My Project Report – Connect 4 against the computer. Jamie Wilkins

# 3.1.1

I believe that my project is suitable for both mathematics or computer science teachers for lower schools because they can use it in their lessons in order to allow their students to develop their thinking skills and it will also force them to think computationally as they will have to abstract the issue with the problem and then have to think about how each of their actions will affect the current situation. For example what will the player do if the computer puts a piece in a certain position? This means that the students will have to think ahead and plan an adaptable strategy to use. I believe that this project can be made complex by many different methods. To begin with the computer will be a complex AI because it will adapt to a situation. What this means is that there will not be a specific path or strategy the computer will use. Otherwise if there was then it will be completely predictable and you would know exactly how to beat it. I would also need to make a graphical grid because then you can visualise what is happening easier and it feels natural and close to the actual game. This grid will be the same size as the original which is seven slots wide by six slots high. I also need to make a grid because if I don’t then there are two issues that could happen. The first is the fact that you could see counters overlapping each other and not going in the requested position which therefore makes the game unplayable. This is key in the testing section of the project. The second is the fact that there needs to be a specific grid size and not allow counters to overspill. This means that there must be an error message if someone tries to overspill the grid and not allow them to place the counter. This will also mean that there is only a certain amount of moves that can be made. With only having a specific move limit this means that there is a potential for a draw and not just a win or lose. The student will be able to select different difficulties of AI to play against that will adapt to situations differently. For example, if the student decides to play on an easy difficulty and the computer has a clear chance to win, the computer may decide to not put it in the position to win, whereas if the player decides to play on a hard difficulty then the computer will learn how to block you and also know when it has a winning potential.

# 3.1.2

As I have already mentioned, my project stakeholder will be either mathematics or computer science teachers in lower schools. These will be the clients that shall be requesting my project to be made. The project is appropriate for them because it shall include allow the teachers to use a different method to allow their students to learn new skills in a more interactive and fun way. The users will be the students of the teachers. This is because it can teach them different skills such as strategizing and thinking ahead in order to achieve the best possible move. By doing this their skills of planning will be developed and used in their lives all of the time. A teacher may use it in order to determine which difficulties for each student to use due to their skills they already have. Each student will have their different needs as it will have a varying difficulty feature meaning that students that are more able do not have to stick to a difficulty that is lower than their skill level.

|  |  |  |
| --- | --- | --- |
| Type of user | Use it for? | Relevant Knowledge (user skill) |
| Teacher | A teacher would use the program in order to test the appropriate skill levels in order to determine which difficulty level will be appropriate for their students. | A teacher would have a lot of skills including the relevant ones in order to use the program. |
| KS3 Student | A KS3 student may play the game in order to improve their thinking skills. By this I mean that they will develop their skills of adapting to a situation and also their ability to plan ahead to perform a strategy towards success. | A KS3 student should have the relevant skills in order to use the program as it is just a matter of clicking a position to drop your piece. The only thing they may need help with is deciding what their relevant skill level is for the difficulty decider. However this will be decided by the teacher in the first instance. |
| KS4 Student | A KS4 student may play the game in order to improve their thinking skills. By this I mean that they will develop their skills of adapting to a situation and also their ability to plan ahead to perform a strategy towards success. | A KS4 student should have the relevant skills in order to use the program as it is just a matter of clicking a position to drop your piece. The only thing they may need help with is deciding what their relevant skill level is for the difficulty decider. However this will be decided by the teacher in the first instance. |

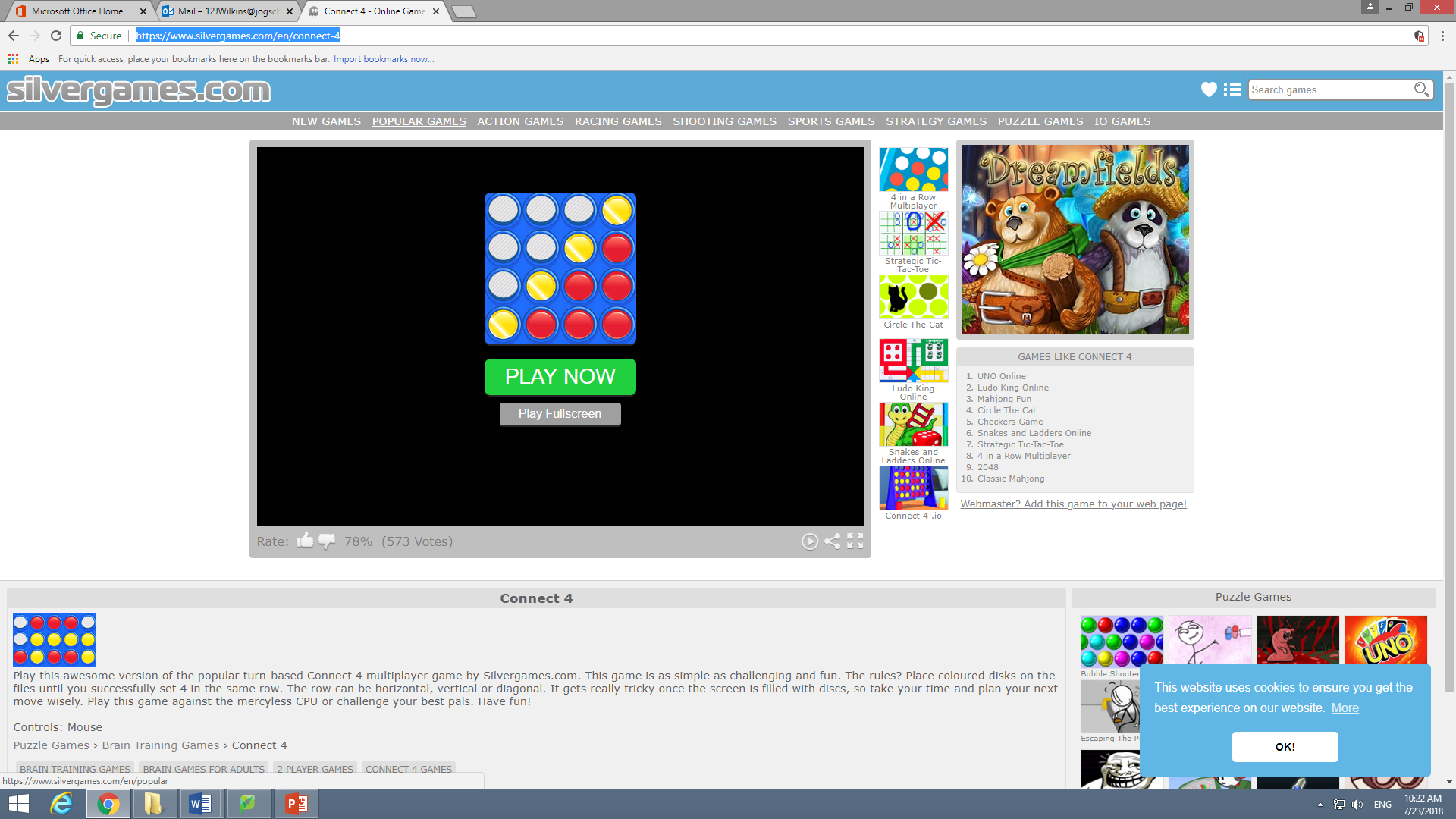
# 3.1.3

Online connect four games are widely available across the internet. For example, there are some available at places such as:

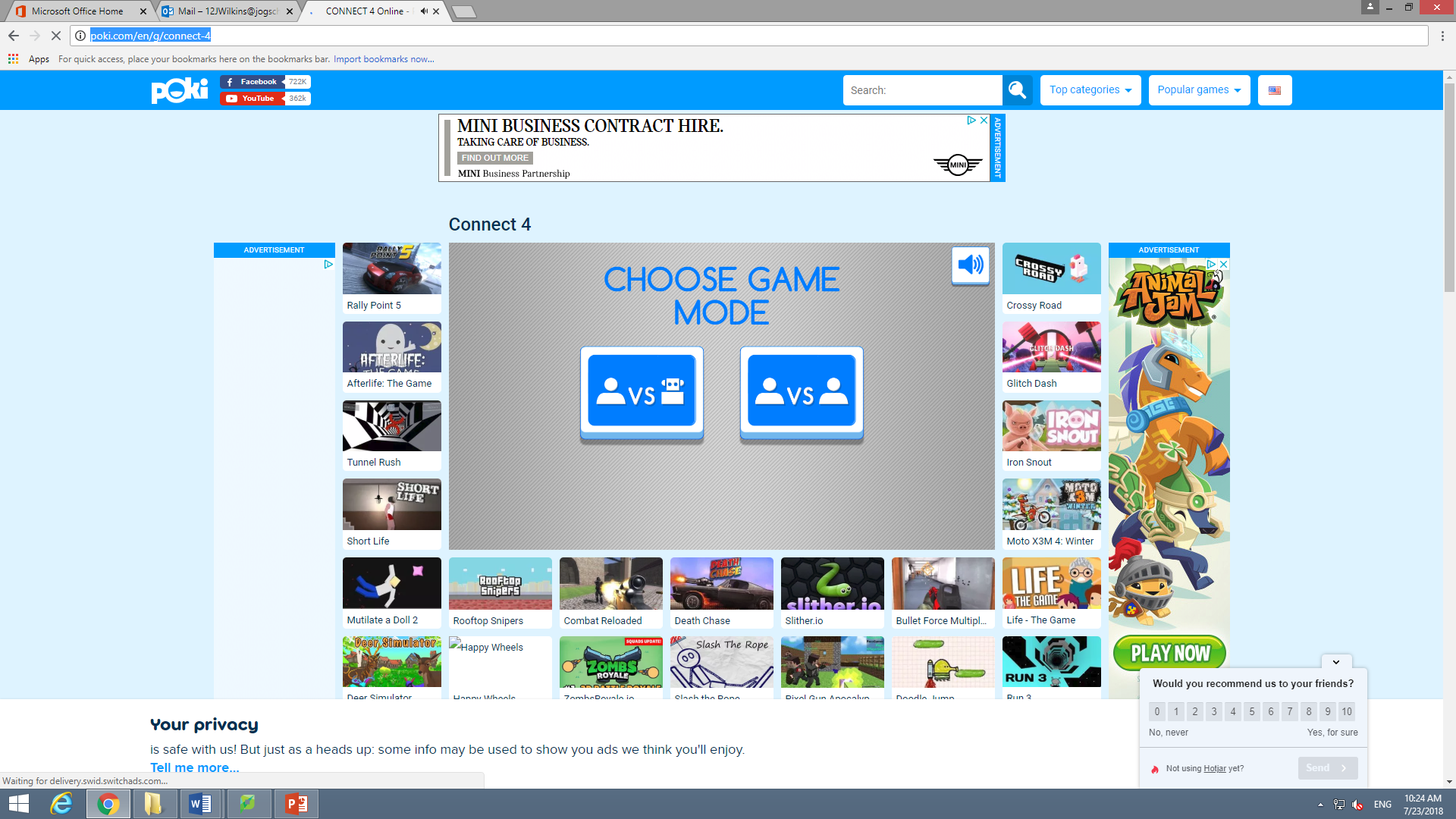
<https://www.mathsisfun.com/games/connect4.html>



