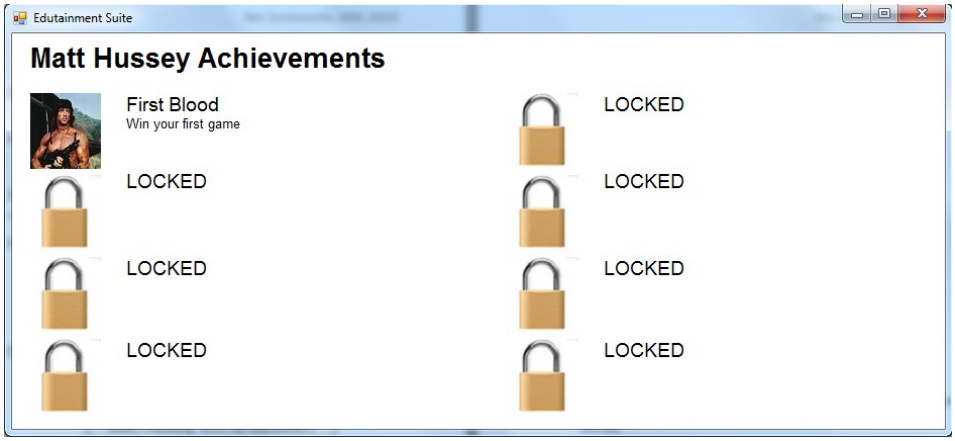
As Expected

PASS

2.1.7

Actual Result

As Expected



PASS

2.1.8

Actual Result

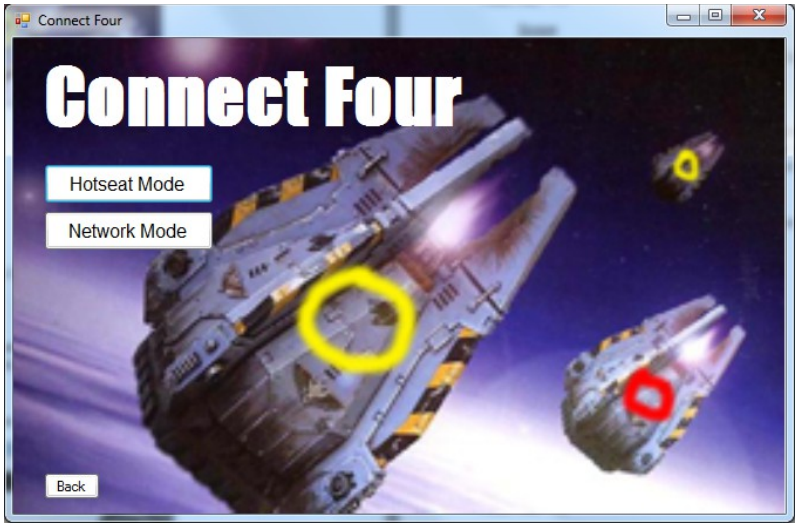
As Expected

PASS

2.1.9

Actual Result

As Expected



PASS

2.1.10

Actual Result

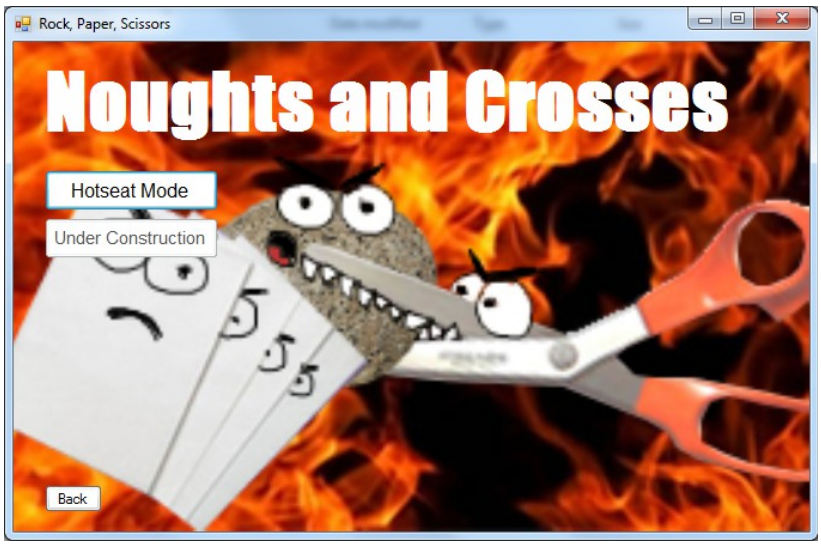
As Expected

PASS

2.1.11

Actual Result

As Expected



PASS

Sub-Section 2: Staff Experience

2.2.1

Actual Result

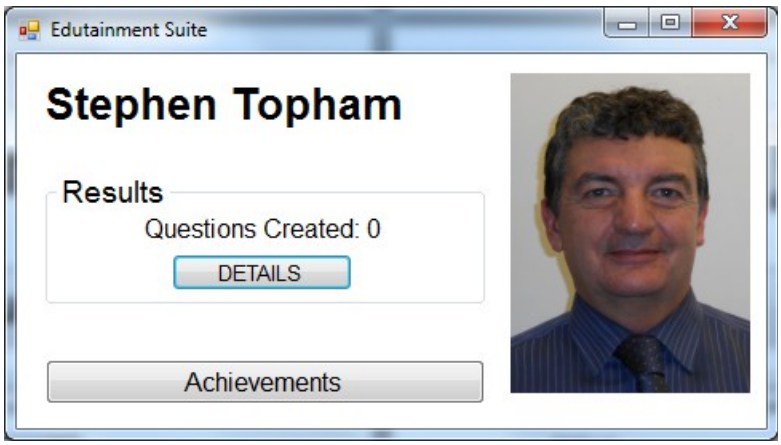
As Expected

PASS

2.2.2

Actual Result

As Expected



PASS

2.2.3

Actual Result

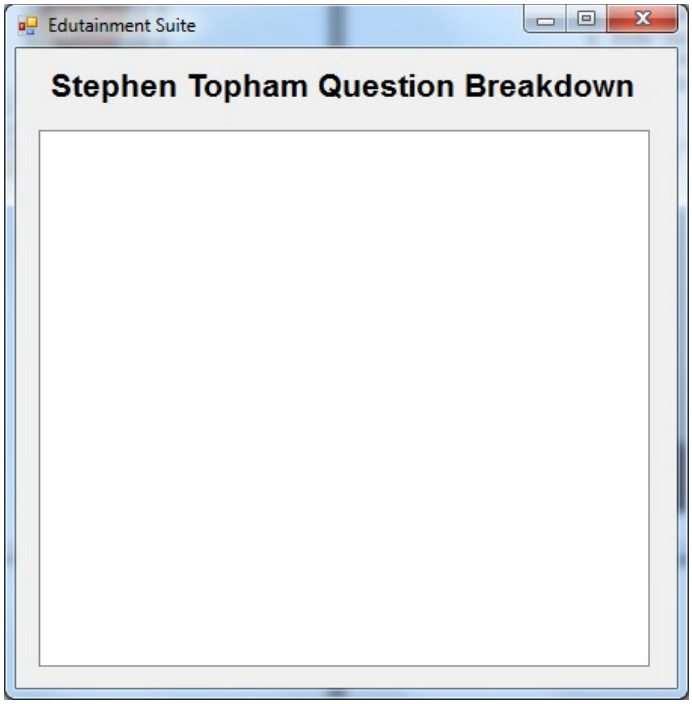
As Expected

PASS

2.2.4

Actual Result

As Expected



PASS

2.2.5

Actual Result

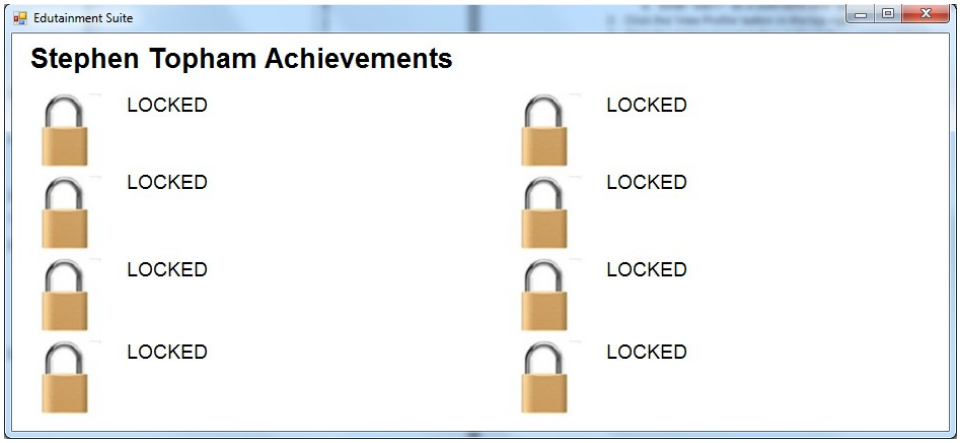
As Expected

PASS

2.2.6

Actual Result

As Expected



PASS

2.2.7

Actual Result

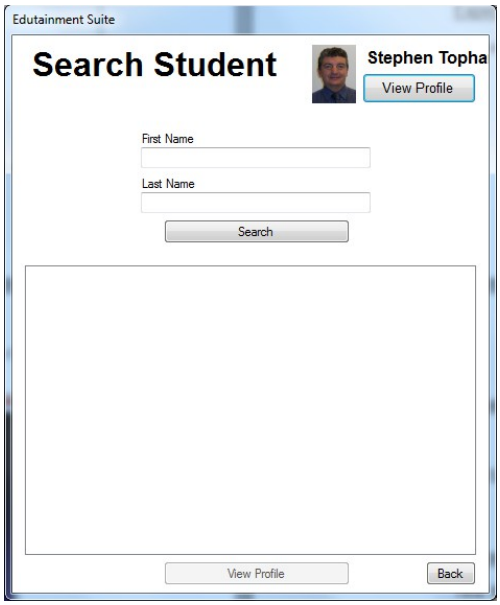
As Expected

PASS

2.2.8

Actual Result

As Expected

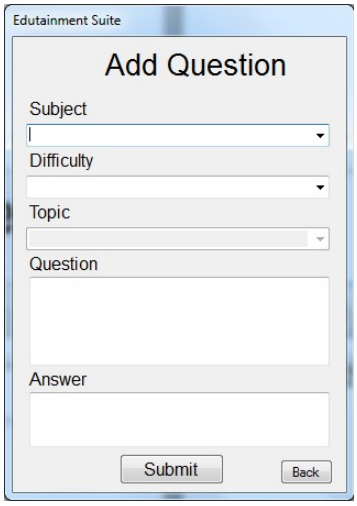


PASS

2.2.9

Actual Result

As Expected

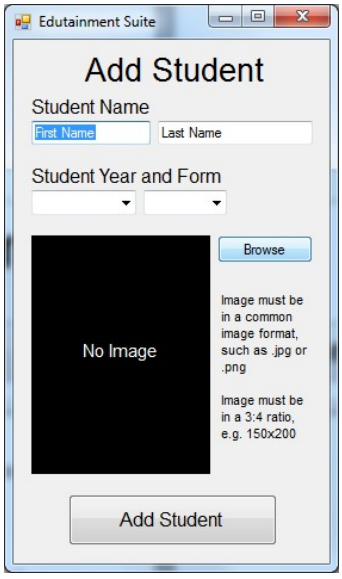


PASS

2.2.10

Actual Result

As Expected



PASS

Section 3: Connect Four

3.1

Actual Result

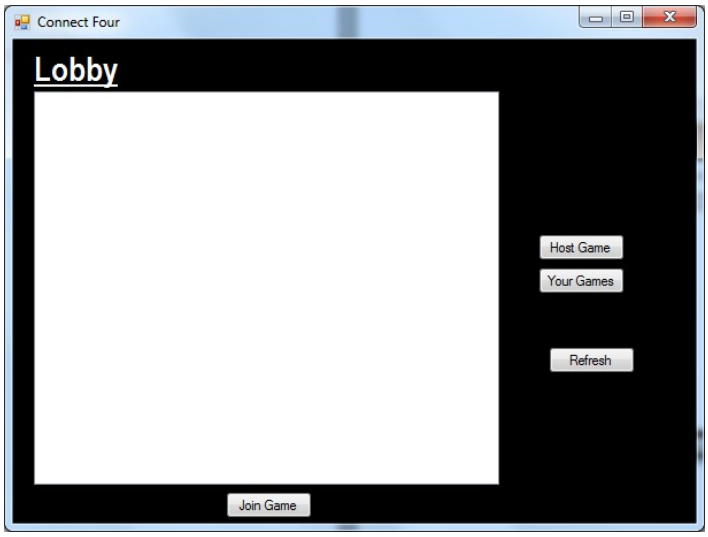
As expected

PASS

3.2

Actual Result

As expected



PASS

Sub-Section 1: Hotseat Game

3.1.1

Actual Result

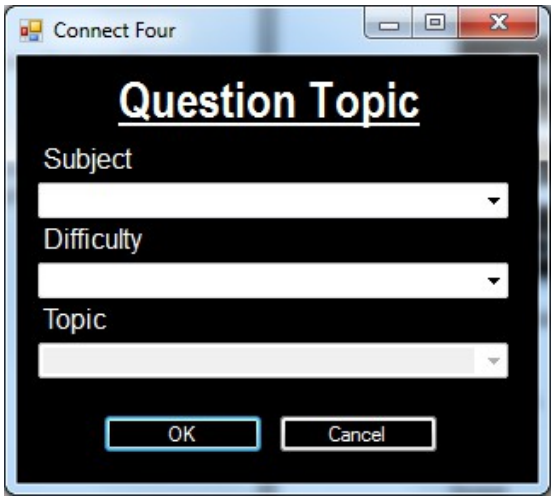
As Expected

PASS

3.1.2

Actual Result

As Expected



PASS

3.1.3

Actual Result

As Expected

PASS

3.1.4

Actual Result

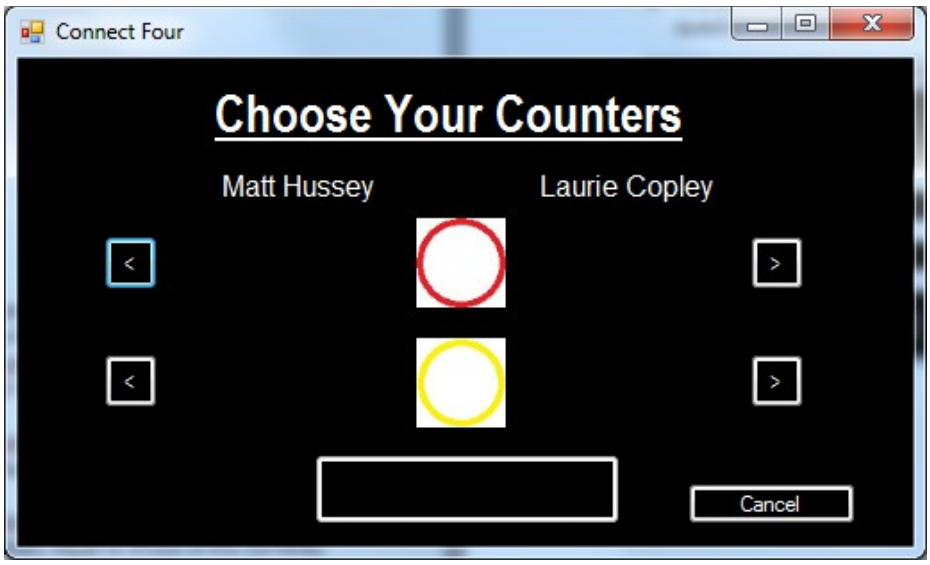
As Expected

PASS

3.1.5

Actual Result

As Expected



PASS

3.1.6

Actual Result

As Expected

PASS

3.1.7

Actual Result

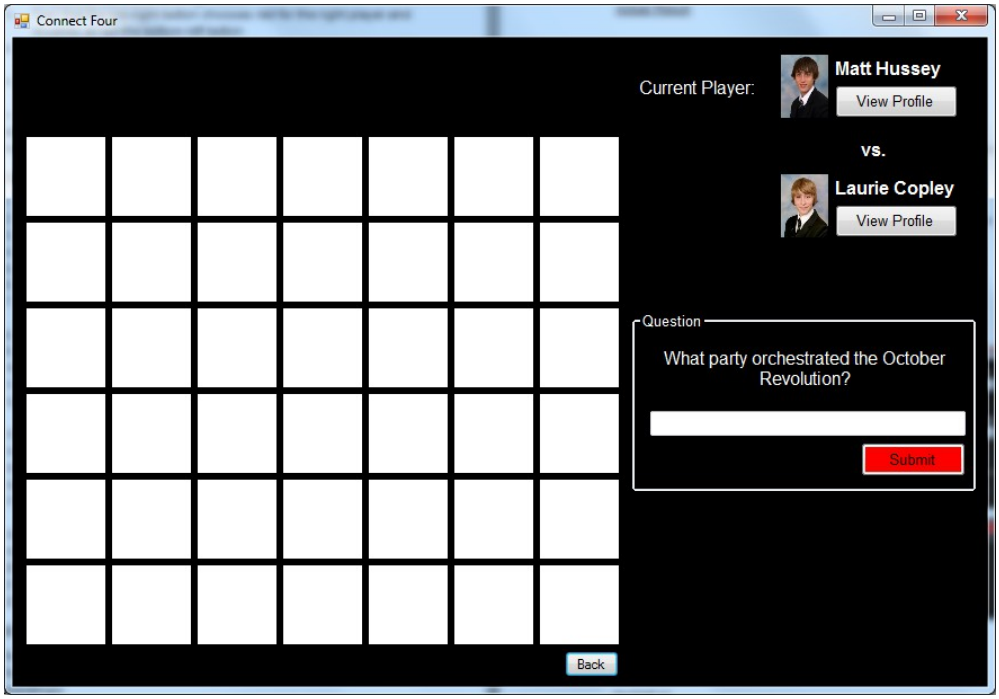
As Expected

PASS

3.1.8

Actual Result

As Expected



PASS

3.1.9

Actual Result

As Expected

PASS

3.1.10

Actual Result

As Expected

PASS

3.1.11

Actual Result

As Expected

PASS

3.1.12

Actual Result

As Expected

PASS

3.1.13

Actual Result

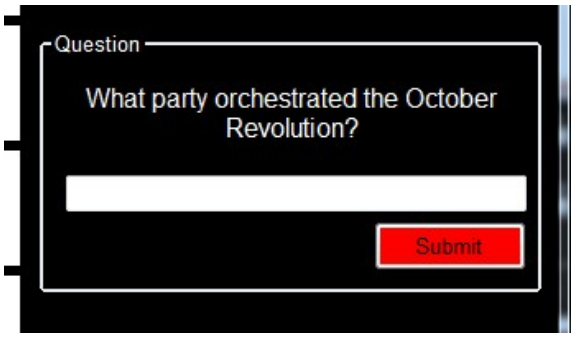
As Expected

PASS

3.1.14

Actual Result

As Expected



PASS

3.1.15

Actual Result