<https://www.silvergames.com/en/connect-4>



<http://poki.com/en/g/connect-4>



As you can see from above, only 1 of the three examples I have shown have a feature where you can change the difficulty of the computer player. In order to determine whether either the clients or the user are able to use the program, I will use a questionnaire as provided below in order to determine whether I need to make any improvements or any kind of feature that will allow all users to access the program:

What is your knowledge of connect four?

………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

Is there a specific colour that would be helpful for the background colour?

…………………………………………………………………………………………………………………………………………………………………………………………………………………………

If you don’t know how connect four works, would it be helpful for the rues of the game to be presented to you at the beginning before attempting the game against the computer?

* Yes [ ]
* No [ ]

Would having the rules being told to you be more helpful rather than reading it off the screen?

* Yes [ ]
* No [ ]

Would recording accounts be helpful in order to store your score?

* Yes [ ]
* No [ ]

Would you like the score to be equivalent to the number of games won in the lesson or whether it records how many games you won in a row?

* Total score [ ]
* Win Streak [ ]

Would a leaderboard feature be helpful in order to track how well each student is doing? This will include the difficulty level they played on and you can therefore track whether you could recommend if a student should change their difficulty level.

* Yes [ ]
* No [ ]

Would it be helpful to have a colour selection feature for the tokens in order for everyone to access the game?

* Yes [ ]
* No [ ]

# 3.1.4

Firstly, I need to create a success criteria. This will include:

* Whether the game will run.
* Whether the AI will respond well.
* Whether I can login.
* Whether the colours are correct for every user (ability to change the colours).

In order to run the program, the minimum requirements you need to run is:

* Windows 2000
* Python 3.4
* Pygame
* Any form of computer from a raspberry pi up to a top of the line computer
* Keyboard
* Mouse

However, the recommended requirements you need to run is:

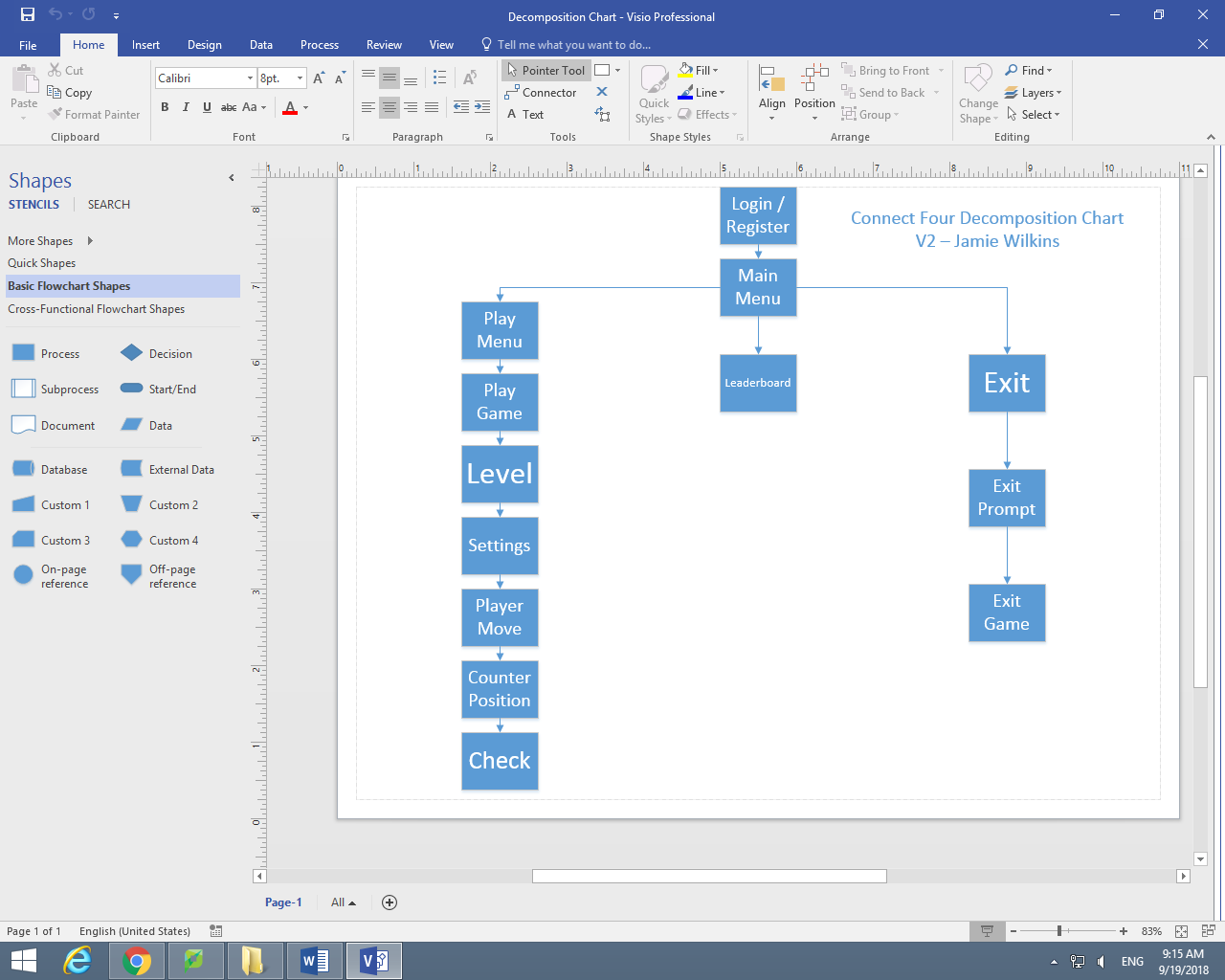
* Windows 8
* Python 3.4
* Pygame
* Any computer
* Keyboard
* Mouse

|  |  |  |
| --- | --- | --- |
| Aesthetics | Requirements | How is it measured? |
| 1 | Is the background a suitable colour? | Feedback from user questionnaire. |
| 2 | Score recorder | It will measure how many games won towards an account if there is positive feedback from the questionnaire for the account system. |
| 3 | Is having a leaderboard helpful? | Feedback from user questionnaire. |
| 4 | Is having a token colour selector helpful? | Feedback from user questionnaire. |

|  |  |  |
| --- | --- | --- |
| Inputs | Requirements | How is it measured? |
| 1 | User: inputting a username and password that is valid. | Whether the account details have been taken already with a True or False measure. This will be measured in the testing section. |
| 2 | Token placement validity. | Whether the position of the token can be played with a True or False measure. For example, if the token would overspill the board This will be measured in the testing section. |
| 3 | User: Inputting the score achieved at the end of each lesson being recorded into the leader board. | This will be measured in the testing stage when I try to input a score and then check later at the scoreboard to see if the recorded core with the account appears. |

# 3.2.1

Below is my top down design (Decomposition chart). This is to show me how each of my different functions and how they intertwine with each other. This is useful because I can also use it as a checklist for me as I am creating each function.