As Expected

PASS

3.1.16

Actual Result

As Expected

PASS

3.1.17

Actual Result

As Expected

PASS

3.1.18

Actual Result

As Expected

PASS

3.1.19

Actual Result

As Expected

PASS

3.1.20

Actual Result

As Expected

PASS

3.1.21

Actual Result

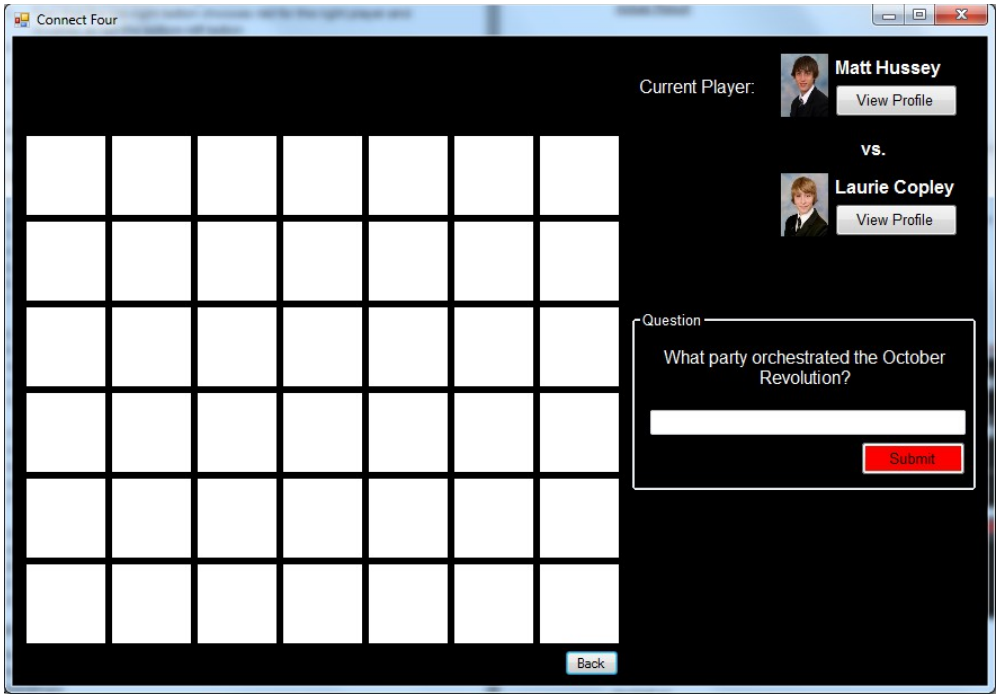
As Expected

PASS

3.1.22

Actual Result

As Expected



PASS

3.1.23

Actual Result

As Expected

PASS

3.1.24

Actual Result

As Expected

PASS

Sub-Section 2: Network Game

3.2.1

Actual Result

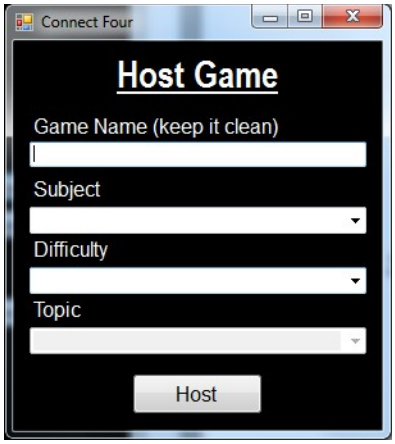
As Expected

PASS

3.2.2

Actual Result

As Expected

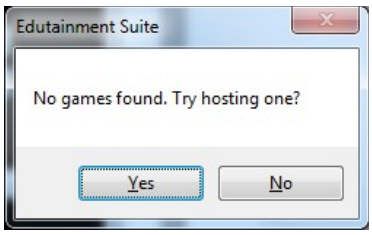


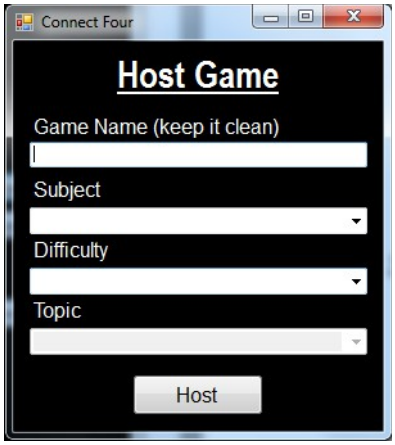
PASS

3.2.3

Actual Result

As Expected





PASS

3.2.4

Actual Result

As Expected

PASS

3.2.5

Actual Result

As Expected

PASS

3.2.6

Actual Result

As Expected

PASS

3.2.7

Actual Result

As Expected

PASS

3.2.8

Actual Result

As Expected

PASS

3.2.9

Actual Result

As Expected

PASS

3.2.20

Actual Result

As Expected

PASS

3.2.21

Actual Result

As Expected

PASS

3.2.22

Actual Result

As Expected

PASS

3.2.23

Actual Result

As Expected

PASS

3.2.24

Actual Result

Nothing happens

FAIL

Changes Made

Timer set to enable correctly

Actual Result

As expected

PASS

3.2.25

Actual Result

As Expected

PASS

3.2.26

Actual Result

As Expected

PASS

3.2.27

Actual Result

As Expected

PASS

3.2.28

Actual Result

As Expected

PASS

3.2.29

Actual Result

As Expected

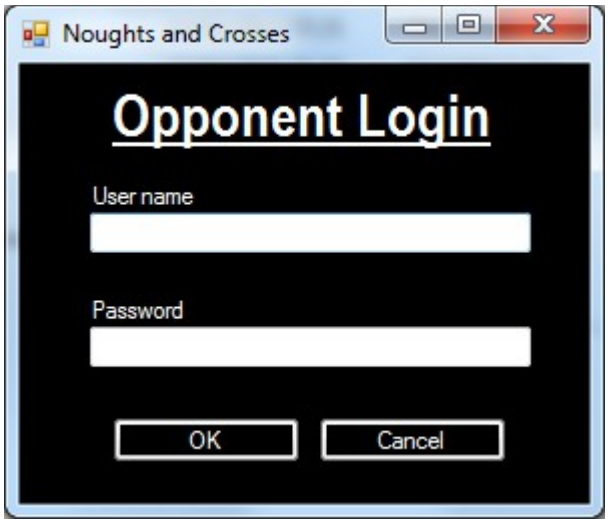
PASS

Section 4: Noughts and Crosses

4.1

Actual Result

As expected



PASS

4.2

Actual Result

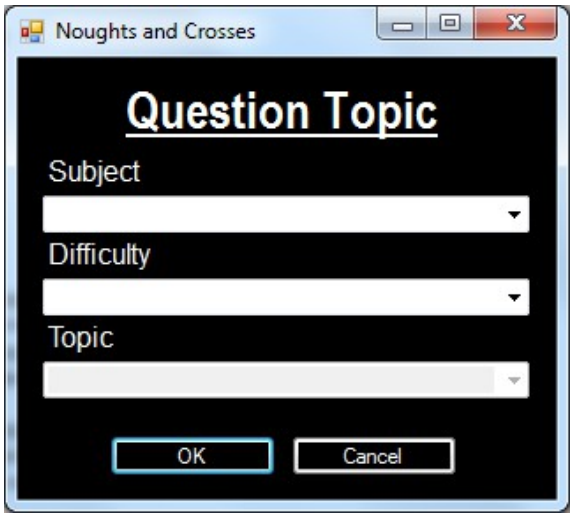
As expected

PASS

4.3

Actual Result

As expected



PASS

4.4

Actual Result

As expected

PASS

4.5

Actual Result

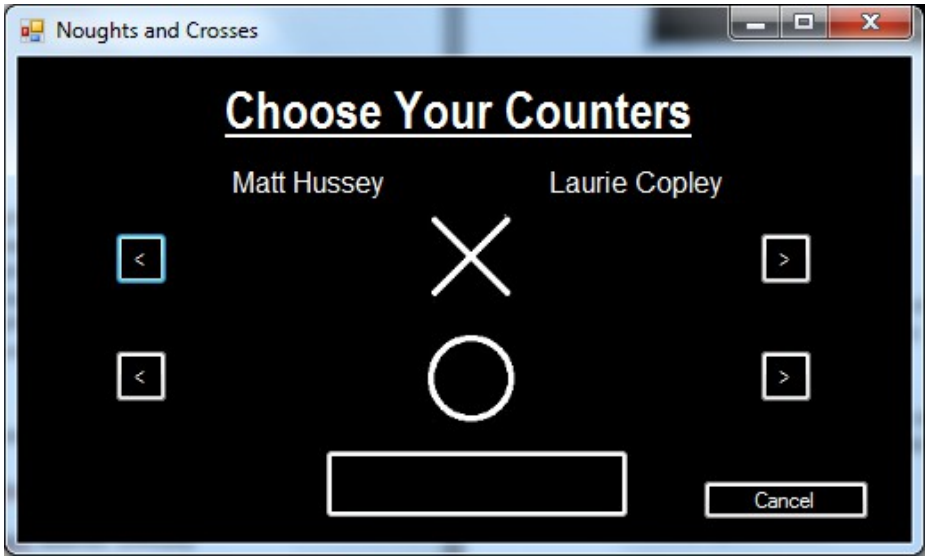
As expected

PASS

4.6

Actual Result

As expected



PASS

4.7

Actual Result

Back button does nothing

FAIL

Changes Made

Code added to button

New Result

As expected

PASS

4.8

Actual Result