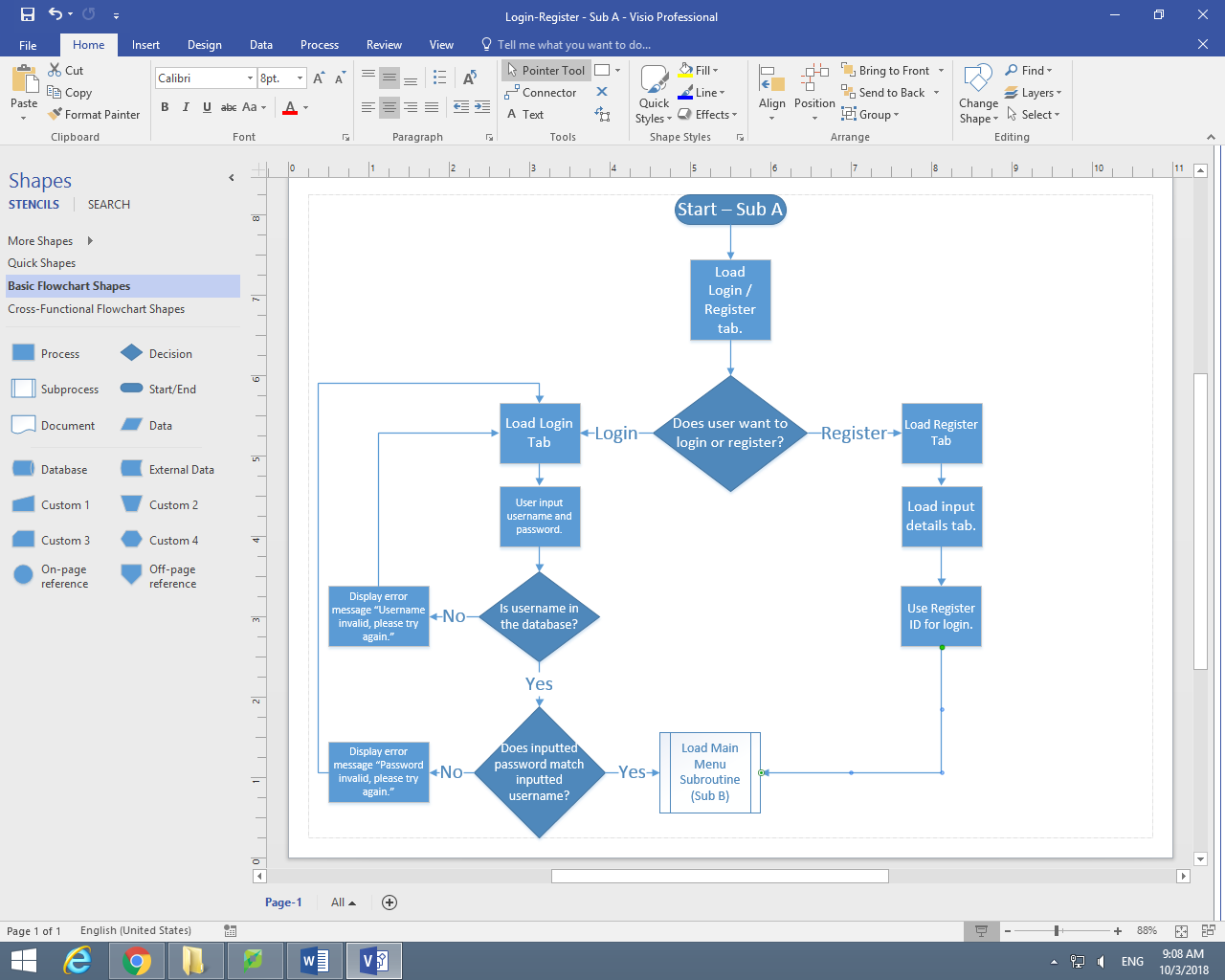
# 3.2.2 (A)

As you can see from above, the first step of my program is a login / register tab. The purpose is to allow different users to track their wins in a leaderboard. In order to do this they must log into their accounts. Also once an account has been made then their detail will be stored in an external database and then they will be able to simply log into their account instead of just having to register as it will recognise the correct details in the database. This will then lead to the main menu, where there will be three options, one is to start playing games, one to load up the leaderboard and the other is to exit the game. As you can see from above, if the user selects to exit an exit prompt will be displayed giving the user the option return to main menu or to exit. If they choose return to main menu then they will be returned to the main menu and can select a different option. If they choose exit then the game will be closed down. It is important to display a prompt allowing the user to return to the main menu because if you do not then the user will have to login into the game again and that can be time consuming and an annoyance because they may have never intended to click exit. Another feature of the menu is to select the leaderboard. This will take you to a leaderboard in which you can see how you rank compared to your class or within the school. You will be ranked either depending on the number of wins achieved in each difficulty or depending on your highest win streak depending on your difficulty. This will add a competitive nature in the classroom and therefore each student will try to beat one another to get to the top and therefore increases their skills and focus during class. The final section, the main section, is the play menu. When the player selects this, they will then be taken to another menu, which will have different options to click. When the user clicks play game, the game will play with the current selections and settings active. In order to change the settings the player will have to click settings. In the settings section, the player will be able to change the colour of the background they are playing on as well as the colour of their tokens and the computer’s tokens. I have added this function because it allows people who are colour blind or have sight difficulties to see the counters they need to help them. The level selection will allow the player to change the difficulty the computer will be playing at. When play game has been selected, the game will start and as you can see from the decomposition chart, the game will play when a player is notified it is their move. Then they will place the counter. Finally, it will check to see if you have four in a row.

# 3.2.2 (B)

Below are my flowcharts and the pseudo code for each of the problems (subroutines) in the game. There will be a flowchart and then there will be the pseudo code for each subroutine.:

## Login / Register:



Login/Register()

LOAD login/register tab

IF USER clicks Login Button:

LOAD Login Tab

Username = input(“Input your Username”)

Password = input(“Input your Password”)

IF Username in Database:

IF Password == Password assigned to Username in Database:

MainMenu()

ELSE:

PRINT (“Password invalid, please try again.”)

Login/Register()

ELSE:

PRINT(“Username invalid, please try again.”)

Login/Register()

#(USER clicks Register Button)

ELSE:

LOAD Register Tab

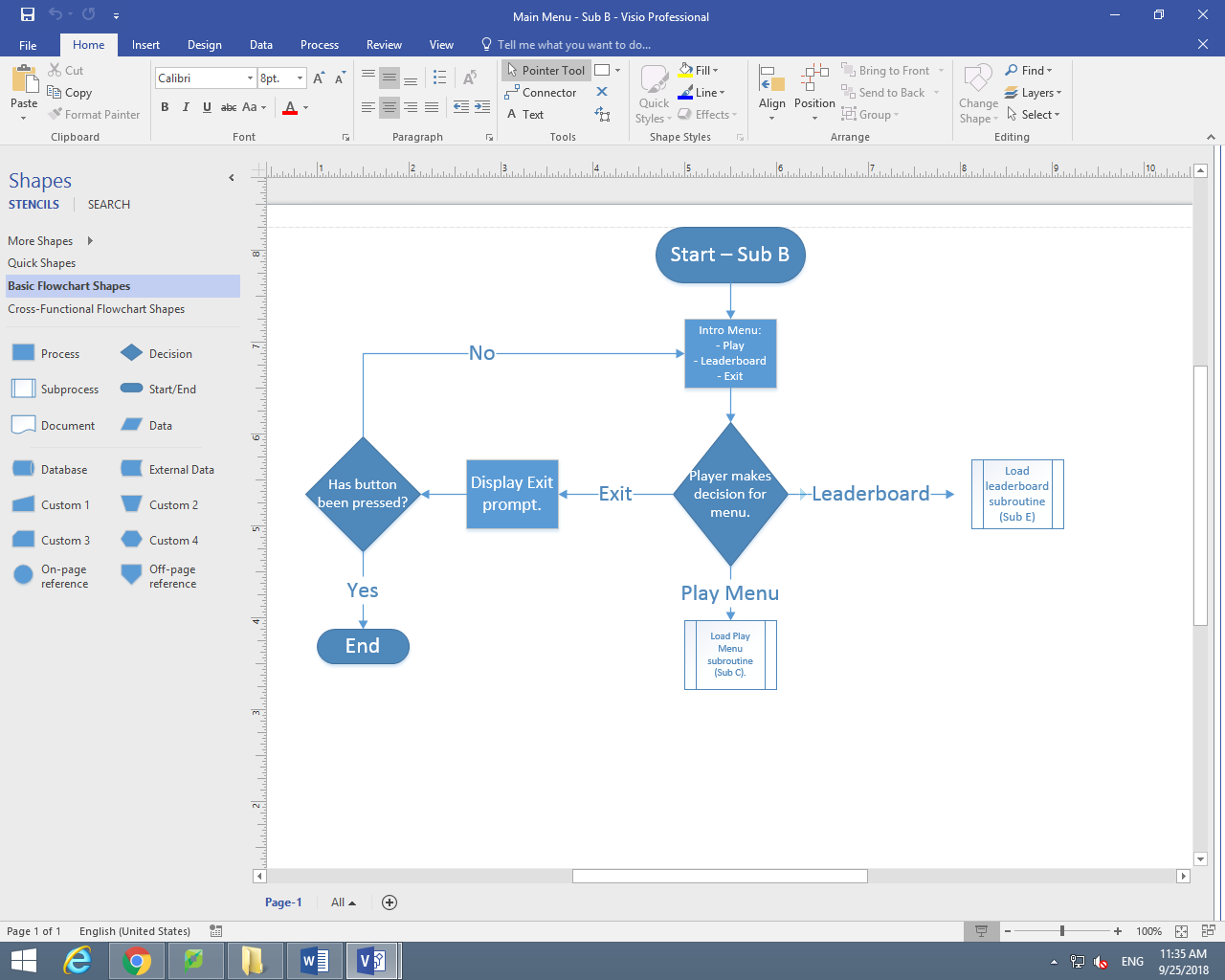
Username = input(“Choose Username. There must be one number, one capital letter and one special character. It also cannot be more than twelve characters.”)

Password = input(“Choose Password. There must be one number, one capital letter and one special character. It also cannot be more than twelve characters.”)

PRINT (“Now you can login. Use the Login button.”)