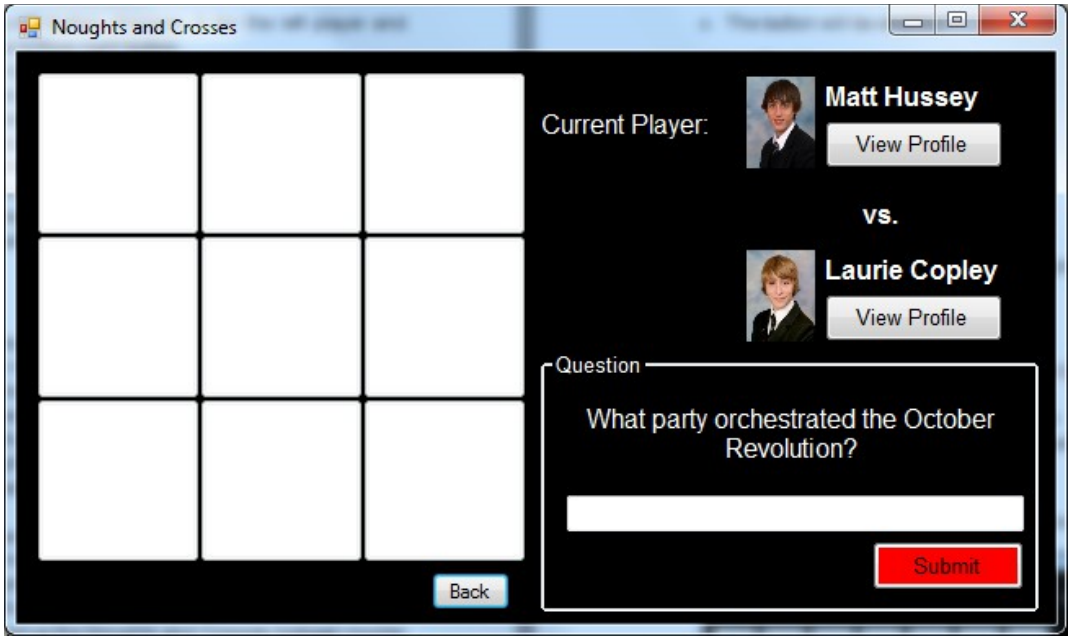
As expected

PASS

4.9

Actual Result

As expected



PASS

4.10

Actual Result

As expected

PASS

4.11

Actual Result

As expected

PASS

4.12

Actual Result

As expected

PASS

4.13

Actual Result

As expected

PASS

4.14

Actual Result

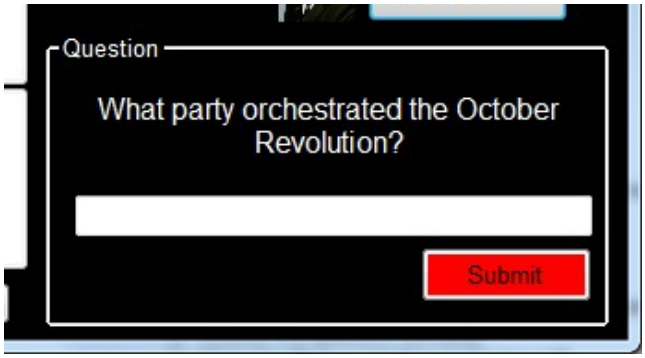
As expected

PASS

4.15

Actual Result

As expected



PASS

4.16

Actual Result

As expected

PASS

4.17

Actual Result

As expected

PASS

4.18

Actual Result

As expected

PASS

4.19

Actual Result

As expected

PASS

4.20

Actual Result

As expected

PASS

4.21

Actual Result

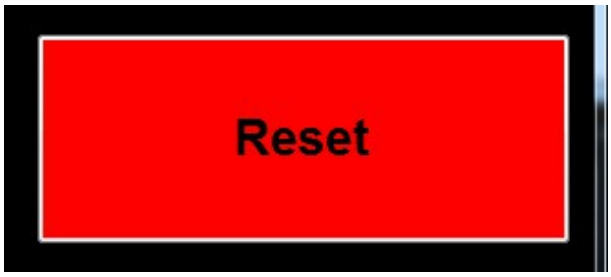
As expected

PASS

4.22

Actual Result

As expected



PASS

4.23

Actual Result

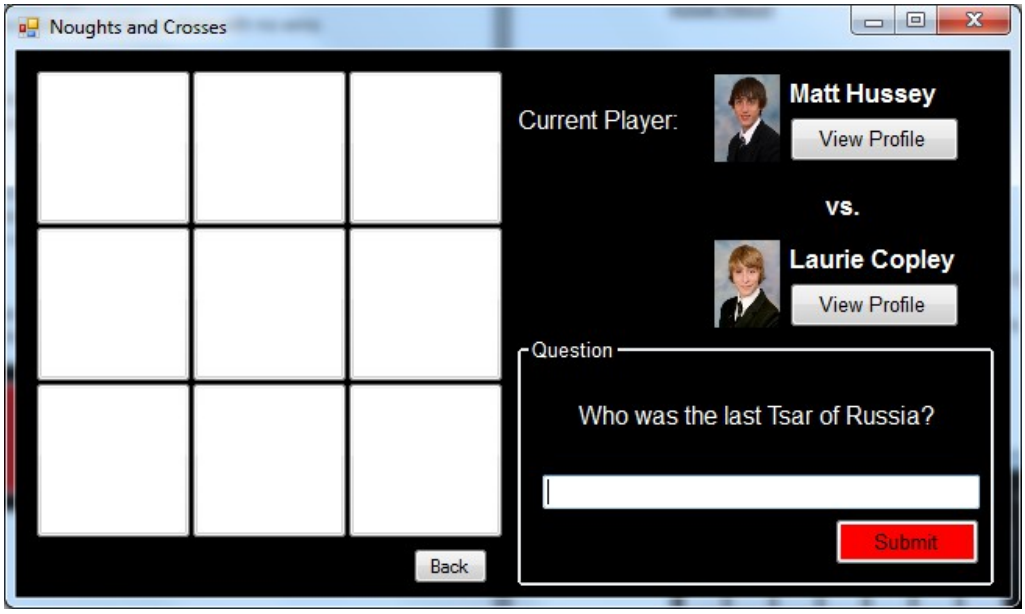
As expected

PASS

4.24

Actual Result

As expected



PASS

4.25

Actual Result

As expected

PASS

4.26

Actual Result

As expected

PASS

Section 4: Rock, Paper, Scissors

5.1

Actual Result

As expected

PASS

5.2

Actual Result

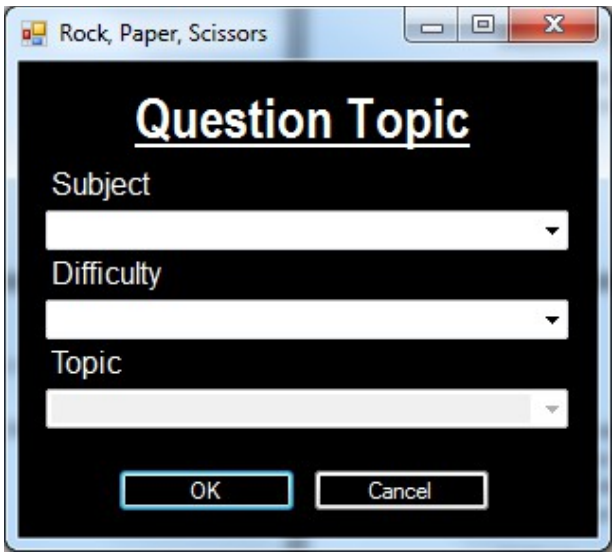
As expected

PASS

5.3

Actual Result

As expected



PASS

5.4

Actual Result

As expected

PASS

5.5

Actual Result

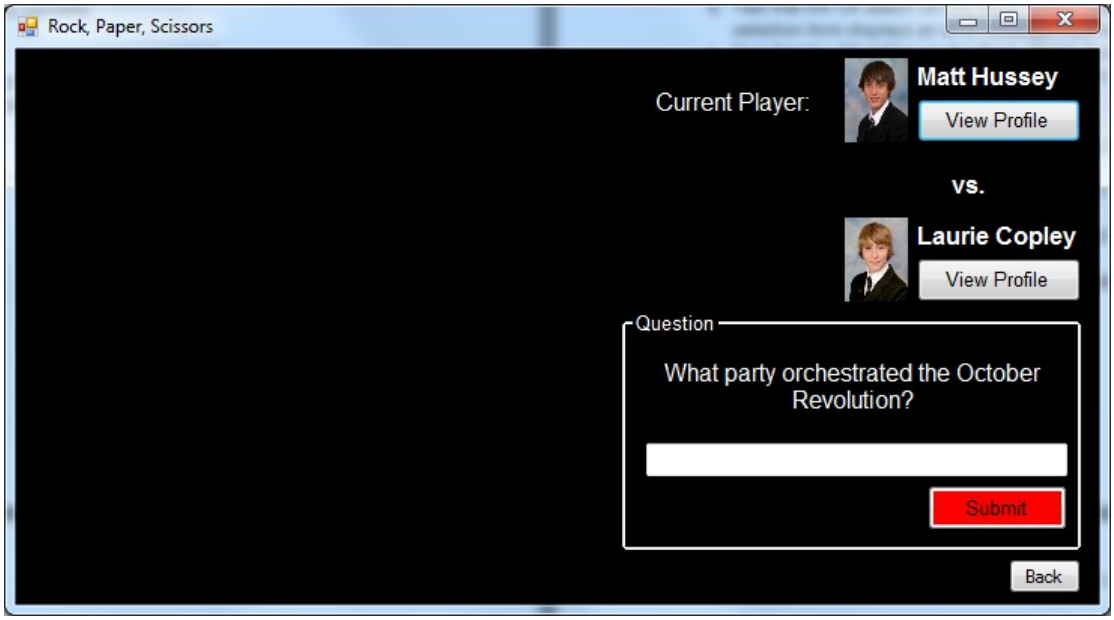
As expected

PASS

5.6

Actual Result

As expected



PASS

5.7

Actual Result

As expected

PASS

5.8

Actual Result

As expected

PASS

5.9

Actual Result

Account section populates with details of wrong player

FAIL

Changes Made

Code amended to populate with correct student

Actual Result

As expected

PASS

5.10

Actual Result

As expected

PASS

5.11

Actual Result

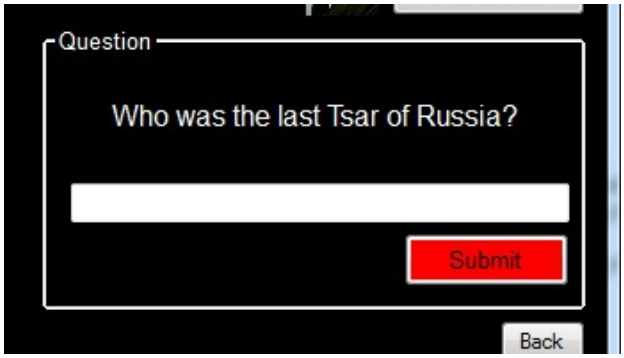
As expected

PASS

5.12

Actual Result

As expected



PASS

5.13

Actual Result

As expected

PASS

5.14

Actual Result

As expected

PASS

5.15

Actual Result

As expected

PASS

5.16

Actual Result

As expected

PASS

5.17

Actual Result

As expected

PASS

5.18

Actual Result

As expected

PASS

5.19

Actual Result

As expected

Matt Hussey **Laurie Copley**

3

PASS