Login/Register()

## Main Menu:



MainMenu()

LOAD Intro Menu

IF USER clicks Play button:

PlayMenu()

ELIF USER clicks Leaderboard button:

Leaderboard()

#(USER clicks Exit button)

ELSE:

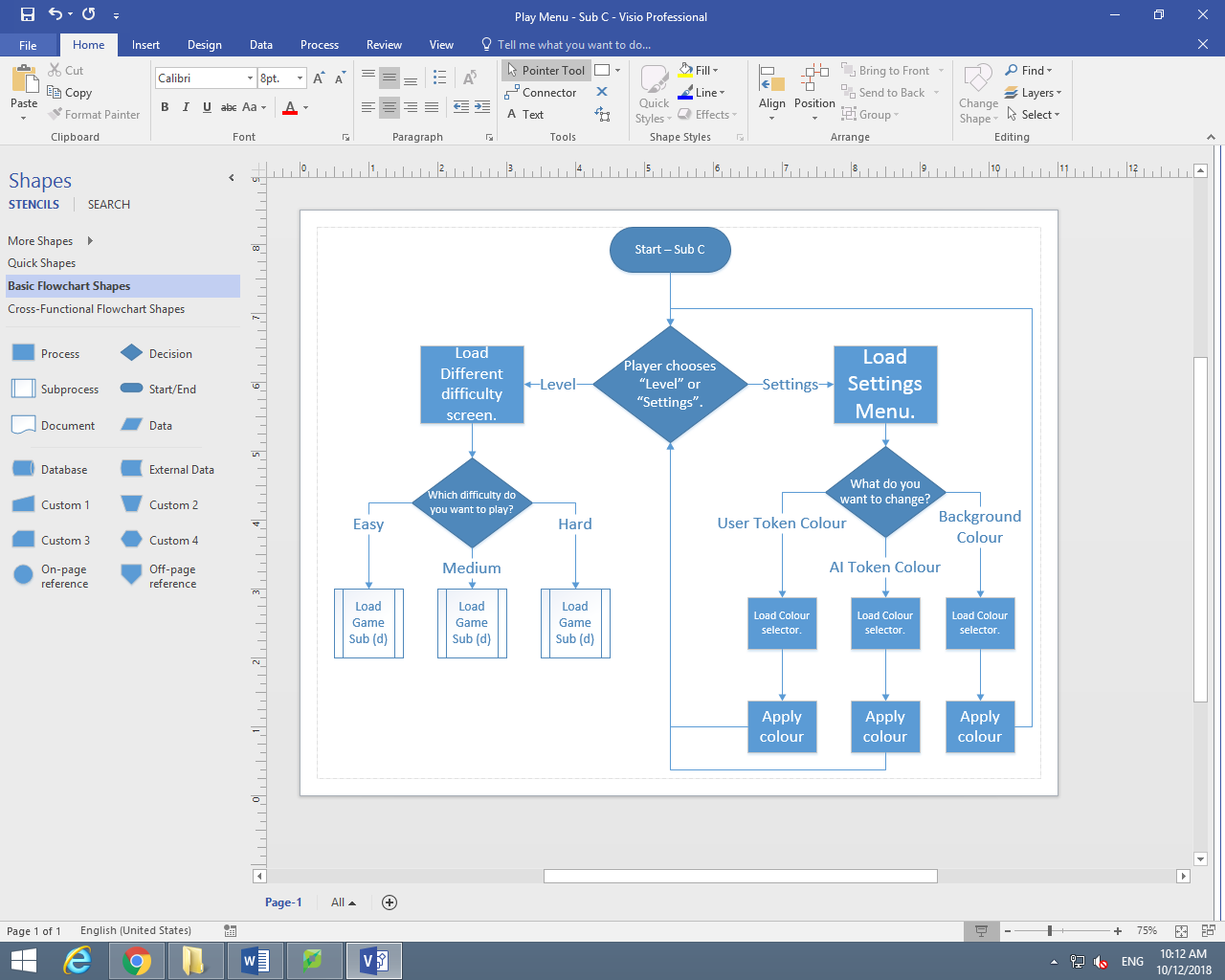
LOAD Exit tab

IF USER clicks yes:

END

ELSE:  
 MainMenu()

## Play Menu:



PlayMenu()

LOAD Level/Settings tab

IF USER clicks Level button:

LOAD DifferentDifficulty tab

IF USER clicks Easy button:

TheGame()

ELIF USER clicks Medium button:

TheGame()

#(USER clicks Hard button)

ELSE:

TheGame()

(USER clicks Settings)

ELSE:

LOAD Settings Tab

IF USER clicks User Token Colour:

LOAD UserTokenColour Tab

IF USER clicks UserRedApply:

UserToken = Red

PlayMenu()

ELIF USER clicks UserBlueApply:

UserToken = Blue

PlayMenu()

ELIF USER clicks UserBlackApply:

UserToken = Black

PlayMenu()

ELIF USER clicks UserOrangeApply:

UserToken = Orange

PlayMenu()

ELIF USER clicks UserPurpleApply:

UserToken = Purple

PlayMenu()

ELIF USER clicks UserBrownApply:

UserToken = Brown

PlayMenu()

ELIF USER clicks UserYellowApply:

UserToken = Yellow

PlayMenu()

ELIF USER clicks UserGreenApply:

UserToken = Green

PlayMenu()

ELIF USER clicks UserGreyApply:

UserToken = Grey

PlayMenu()

#(USER clicks UserPinkApply)

ELSE:

UserToken = Pink

PlayMenu()

ELIF USER clicks AI Token Colour:

IF USER clicks AIRedApply:

AIToken = Red

PlayMenu()

ELIF USER clicks AIBlueApply:

AIToken = Blue

PlayMenu()

ELIF USER clicks AIBlackApply:

AIToken = Black

PlayMenu()

ELIF USER clicks AIOraneApply:

AIToken = Orange

PlayMenu()

ELIF USER clicks AIPurpleApply:

AIToken = Purple

PlayMenu()

ELIF USER clicks AIBrownApply:

AIToken = Brown

PlayMenu()

ELIF USER clicks AIYellowApply:

AIToken = Yellow

PlayMenu()

ELIF USER clicks AIGreenApply:

AIToken = Green

PlayMenu()

ELIF USER clicks AIGreyApply:

AIToken = Grey

PlayMenu()

#(USER clicks AIPinkApply)

ELSE:

AIToken = Pink

PlayMenu()

#(USER clicks Background Colour)

ELSE:

IF USER clicks BackgroundRedApply:

Background = Red

PlayMenu()

ELIF USER clicks BackgroundBlueApply:

Background = Blue

PlayMenu()

ELIF USER clicks BackgroundBlackApply:

BackgroundToken = Black

PlayMenu()

ELIF USER clicks BackgroundOraneApply:

BackgroundToken = Orange

PlayMenu()

ELIF USER clicks BackgroundPurpleApply:

BackgroundToken = Purple

PlayMenu()

ELIF USER clicks BackgroundBrownApply:

BackgroundToken = Brown

PlayMenu()

ELIF USER clicks BackgroundYellowApply:

BackgroundToken = Yellow

PlayMenu()

ELIF USER clicks BackgroundGreenApply:

BackgroundToken = Green

PlayMenu()

ELIF USER clicks BackgroundGreyApply:

BackgroundToken = Grey

PlayMenu()

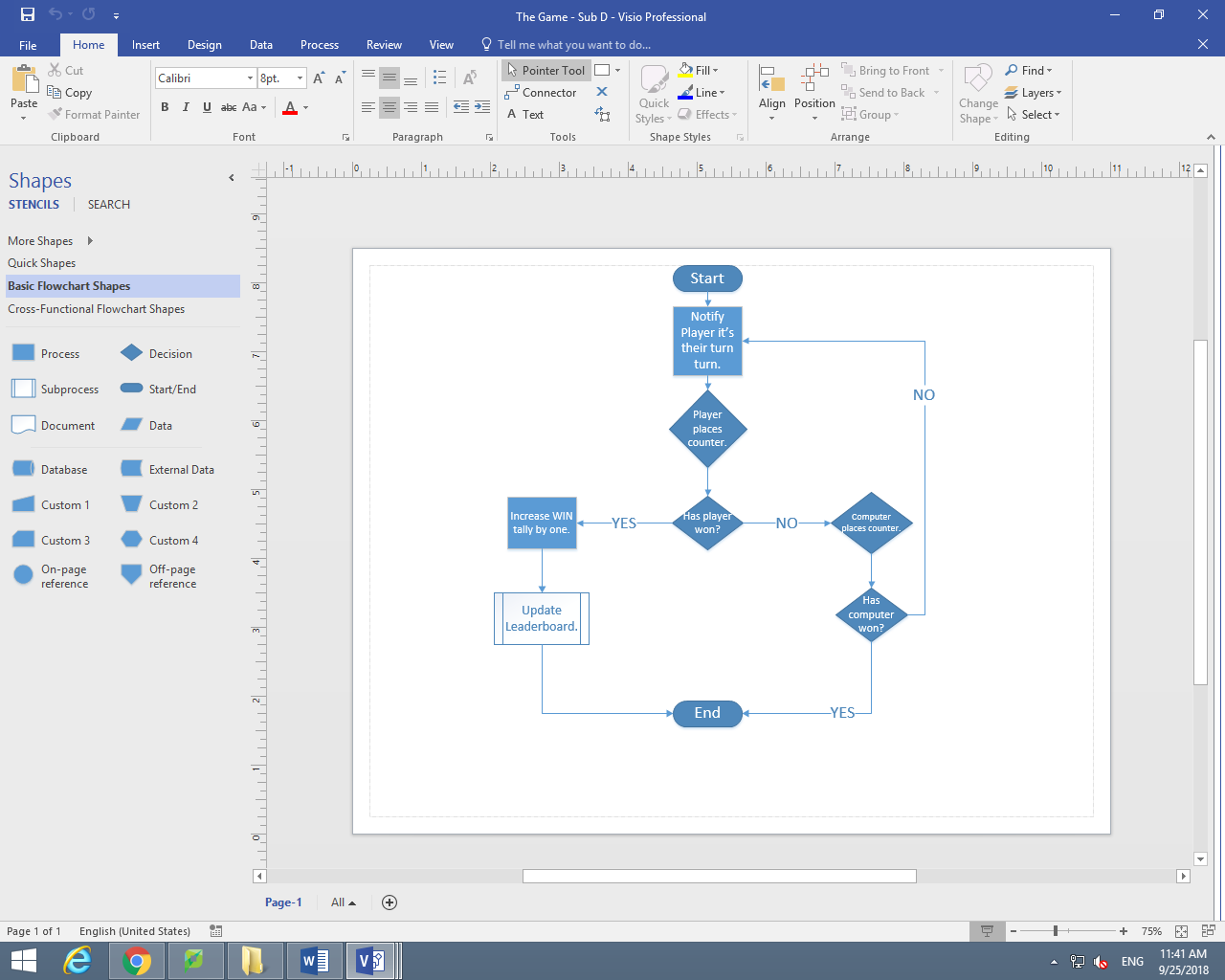
#(USER clicks BackgroundPinkApply)

ELSE:

BackgroundToken = Pink

PlayMenu()

## The Game:



TheGame()

PRINT (“It is your turn.”)

IF Coordinate(0,0) == 0:

Coordinate(0,0) = 1

DOES 4 in a row == True

IF YES:

TotalWin = TotalWin + 1

WinStreak = WinStreak + 1

ELSE:

Coordinate(0,0) = 2

DOES 4 in a row == True

IF YES:

PRINT (“Unlucky, try again.”)

ELSE:

TheGame()

IF Coordinate(0,1) == 0:

Coordinate(0,1) = 1

DOES 4 in a row == True

IF YES:

TotalWin = TotalWin + 1

WinStreak = WinStreak + 1

ELSE:

Coordinate(0,1) = 2

DOES 4 in a row == True

IF YES:

PRINT (“Unlucky, try again.”)

ELSE:

TheGame()

IF Coordinate(0,2) == 0:

Coordinate(0,2) = 1

DOES 4 in a row == True

IF YES:

TotalWin = TotalWin + 1

WinStreak = WinStreak + 1

ELSE:

Coordinate(0,2) = 2

DOES 4 in a row == True

IF YES:

PRINT (“Unlucky, try again.”)

ELSE:

TheGame()

IF Coordinate(0,3) == 0:

Coordinate(0,3) = 1

DOES 4 in a row == True

IF YES:

TotalWin = TotalWin + 1

WinStreak = WinStreak + 1

ELSE:

Coordinate(0,3) = 2

DOES 4 in a row == True

IF YES:

PRINT (“Unlucky, try again.”)

ELSE:

TheGame()

IF Coordinate(0,4) == 0:

Coordinate(0,4) = 1

DOES 4 in a row == True

IF YES:

TotalWin = TotalWin + 1

WinStreak = WinStreak + 1

ELSE:

Coordinate(0,4) = 2

DOES 4 in a row == True

IF YES:

PRINT (“Unlucky, try again.”)

ELSE:

TheGame()

IF Coordinate(0,5) == 0:

Coordinate(0,5) = 1

DOES 4 in a row == True

IF YES:

TotalWin = TotalWin + 1

WinStreak = WinStreak + 1

ELSE:

Coordinate(0,5) = 2

DOES 4 in a row == True

IF YES:

PRINT (“Unlucky, try again.”)

ELSE:

TheGame()

IF Coordinate(0,6) == 0:

Coordinate(0,6) = 1

DOES 4 in a row == True

IF YES:

TotalWin = TotalWin + 1

WinStreak = WinStreak + 1

ELSE:

Coordinate(0,6) = 2

DOES 4 in a row == True

IF YES:

PRINT (“Unlucky, try again.”)

ELSE:

TheGame()

IF Coordinate(1,0) == 0:

Coordinate(1,0) = 1

DOES 4 in a row == True

IF YES:

TotalWin = TotalWin + 1

WinStreak = WinStreak + 1

ELSE:

Coordinate(1,0) = 2

DOES 4 in a row == True

IF YES:

PRINT (“Unlucky, try again.”)

ELSE:

TheGame()

IF Coordinate(1,1) == 0:

Coordinate(1,1) = 1

DOES 4 in a row == True

IF YES:

TotalWin = TotalWin + 1

WinStreak = WinStreak + 1

ELSE:

Coordinate(1,1) = 2

DOES 4 in a row == True

IF YES:

PRINT (“Unlucky, try again.”)

ELSE:

TheGame()

IF Coordinate(1,2) == 0:

Coordinate(1,2) = 1

DOES 4 in a row == True

IF YES:

TotalWin = TotalWin + 1

WinStreak = WinStreak + 1

ELSE:

Coordinate(1,2) = 2

DOES 4 in a row == True

IF YES:

PRINT (“Unlucky, try again.”)

ELSE:

TheGame()

IF Coordinate(1,3) == 0:

Coordinate(1,3) = 1

DOES 4 in a row == True

IF YES:

TotalWin = TotalWin + 1

WinStreak = WinStreak + 1

ELSE:

Coordinate(1,3) = 2

DOES 4 in a row == True

IF YES:

PRINT (“Unlucky, try again.”)

ELSE:

TheGame()

IF Coordinate(1,4) == 0:

Coordinate(1,4) = 1

DOES 4 in a row == True

IF YES:

TotalWin = TotalWin + 1

WinStreak = WinStreak + 1

ELSE:

Coordinate(1,4) = 2

DOES 4 in a row == True

IF YES:

PRINT (“Unlucky, try again.”)

ELSE:

TheGame()

IF Coordinate(1,5) == 0:

Coordinate(1,5) = 1

DOES 4 in a row == True

IF YES:

TotalWin = TotalWin + 1

WinStreak = WinStreak + 1

ELSE:

Coordinate(1,5) = 2

DOES 4 in a row == True

IF YES:

PRINT (“Unlucky, try again.”)

ELSE:

TheGame()

IF Coordinate(1,6) == 0:

Coordinate(1,6) = 1

DOES 4 in a row == True

IF YES:

TotalWin = TotalWin + 1

WinStreak = WinStreak + 1

ELSE:

Coordinate(1,6) = 2

DOES 4 in a row == True

IF YES:

PRINT (“Unlucky, try again.”)

ELSE:

TheGame()

IF Coordinate(2,0) == 0:

Coordinate(2,0) = 1

DOES 4 in a row == True

IF YES:

TotalWin = TotalWin + 1

WinStreak = WinStreak + 1

ELSE:

Coordinate(2,0) = 2

DOES 4 in a row == True

IF YES:

PRINT (“Unlucky, try again.”)

ELSE:

TheGame()

IF Coordinate(2,1) == 0:

Coordinate(2,1) = 1

DOES 4 in a row == True

IF YES:

TotalWin = TotalWin + 1

WinStreak = WinStreak + 1

ELSE:

Coordinate(2,1) = 2

DOES 4 in a row == True

IF YES:

PRINT (“Unlucky, try again.”)

ELSE:

TheGame()

IF Coordinate(2,2) == 0:

Coordinate(2,2) = 1

DOES 4 in a row == True

IF YES:

TotalWin = TotalWin + 1

WinStreak = WinStreak + 1

ELSE:

Coordinate(2,2) = 2

DOES 4 in a row == True

IF YES:

PRINT (“Unlucky, try again.”)

ELSE:

TheGame()

IF Coordinate(2,3) == 0:

Coordinate(2,3) = 1

DOES 4 in a row == True

IF YES:

TotalWin = TotalWin + 1

WinStreak = WinStreak + 1

ELSE:

Coordinate(2,3) = 2

DOES 4 in a row == True

IF YES:

PRINT (“Unlucky, try again.”)

ELSE:

TheGame()

IF Coordinate(2,4) == 0:

Coordinate(2,4) = 1

DOES 4 in a row == True

IF YES:

TotalWin = TotalWin + 1

WinStreak = WinStreak + 1

ELSE:

Coordinate(2,4) = 2

DOES 4 in a row == True

IF YES:

PRINT (“Unlucky, try again.”)

ELSE:

TheGame()

IF Coordinate(2,5) == 0:

Coordinate(2,5) = 1

DOES 4 in a row == True

IF YES:

TotalWin = TotalWin + 1

WinStreak = WinStreak + 1

ELSE:

Coordinate(2,5) = 2

DOES 4 in a row == True

IF YES:

PRINT (“Unlucky, try again.”)

ELSE:

TheGame()

IF Coordinate(2,6) == 0:

Coordinate(2,6) = 1

DOES 4 in a row == True

IF YES:

TotalWin = TotalWin + 1

WinStreak = WinStreak + 1

ELSE:

Coordinate(2,6) = 2

DOES 4 in a row == True

IF YES:

PRINT (“Unlucky, try again.”)