5.20

Actual Result

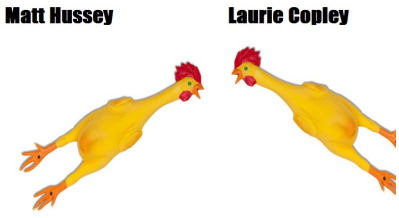
As expected

PASS

5.21

Actual Result

As expected



PASS

5.22

Actual Result

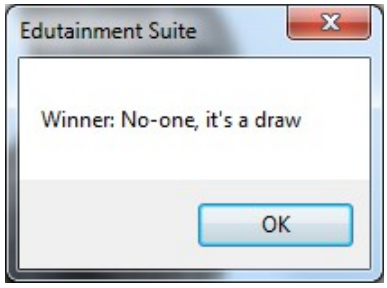
As expected

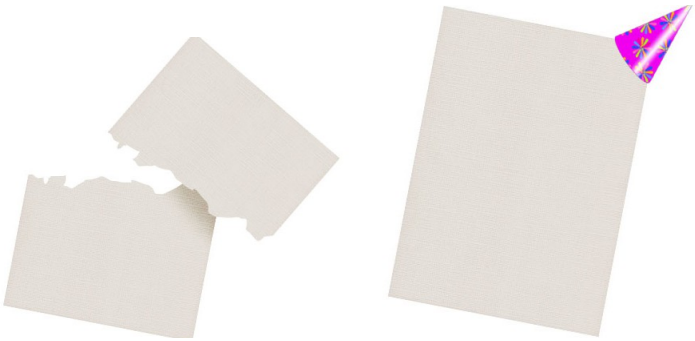
PASS

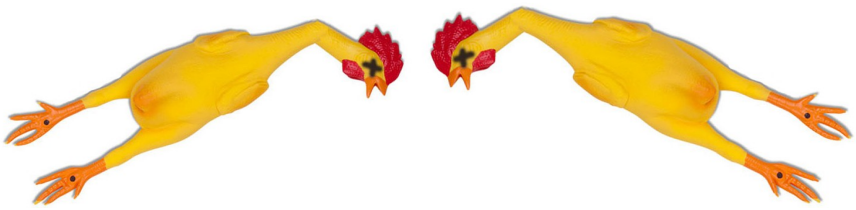
5.23

Actual Result

As expected







PASS

Section 6: Teacher Forms

Sub-Section 1: View Student

6.1.1

Actual Result

As Expected

PASS

6.1.2

Actual Result

As Expected

PASS

6.1.3

Actual Result

As Expected

PASS

6.1.4

Actual Result

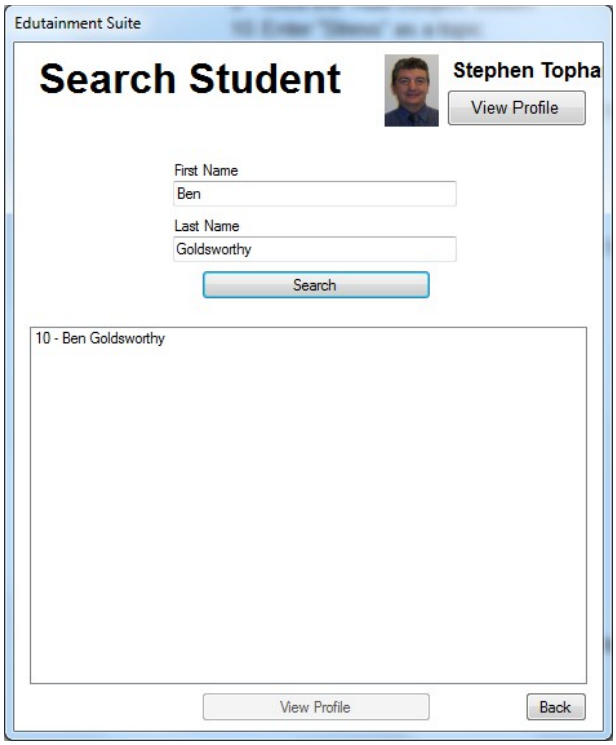
As Expected

PASS

6.1.5

Actual Result

As Expected



PASS

6.1.6

Actual Result

As Expected

PASS

6.1.7

Actual Result

As Expected

PASS

6.1.8

Actual Result

As Expected

PASS

6.1.9

Actual Result

As Expected

PASS

Sub-Section 2: Create Question

6.2.1

Actual Result

As Expected

PASS

6.2.2

Actual Result

As Expected

PASS

6.2.3

Actual Result

As Expected

PASS

6.2.4

Actual Result

As Expected

PASS

Evaluation

Evaluation of Test Results

The test results section went very well, which is remarkable when the comprehensiveness of it is considered. Of 159 tests, only three initially failed. Luckily only small changes were required to fix them, and afterwards all tests completed as expected. All three errors were a result of simply forgetting to add in the relevant code, rather than anything serious.

Test 3.2.24

Code Before

```
Private Sub frmConnect4Network_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        'Runs the OpenDB subroutine
        OpenDB()

        'If the logged-in student is red...
        If LoggedInStudent.C4Player = "Red" Then
```

Code After

```
Private Sub frmConnect4Network_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        'Runs the OpenDB subroutine
        OpenDB()

        'Starts the update timer
        tmrUpdate.Enabled = True

        'If the logged-in student is red...
        If LoggedInStudent.C4Player = "Red" Then
```

Test 4.7