ELSE:

TheGame()

IF Coordinate(3,0) == 0:

Coordinate(3,0) = 1

DOES 4 in a row == True

IF YES:

TotalWin = TotalWin + 1

WinStreak = WinStreak + 1

ELSE:

Coordinate(3,0) = 2

DOES 4 in a row == True

IF YES:

PRINT (“Unlucky, try again.”)

ELSE:

TheGame()

IF Coordinate(3,1) == 0:

Coordinate(3,1) = 1

DOES 4 in a row == True

IF YES:

TotalWin = TotalWin + 1

WinStreak = WinStreak + 1

ELSE:

Coordinate(3,1) = 2

DOES 4 in a row == True

IF YES:

PRINT (“Unlucky, try again.”)

ELSE:

TheGame()

IF Coordinate(3,2) == 0:

Coordinate(3,2) = 1

DOES 4 in a row == True

IF YES:

TotalWin = TotalWin + 1

WinStreak = WinStreak + 1

ELSE:

Coordinate(3,2) = 2

DOES 4 in a row == True

IF YES:

PRINT (“Unlucky, try again.”)

ELSE:

TheGame()

IF Coordinate(3,3) == 0:

Coordinate(3,3) = 1

DOES 4 in a row == True

IF YES:

TotalWin = TotalWin + 1

WinStreak = WinStreak + 1

ELSE:

Coordinate(3,3) = 2

DOES 4 in a row == True

IF YES:

PRINT (“Unlucky, try again.”)

ELSE:

TheGame()

IF Coordinate(3,4) == 0:

Coordinate(3,4) = 1

DOES 4 in a row == True

IF YES:

TotalWin = TotalWin + 1

WinStreak = WinStreak + 1

ELSE:

Coordinate(3,4) = 2

DOES 4 in a row == True

IF YES:

PRINT (“Unlucky, try again.”)

ELSE:

TheGame()

IF Coordinate(3,5) == 0:

Coordinate(3,5) = 1

DOES 4 in a row == True

IF YES:

TotalWin = TotalWin + 1

WinStreak = WinStreak + 1

ELSE:

Coordinate(3,5) = 2

DOES 4 in a row == True

IF YES:

PRINT (“Unlucky, try again.”)

ELSE:

TheGame()

IF Coordinate(3,6) == 0:

Coordinate(3,6) = 1

DOES 4 in a row == True

IF YES:

TotalWin = TotalWin + 1

WinStreak = WinStreak + 1

ELSE:

Coordinate(3,6) = 2

DOES 4 in a row == True

IF YES:

PRINT (“Unlucky, try again.”)

ELSE:

TheGame()

IF Coordinate(4,0) == 0:

Coordinate(4,0) = 1

DOES 4 in a row == True

IF YES:

TotalWin = TotalWin + 1

WinStreak = WinStreak + 1

ELSE:

Coordinate(4,0) = 2

DOES 4 in a row == True

IF YES:

PRINT (“Unlucky, try again.”)

ELSE:

TheGame()

IF Coordinate(4,1) == 0:

Coordinate(4,1) = 1

DOES 4 in a row == True

IF YES:

TotalWin = TotalWin + 1

WinStreak = WinStreak + 1

ELSE:

Coordinate(4,1) = 2

DOES 4 in a row == True

IF YES:

PRINT (“Unlucky, try again.”)

ELSE:

TheGame()

IF Coordinate(4,2) == 0:

Coordinate(4,2) = 1

DOES 4 in a row == True

IF YES:

TotalWin = TotalWin + 1

WinStreak = WinStreak + 1

ELSE:

Coordinate(4,2) = 2

DOES 4 in a row == True

IF YES:

PRINT (“Unlucky, try again.”)

ELSE:

TheGame()

IF Coordinate(4,3) == 0:

Coordinate(4,3) = 1

DOES 4 in a row == True

IF YES:

TotalWin = TotalWin + 1

WinStreak = WinStreak + 1

ELSE:

Coordinate(4,3) = 2

DOES 4 in a row == True

IF YES:

PRINT (“Unlucky, try again.”)

ELSE:

TheGame()

IF Coordinate(4,4) == 0:

Coordinate(4,4) = 1

DOES 4 in a row == True

IF YES:

TotalWin = TotalWin + 1

WinStreak = WinStreak + 1

ELSE:

Coordinate(4,4) = 2

DOES 4 in a row == True

IF YES:

PRINT (“Unlucky, try again.”)

ELSE:

TheGame()

IF Coordinate(4,5) == 0:

Coordinate(4,5) = 1

DOES 4 in a row == True

IF YES:

TotalWin = TotalWin + 1

WinStreak = WinStreak + 1

ELSE:

Coordinate(4,5) = 2

DOES 4 in a row == True

IF YES:

PRINT (“Unlucky, try again.”)

ELSE:

TheGame()

IF Coordinate(4,6) == 0:

Coordinate(4,6) = 1

DOES 4 in a row == True

IF YES:

TotalWin = TotalWin + 1

WinStreak = WinStreak + 1

ELSE:

Coordinate(4,6) = 2

DOES 4 in a row == True

IF YES:

PRINT (“Unlucky, try again.”)

ELSE:

TheGame()

IF Coordinate(5,0) == 0:

Coordinate(5,0) = 1

DOES 4 in a row == True

IF YES:

TotalWin = TotalWin + 1

WinStreak = WinStreak + 1

ELSE:

Coordinate(5,0) = 2

DOES 4 in a row == True

IF YES:

PRINT (“Unlucky, try again.”)

ELSE:

TheGame()

IF Coordinate(5,1) == 0:

Coordinate(5,1) = 1

DOES 4 in a row == True

IF YES:

TotalWin = TotalWin + 1

WinStreak = WinStreak + 1

ELSE:

Coordinate(5,1) = 2

DOES 4 in a row == True

IF YES:

PRINT (“Unlucky, try again.”)

ELSE:

TheGame()

IF Coordinate(5,2) == 0:

Coordinate(5,2) = 1

DOES 4 in a row == True

IF YES:

TotalWin = TotalWin + 1

WinStreak = WinStreak + 1

ELSE:

Coordinate(5,2) = 2

DOES 4 in a row == True

IF YES:

PRINT (“Unlucky, try again.”)

ELSE:

TheGame()

IF Coordinate(5,3) == 0:

Coordinate(5,3) = 1

DOES 4 in a row == True

IF YES:

TotalWin = TotalWin + 1

WinStreak = WinStreak + 1

ELSE:

Coordinate(5,3) = 2

DOES 4 in a row == True

IF YES:

PRINT (“Unlucky, try again.”)

ELSE:

TheGame()

IF Coordinate(5,4) == 0:

Coordinate(5,4) = 1

DOES 4 in a row == True

IF YES:

TotalWin = TotalWin + 1

WinStreak = WinStreak + 1

ELSE:

Coordinate(5,4) = 2

DOES 4 in a row == True

IF YES:

PRINT (“Unlucky, try again.”)

ELSE:

TheGame()

IF Coordinate(5,5) == 0:

Coordinate(5,5) = 1

DOES 4 in a row == True

IF YES:

TotalWin = TotalWin + 1

WinStreak = WinStreak + 1

ELSE:

Coordinate(5,5) = 2

DOES 4 in a row == True

IF YES:

PRINT (“Unlucky, try again.”)

ELSE:

TheGame()

IF Coordinate(5,6) == 0:

Coordinate(5,6) = 1

DOES 4 in a row == True

IF YES:

TotalWin = TotalWin + 1

WinStreak = WinStreak + 1

ELSE:

Coordinate(5,6) = 2

DOES 4 in a row == True

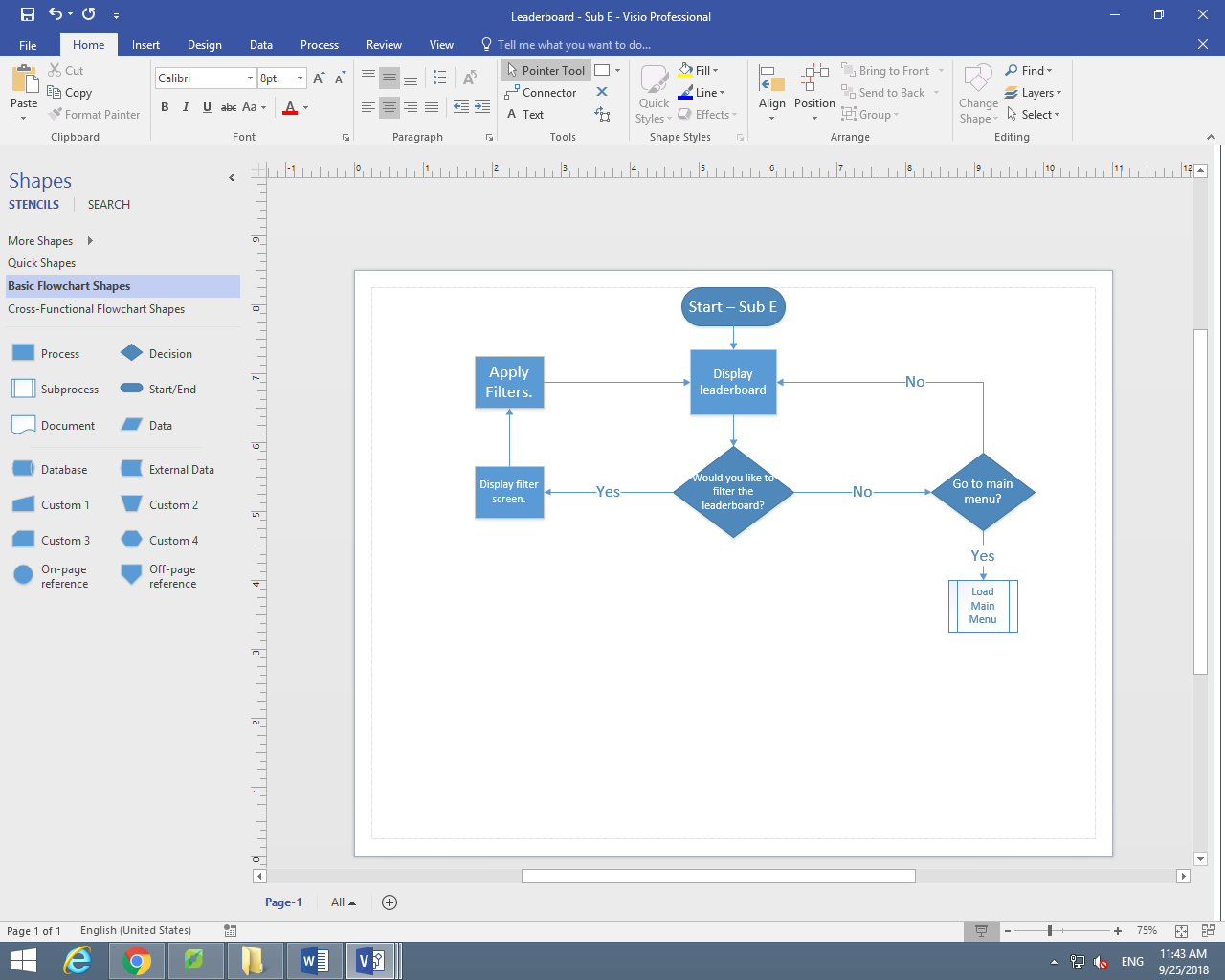
IF YES:

PRINT (“Unlucky, try again.”)

ELSE:

TheGame()

## Leaderboard:



Leaderboard()

LOAD Leaderboard

IF USER clicks Filter button:

LOAD Filter tab

IF USER clicks DisplayTotalWins:

DisplayTotalWins == True

Leaderboard()

#(USER clicks DisplayWinStreak)

ELSE:

DisplayWinStreak == True

Leaderboard()

#(USER doesn’t click Filter button)

ELSE:

DISPLAY Leaderboard

IF USER clicks Main Menu:

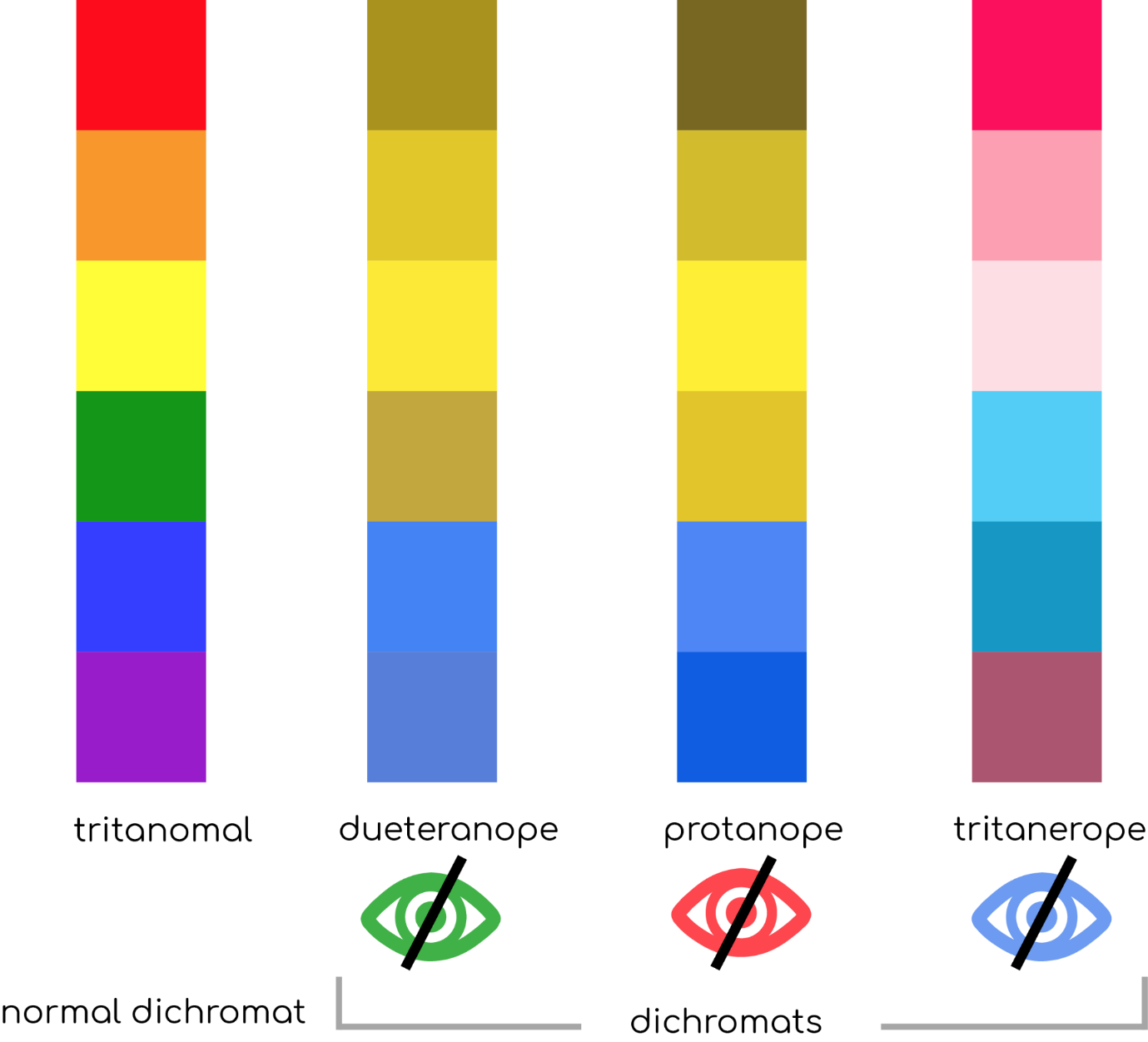
MainMenu()

ELSE:

Leaderboard()

# 3.2.2 (C)

One of the main usability features that need to be accounted for is the fact that there is a potential for students who play the game may be colour-blind. As you can see from the chart below, people have different types of colour-blindness. If your eye sight is tritanomal vision then you have normal vision but if you have any other form of vision then that can impact your sight to see shades of green, red or blue and also impact how those shades blend with other colours. This therefore means that I need to include a setting that allows players to change their token colour as well as the background colour to suit their needs. This means that more people can easily play the game, which can then support their learning.



# 3.2.2 (D)

## Data Dictionary

|  |  |  |  |
| --- | --- | --- | --- |
| Name of Field | Data Type | Description | Validation |
| Username | String | The purpose of this field is to store the different usernames that the students want to assign to themselves. It is a string because it will not just contain a single character but instead will include multiple different characters. | Validation will be used here as the username must contain at least one number, one capital letter and one special character and finally cannot be longer than 12 characters. |
| Password | String | The purpose of this field is to store the password that links with each of the usernames. It is a string because it will not just contain a single character but instead will include multiple different characters. | Validation will be used here as the username must contain at least one number, one capital letter and one special character and finally cannot be longer than 12 characters. |
| Total Win | Integer | The purpose of this field is track the number of wins each user has and will increment by one each time the user wins a game. It is an integer because it will need to track a number and therefore if it is stored as anything else then it will not recognise the number. | This is just an incrementing number and therefore no validation is needed. |