Code Before

```
'Subroutine runs when the cancel button is clicked
Private Sub btnCancel_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnCancel.Click

    End Sub
```

Code After

```
'Subroutine runs when the cancel button is clicked
Private Sub btnCancel_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnCancel.Click
        'Shows the Noughts and Crosses hotseat question selection form
        frmNaCHotseatSubject.Show()
        'Closes this form
        Me.Close()
End Sub
```

Test 5.9

Code Before

```
'Subroutine runs when view logged-in student's profile button is clicked
Private Sub btnViewProfile_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnViewProfile.Click
        'Sets viewed profile to logged-in student's
        Viewing = 2
        'Shows the student account form
        frmStudentAccount.Show()
End Sub
```

Code After

```
'Subroutine runs when view logged-in student's profile button is clicked
Private Sub btnViewProfile_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnViewProfile.Click
        'Sets viewed profile to logged-in student's
        Viewing = 1
        'Shows the student account form
        frmStudentAccount.Show()
End Sub
```

The program is very stable across the board, and in its current state could be reasonable considered a finished product. White box testing, using four users of varying degrees of computer literacy and giving them tasks that would guide them through using every element of functionality within the program, revealed the user experience to intuitive and reliably stable and black box testing designed to collapse the system had no or little effect. White box testers commented on the ease of use and the responsiveness of the network game of Connect Four, amongst other things.

Solution Satisfies the Objectives

In the Analysis section at the beginning of this project I outlined a series of objectives for the final project. In this section I will go through them and compare them to the finished product.

***

“Provide an intuitive UI that is simple and attractive”

Testing the program using unfamiliar users revealed some issues with ambiguity with regards to what to do next, but these issues were ironed out. Beauty is in the eye of the beholder, but personally I believe the appearance of the program is uniformly functional, yet elegant. Testers commented on the simple yet clean visual style of the program, and the lack of ambiguity in how to perform tasks.

“Provide versions of hangman, connect four and noughts & crosses with subject- and topic-specific questions selectable on the main menu”

Whilst Connect Four and Noughts & Crosses are present in the program, I substituted Hangman for Rock, Paper, Scissors in the end as I expected it to be simpler to program and so would take up less time in an already time-consuming project. The question selection, whilst not so much on the main menu, is present and functional.

“Provide a rudimentary method of inputting new questions”

Originally I planned the question creation functionality to be little more than a step up from going into the database itself and adding the record manually. In the end I made in a complete part of the program and it received at much love and attention as any other aspect. Adding a question is achieved by logging in as a teacher, clicking the ‘Create Question’ button, filling out the form and clicking submit.

“Provide an account system, whereby each student has a unique account that will track wins and losses”

A comprehensive account system is provided in which wins, losses and draws are tracked, as well as what game they were in and who the other player was. Achievements are also present as a way of tracking progress, although there is only one student achievement as it is. Much like the question creation functionality, my plans for a rudimentary staff user experience again expanded into unique staff account as well, tracking question creation and featuring achievements.

“Provide a rudimentary method of creating and deleting accounts”

Deleting accounts is something I didn’t have the time to implement, but account creation was another instance of expanding plans of a rudimentary experience into one that suited the rest of the program. Creating accounts is a case of logging in as

a teacher, clicking the ‘Add Student’ button, filling out the form and clicking submit. Teacher accounts have to entered in via the first-time setup or by manually adding a record to the database.

***

The following objectives were to be considered additional refinements after the other objectives were satisfied.

“Providing more games, such as squares or rock, paper, scissors”

I didn’t have time to implement more than the three games that are present, although one of them is Rock, Paper, Scissors. Given more time other games could easily be implemented, or network modes of the existing games.

“Providing a way of saving games and restoring them at a later date”

The possibility of this functionality is present in the network Connect Four games, although more as an unintended side-effect than anything else. In a product for release to the public, a save/load functionality could be considered, although it is unlikely any game would last long enough to make it worthwhile.

“Providing more detailed student accounts, with information such as which subject they are doing best at”

This stat-tracking functionality never made it in to the final product, but would be straightforward to add at a later date as much of the necessary data is already stored in the database.

“Provide a more complex and visually appealing system for monitoring student progress, inputting new questions and creating/deleting accounts”

Account deleting notwithstanding, as the functionality isn’t present in the program as it is, all of the other functionalities received expansion in their methods that brought them up to par with the rest of the program.

“Some form of achievement system”

There is an achievement system present for both student and staff users, although there is only one and two achievements programmed in, respectively. Adding more is simply a case of writing an SQL string and adding a subroutine for the achievement, however.

“Support for teams (similar to how clans work in online gaming)”

I made no attempt at adding this, although looking at it now it would be relatively easy to implement. The use of a clan system in a school edutainment suite is limited, however.

Solution Satisfies the Evaluation Criteria

In the Design section of this project I outlined a series of evaluation criteria for the final project. These revolved around more easily quantifiable tests, such as time taken to load the program.

***

"A new user, who has never experienced the program before and who is at least slightly computer literate, must be able to complete the first-time setup in less than five minutes with no assistance."

To test this, I gave the program to a user unfamiliar with the program, along with a list of data to input, such as a student, a staff member and a subject. It took them 4:31 to finish the first-time setup.

"The user must be able to start and both win and lose a game of Hangman, with the appropriate result recorded in his account's details, within 10 minutes."

To test this, I got the same user to start a game of Rock, Paper, Scissors, which I replaced Hangman with, and win a game against me. I repeated this, telling them to lose. Afterwards we examined both our accounts' details. It took 4:23 to finish.

"The user must be able to start and win, lose and draw a game of Noughts & Crosses in hotseat mode, with the appropriate result recorded in his account's details, within 10 minutes."

To test this I repeated the previous test, substituting Noughts & Crosses for Rock, Paper, Scissors. It took 6:55 to finish.

"The user must be able to log in as both a student and as a teacher in under a minute."

To test this, I got the increasingly familiar user from earlier and told them to log in with student login credentials I gave them. Then I told them to close the program and repeat this with staff login credentials. It took 54 seconds.

"The user must be able to view his account details within 30 seconds of the Home form being displayed."

This took the user only 5 seconds.

"Both the student and the teacher must have distinct and tailored experiences: a teacher must be able to view the details of a specific student within 2 minutes; a student must be able to start a game within 2 minutes"

I got the user to login as a student and start a game of Noughts and Crosses up, getting to the game form in 1:44. I then got them to login as a staff member and look up themselves, taking 1:50

“The program will work both in school and out of it.”

I took the program .exe home with me and tried to run it. It connected to the database and functioned as usual.

Future Improvements

Possible improvements to the program would include adding missing functionality, such as the deletion of accounts/questions/subjects, the changing of account passwords, more achievements, a special admin account that can access the creation functionality present in the first-time setup, an expanded list of games including Hangman, Squares or perhaps even Chess, the saving and loading of games and replays of games, leaderboards and clan support and network versions of all the games. Something else that could do with work is the question system, allowing different types of questions, such as multiple choice questions, as well as making the answer inputting less finicky, as it currently requires the user to capitalise their answer, get the perfect spelling and clairvoyantly know which definite articles the teacher did or did not use when creating the question.

Another possible improvement, although it would be a large undertaking, would be to port the program to a web language, such as PHP, in order to make it largely platform-independent. The use of a MySQL database, rather than an Access database, would make this transition less painful, as the database is already platform-independent.

Other, smaller cosmetic improvements could include 3D graphics powered by OpenGL, music and sounds, a fullscreen mode with forms that resize correctly to various resolutions and perhaps some form of tutorial, although the program is probably intuitive enough that it wouldn’t be overly useful.

Minor improvements could also be made to the code itself, such as more use of the With statement to group together multiple methods of the same class, more use of subs to handle tasks that crop up repeatedly in the program and more functions for some of the mathematical operations, but these would be minor improvements, more noticeable if at all by a programmer looking at the source code than the end user. A considerable amount of work could be put into making the Connect Four network play more stable, as it is prone to a number of collapses as it is.

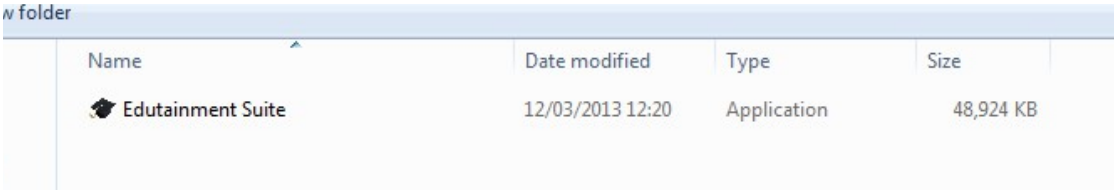
User Documentation

Installation

The following section will explain to the end user how to install their brand new edutainment suite, ‘Are You Not Edutained⁉’

1. Find the installation disk that came with your product.
2. Insert the disk into your DVD drive
3. Double-click on the ‘Computer’ icon on the desktop

4. Double click on the icon for the disk drive you inserted the disk into, by default likely E:
5. Move the .exe file on the disk to a location on the school network where everyone can access it

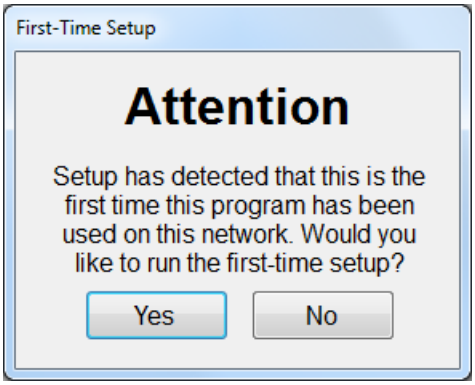


After this, to run the program you simply need to double-click on the .exe each time. The first time you will be prompted to complete the first-time setup, during which you will input the details of staff, students, subjects etc., so it is recommended that you take some time out of your schedule, make a cup of tea and get comfortable beforehand.

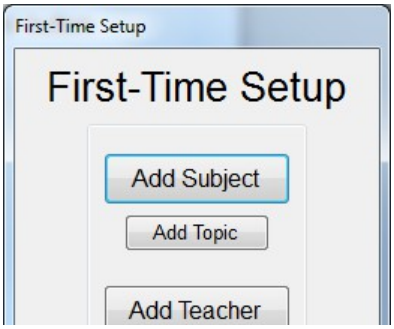
Use

Section 1: First-Time Setup

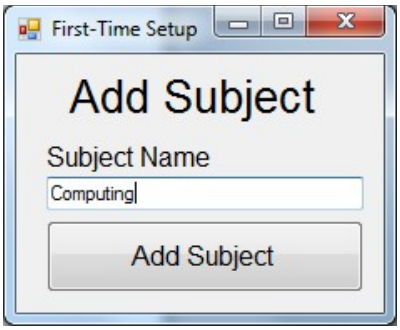
1. Run the .exe file for the first time
2. In the message that appears after the splash screen, click the ‘Yes’ button



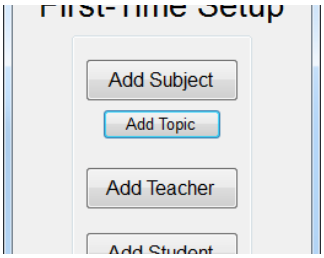
3. Click the ‘Add Subject’ button



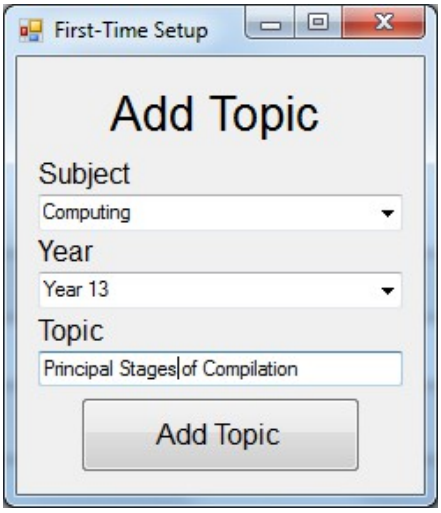
4. Type in the subject name and click the ‘Add Subject’ button



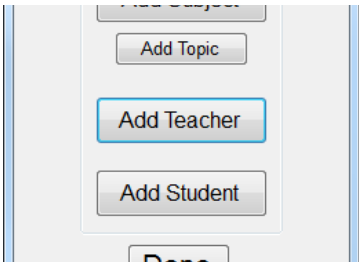
5. Repeat for all subjects, then click the red cross in the top-right
6. Click the ‘Add Topic’ button



7. Input the data and click the ‘Add Topic’ button and click the ‘Yes’ button in the message box that appears



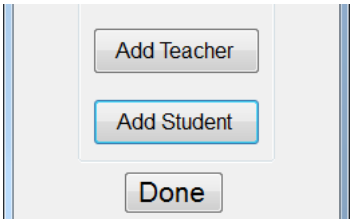
8. Repeat for all topics, then click the red cross
9. Click the ‘Add Teacher’ button



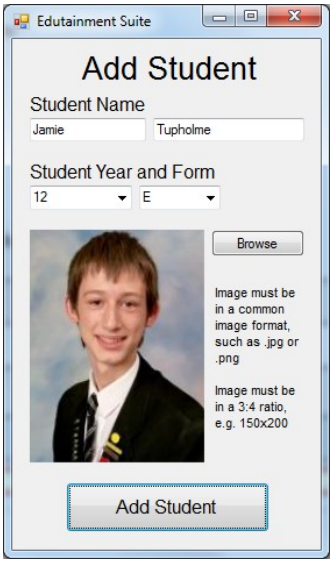
10. Input the data and click the ‘Add Teacher’ button and click the ‘Yes’ button in the message box that appears

11. Repeat for all teachers, then click the red cross

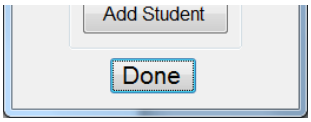
12. Click the ‘Add Student’ button



13. Input the data and click the ‘Add Student’ button and click the ‘Yes’ button in the message box that appears



14. Repeat for all teachers, then click the red cross
15. Click the ‘Done’ button

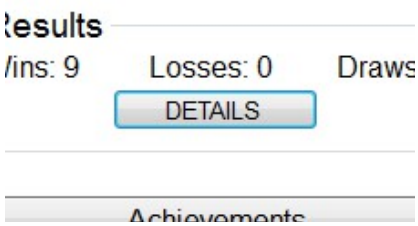


Note: To add subjects, topics and teachers later, access the database and manually create the record using phpMyAdmin

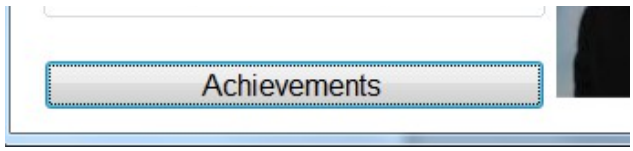
Section 2: Checking Accounts

1. Log in
2. Click the ‘View Profile’ button in the top-right

3. Click the ‘Details’ button to view the user’s breakdown



4. Click the ‘Achievements’ button to view the user’s achievements



Note: The process is the same for both staff and student accounts

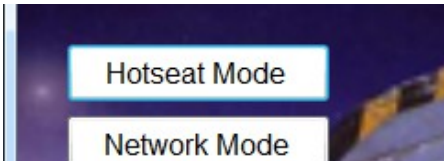
Section 3: Playing Games as a Student

Sub-Section 1: Playing Connect Four

1. Log in as a student
2. Click the ‘Connect Four’ button

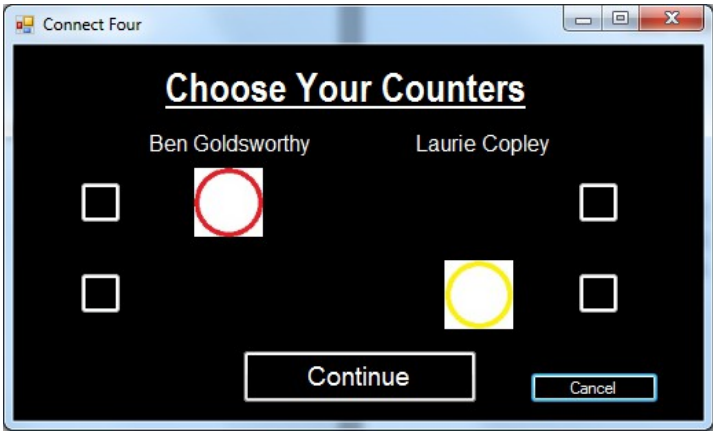
Sub-Sub-Section 1: Playing Connect Four Hotseat

1. Click the ‘Hotseat Mode’ button

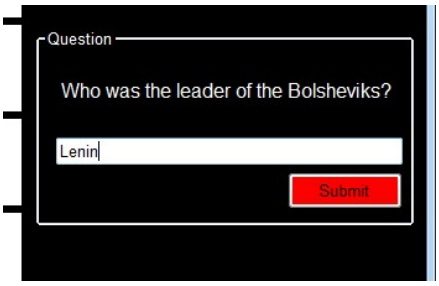


2. Login as another student
3. Select the subject and topic you want to be quizzed on

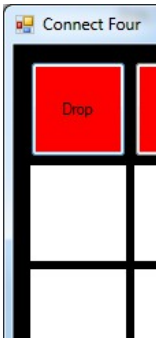
4. Allocate the counters and click the ‘Continue’ button



5. Have the current player answer the question and click the ‘Submit’ button



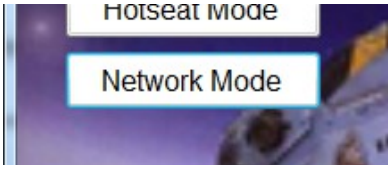
6. Click the ‘OK’ button. If the question was answered incorrectly, go to step 7. Otherwise, choose a column to drop the counter in and click the button to drop



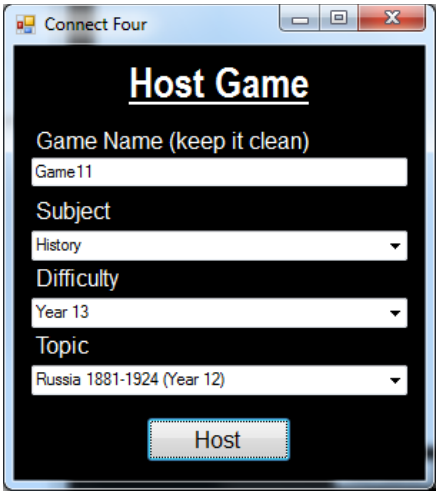
7. Repeat until the game has ended, click the ‘OK’ button and then either the ‘Back’ button or the ‘Reset’ button

Sub-Sub-Section 2: Playing Connect Four Network

1. Click the ‘Network Mode’ button



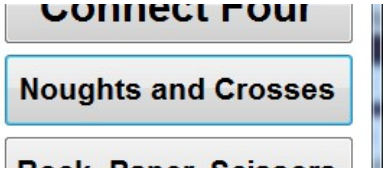
2. Either join a game by clicking on it in the list and clicking the ‘Join Game’ button, view only games you’re participating in by clicking the ‘Your Games’ button or click the ‘Host Game’ button to create a new game
3. Unless you clicked the ‘Host Game’ button, go to step 5. Otherwise, fill in the details and click the ‘Host’ button and the ‘OK’ button on the messagebox that appears



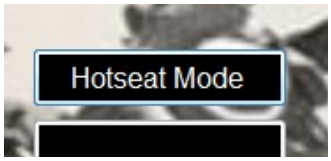
4. After another player has joined the game, you can too
5. Play through the game as you would with a hotseat game. Once the game has ended, click the ‘Quit Game’ button

Sub-Section 2: Playing Noughts and Crosses

1. Log in as a student
2. Click the ‘Noughts and Crosses’ button



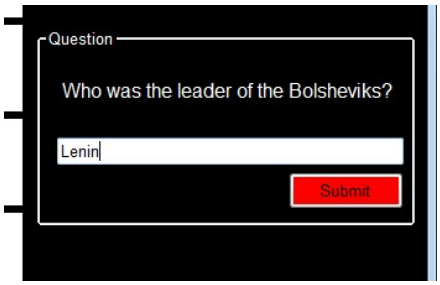
3. Click the ‘Hotseat Mode’ button



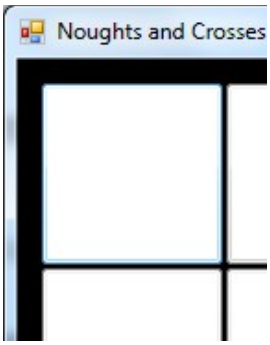
4. Login as another student
5. Select the subject and topic you want to be quizzed on

6. Allocate the counters and click the ‘Continue’ button

7. Have the current player answer the question and click the ‘Submit’ button



8. Click the ‘OK’ button. If the question was answered incorrectly, go to step 9. Otherwise, choose a square to place a counter and click it



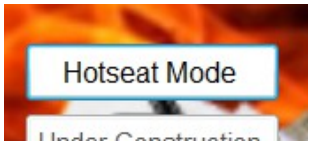
9. Repeat until the game has ended, click the ‘OK’ button and then either the ‘Back’ button or the ‘Reset’ button

Sub-Section 3: Playing Rock, Paper, Scissors

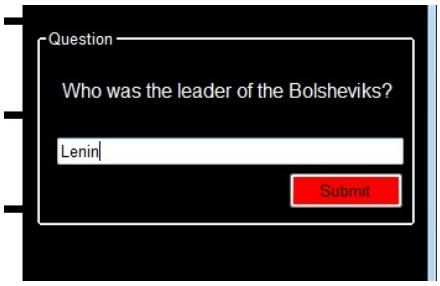
1. Log in as a student

2. Click the ‘Rock, Paper, Scissors’ button

3. Click the ‘Hotseat Mode’ button



4. Login as another student
5. Select the subject and topic you want to be quizzed on
6. Have the current player answer the question and click the ‘Submit’ button



7. Click the ‘OK’ button. If the question was answered incorrectly, go to step 8. Otherwise, choose a weapon and click it

8. Repeat for the other player. Watch the well-animated and pulse-pounding fight to the death. Click the ‘OK’ button on the messagebox that appears and then either go back to step 6 or click

the ‘Back’ button

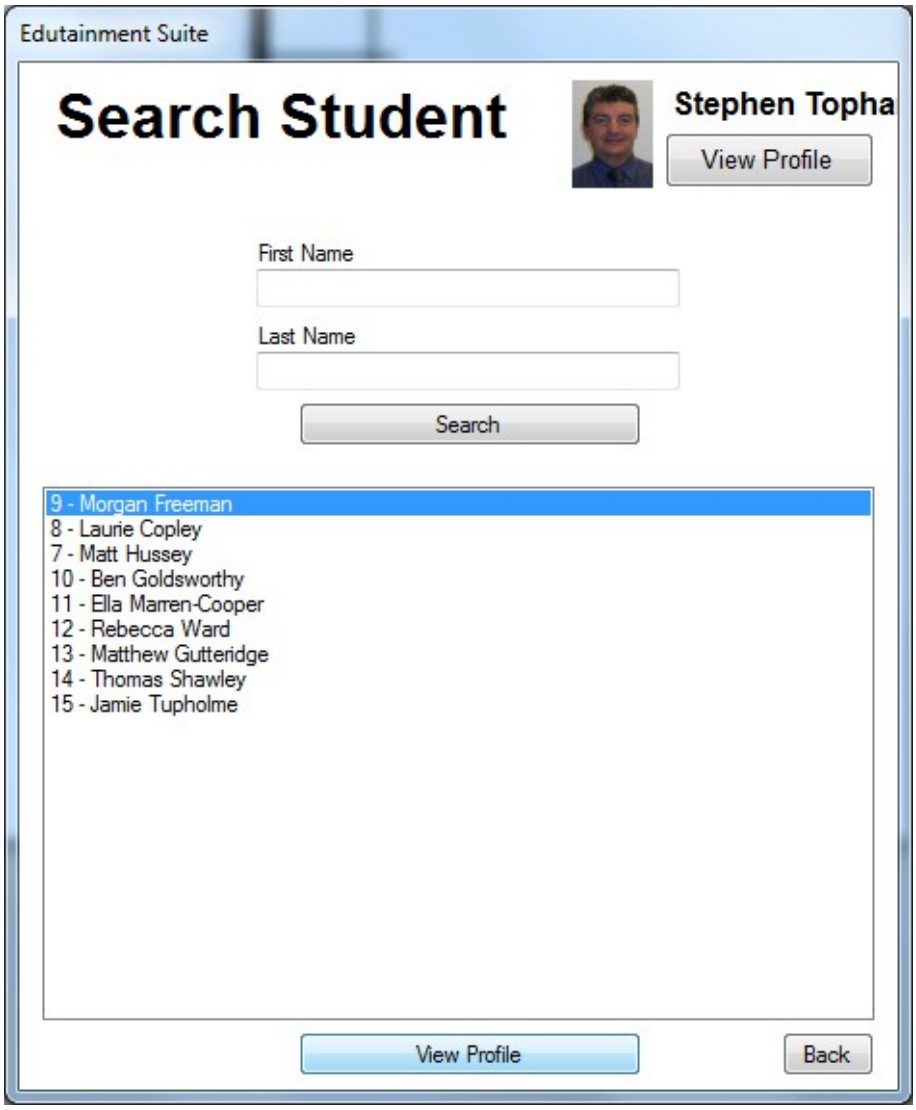
Section 4: Admin Work as a Teacher

Sub-Section 1: View a Student

1. Log in as a teacher
2. Click the ‘View Student’ button

3. Find the student you are after and click them, then click the ‘View Profile’ button. To speed up the process, you can type in their name, or part of it, above the listbox and click the ‘Search’ button to narrow

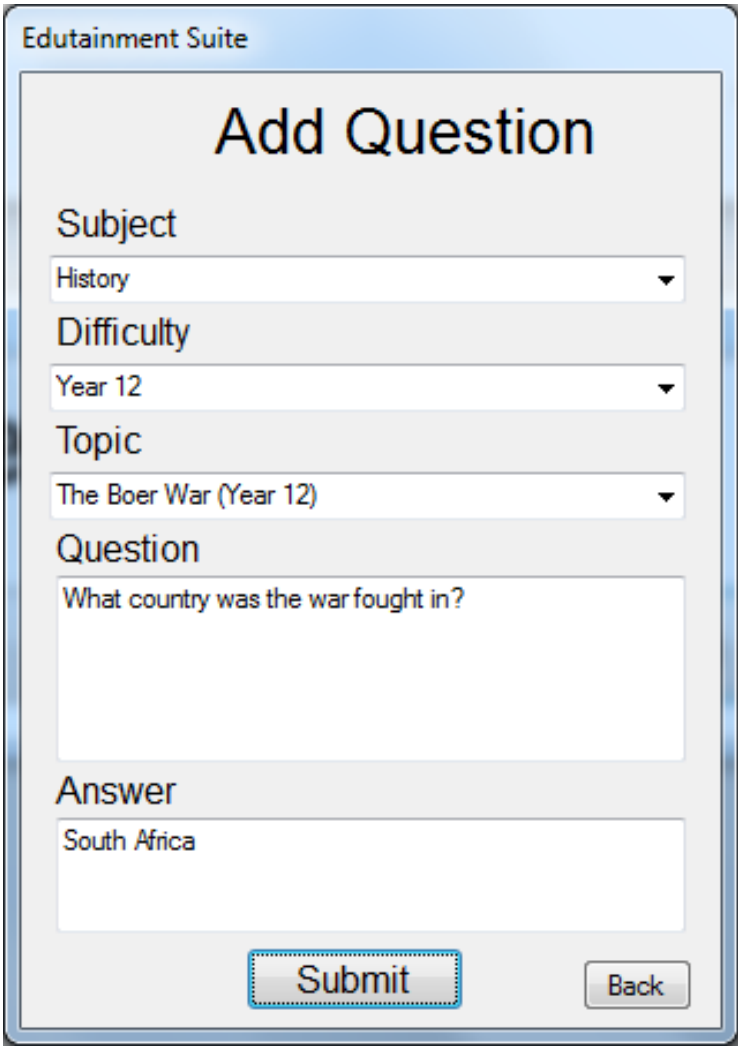
the search



Sub-Section 2: Create a Question

1. Log in as a teacher
2. Click the ‘Create Question’ button

3. Input the data and click the ‘Submit’ button and click the ‘Yes’ button in the message box that appears



GNU Free Documentation License

Version 1.3, 3 November 2008

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in

part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

The "publisher" means any person or entity that distributes copies of the Document to the public.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location

from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
- C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
- D. Preserve all the copyright notices of the Document.
- E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
- H. Include an unaltered copy of this License.
- I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in

the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

- J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
- K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
- L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
- M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.
- N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.
- O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic

form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit

to the present version, but may differ in detail to address new problems or concerns. See http://www.gnu.org/copyleft/.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Document.

11. RELICENSING

"Massive Multiauthor Collaboration Site" (or "MMC Site") means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A "Massive Multiauthor Collaboration" (or "MMC") contained in the site means any set of copyrightable works thus published on the MMC site.

"CC-BY-SA" means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

"Incorporate" means to publish or republish a Document, in whole or in part, as part of another Document.

An MMC is "eligible for relicensing" if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

> Copyright (C) YEAR YOUR NAME.
> Permission is granted to copy, distribute and/or modify this document
> under the terms of the GNU Free Documentation License, Version 1.3