## Variable Table:

|  |  |  |
| --- | --- | --- |
| Variable Name | Type of Variable | Description |
| LoginorRegister |  |  |
| Username | String | This variable will be used to ask the user what their username is for the login subroutine. This is the same for the register subroutine but what will be done is once it has asked for it will be different. When the username is entered after being asked for, it will be added to the database once it has been appended. It will also be used later on if the user wins a game, as it will be used to increment their total wins that are assigned to them. |
| Password | String | This variable will be used to ask the user what their password is for the login subroutine. If this matches the username they have used, they will be logged into the game and ready to play. If not, an error message will be displayed and then they must try again. This is the same for the register subroutine but what will be done is once it has asked for it will be different. When the password is entered after being asked for, it will be added to the database once it has been appended. This will then be assigned to the username that was entered and that will be the user’s login. |
| Play Menu Selected | Boolean | This variable will be used when the user selects to play the game. It is a Boolean because it will only be on or off at a given time (true or false). |
| Leaderboard Selected | Boolean | This variable will be used when the user selects to view the leaderboard. It is a Boolean because it will only be on or off at a given time (true or false). |
| DisplayTotalWin | Boolean | This variable will be used when the user asks to filter the leaderboard to show where every user is based on their total number of wins. It is a Boolean because it will only be given when the users selects the filter on the leaderboard to be like this; if it is on or off (True or False). |
| DisplayWinStreak | Boolean | This variable will be used when the user asks to filter the leaderboard to show where every user is based on their current winstreak. It is a Boolean because it will only be given when the users selects the filter on the leaderboard to be like this; if it is on or off (True or False). |
| Exit Selected | Boolean | This variable will be used when the user selects to exit the game. It is a Boolean because it will only be on or off at a given time (true or false). |
| Level Selected | Boolean | This variable will be used when the user selects to choose what level they want to play the game on. It is a Boolean because it will only be on or off at a given time (true or false). |
| Settings Selected | Boolean | This variable will be used when the user selects to change the settings (the colour of the tokens or the background colour. It is a Boolean because it will only be on or off at a given time (true or false). |
| Background Red Apply | Boolean | This variable will be used when the user selects to change the background colour to be red. It is a Boolean because it will only be on or off at a given time (true or false). |
| Background Blue Apply | Boolean | This variable will be used when the user selects to change the background colour to be blue. It is a Boolean because it will only be on or off at a given time (true or false). |
| Background Black Apply | Boolean | This variable will be used when the user selects to change the background colour to be black. It is a Boolean because it will only be on or off at a given time (true or false). |
| Background Orange Apply | Boolean | This variable will be used when the user selects to change the background colour to be orange. It is a Boolean because it will only be on or off at a given time (true or false). |
| Background Purple Apply | Boolean | This variable will be used when the user selects to change the background colour to be purple. It is a Boolean because it will only be on or off at a given time (true or false). |
| Background Brown Apply | Boolean | This variable will be used when the user selects to change the background colour to be brown. It is a Boolean because it will only be on or off at a given time (true or false). |
| Background Yellow Apply | Boolean | This variable will be used when the user selects to change the background colour to be yellow. It is a Boolean because it will only be on or off at a given time (true or false). |
| Background Green Apply | Boolean | This variable will be used when the user selects to change the background colour to be green. It is a Boolean because it will only be on or off at a given time (true or false). |
| Background Grey Apply | Boolean | This variable will be used when the user selects to change the background colour to be grey. It is a Boolean because it will only be on or off at a given time (true or false). |
| Background Pink Apply | Boolean | This variable will be used when the user selects to change the background colour to be pink. It is a Boolean because it will only be on or off at a given time (true or false). |
| User Token Red Apply | Boolean | This variable will be used when the user selects to change the user’s token colour to be red. It is a Boolean because it will only be on or off at a given time (true or false). |
| User Token Blue Apply | Boolean | This variable will be used when the user selects to change the user’s token colour to be blue. It is a Boolean because it will only be on or off at a given time (true or false). |
| User Token Black Apply | Boolean | This variable will be used when the user selects to change the user’s token colour to be black. It is a Boolean because it will only be on or off at a given time (true or false). |
| User Token Orange Apply | Boolean | This variable will be used when the user selects to change the user’s token colour to be orange. It is a Boolean because it will only be on or off at a given time (true or false). |
| User Token Purple Apply | Boolean | This variable will be used when the user selects to change the user’s token colour to be purple. It is a Boolean because it will only be on or off at a given time (true or false). |
| User Token Brown Apply | Boolean | This variable will be used when the user selects to change the background colour to be brown. It is a Boolean because it will only be on or off at a given time (true or false). |
| User Token Yellow Apply | Boolean | This variable will be used when the user selects to change the user’s token colour to be yellow. It is a Boolean because it will only be on or off at a given time (true or false). |
| User Token Green Apply | Boolean | This variable will be used when the user selects to change the user’s token colour to be green. It is a Boolean because it will only be on or off at a given time (true or false). |
| User Token Grey Apply | Boolean | This variable will be used when the user selects to change the user’s token colour to be grey. It is a Boolean because it will only be on or off at a given time (true or false). |
| User Token Pink Apply | Boolean | This variable will be used when the user selects to change the user’s token colour to be pink. It is a Boolean because it will only be on or off at a given time (true or false). |
| AI Token Red Apply | Boolean | This variable will be used when the user selects to change the AI’s token colour to be red. It is a Boolean because it will only be on or off at a given time (true or false). |
| AI Token Blue Apply | Boolean | This variable will be used when the user selects to change the AI’s token colour to be blue. It is a Boolean because it will only be on or off at a given time (true or false). |
| AI Token Black Apply | Boolean | This variable will be used when the user selects to change the AI’s token colour to be black. It is a Boolean because it will only be on or off at a given time (true or false). |
| AI Token Orange Apply | Boolean | This variable will be used when the user selects to change the AI’s token colour to be orange. It is a Boolean because it will only be on or off at a given time (true or false). |
| AI Token Purple Apply | Boolean | This variable will be used when the user selects to change the AI’s token colour to be purple. It is a Boolean because it will only be on or off at a given time (true or false). |
| AI Token Brown Apply | Boolean | This variable will be used when the user selects to change the AI’s token colour to be brown. It is a Boolean because it will only be on or off at a given time (true or false). |
| AI Token Yellow Apply | Boolean | This variable will be used when the user selects to change the AI’s token colour to be yellow. It is a Boolean because it will only be on or off at a given time (true or false). |
| AI Token Green Apply | Boolean | This variable will be used when the user selects to change the AI’s token colour to be green. It is a Boolean because it will only be on or off at a given time (true or false). |
| AI Token Grey Apply | Boolean | This variable will be used when the user selects to change the AI’s token colour to be grey. It is a Boolean because it will only be on or off at a given time (true or false). |
| AI Token Pink Apply | Boolean | This variable will be used when the user selects to change the AI’s token colour to be pink. It is a Boolean because it will only be on or off at a given time (true or false). |
| Easy Selected | Boolean | This variable will be used when the user selects to play the game on easy difficulty. It is a Boolean because it will only be on or off at a given time (true or false). |
| Medium Selected | Boolean | This variable will be used when the user selects to play the game on medium difficulty. It is a Boolean because it will only be on or off at a given time (true or false). |
| Hard Selected | Boolean | This variable will be used when the user selects to play the game on hard difficulty. It is a Boolean because it will only be on or off at a given time (true or false). |
| Co-ordinate (0,0) | Integer | This variable will be used to determine who owns the co-ordinate. It is an integer because there are three numbers that could be assigned to it. The first is “0” to denote that the space is empty. The second is “1” to denote the user owns that space. Finally, the third is “2” to denote the AI owns that space. |
| Co-ordinate (0,1) | Integer | This variable will be used to determine who owns the co-ordinate. It is an integer because there are three numbers that could be assigned to it. The first is “0” to denote that the space is empty. The second is “1” to denote the user owns that space. Finally, the third is “2” to denote the AI owns that space. |
| Co-ordinate (0,2) | Integer | This variable will be used to determine who owns the co-ordinate. It is an integer because there are three numbers that could be assigned to it. The first is “0” to denote that the space is empty. The second is “1” to denote the user owns that space. Finally, the third is “2” to denote the AI owns that space. |
| Co-ordinate (0,3) | Integer | This variable will be used to determine who owns the co-ordinate. It is an integer because there are three numbers that could be assigned to it. The first is “0” to denote that the space is empty. The second is “1” to denote the user owns that space. Finally, the third is “2” to denote the AI owns that space. |
| Co-ordinate (0,4) | Integer | This variable will be used to determine who owns the co-ordinate. It is an integer because there are three numbers that could be assigned to it. The first is “0” to denote that the space is empty. The second is “1” to denote the user owns that space. Finally, the third is “2” to denote the AI owns that space. |
| Co-ordinate (0,5) | Integer | This variable will be used to determine who owns the co-ordinate. It is an integer because there are three numbers that could be assigned to it. The first is “0” to denote that the space is empty. The second is “1” to denote the user owns that space. Finally, the third is “2” to denote the AI owns that space. |
| Co-ordinate (0,6) | Integer | This variable will be used to determine who owns the co-ordinate. It is an integer because there are three numbers that could be assigned to it. The first is “0” to denote that the space is empty. The second is “1” to denote the user owns that space. Finally, the third is “2” to denote the AI owns that space. |
| Co-ordinate (1,0) | Integer | This variable will be used to determine who owns the co-ordinate. It is an integer because there are three numbers that could be assigned to it. The first is “0” to denote that the space is empty. The second is “1” to denote the user owns that space. Finally, the third is “2” to denote the AI owns that space. |
| Co-ordinate (1,1) | Integer | This variable will be used to determine who owns the co-ordinate. It is an integer because there are three numbers that could be assigned to it. The first is “0” to denote that the space is empty. The second is “1” to denote the user owns that space. Finally, the third is “2” to denote the AI owns that space. |
| Co-ordinate (1,2) | Integer | This variable will be used to determine who owns the co-ordinate. It is an integer because there are three numbers that could be assigned to it. The first is “0” to denote that the space is empty. The second is “1” to denote the user owns that space. Finally, the third is “2” to denote the AI owns that space. |
| Co-ordinate (1,3) | Integer | This variable will be used to determine who owns the co-ordinate. It is an integer because there are three numbers that could be assigned to it. The first is “0” to denote that the space is empty. The second is “1” to denote the user owns that space. Finally, the third is “2” to denote the AI owns that space. |
| Co-ordinate (1,4) | Integer | This variable will be used to determine who owns the co-ordinate. It is an integer because there are three numbers that could be assigned to it. The first is “0” to denote that the space is empty. The second is “1” to denote the user owns that space. Finally, the third is “2” to denote the AI owns that space. |
| Co-ordinate (1,5) | Integer | This variable will be used to determine who owns the co-ordinate. It is an integer because there are three numbers that could be assigned to it. The first is “0” to denote that the space is empty. The second is “1” to denote the user owns that space. Finally, the third is “2” to denote the AI owns that space. |
| Co-ordinate (1,6) | Integer | This variable will be used to determine who owns the co-ordinate. It is an integer because there are three numbers that could be assigned to it. The first is “0” to denote that the space is empty. The second is “1” to denote the user owns that space. Finally, the third is “2” to denote the AI owns that space. |
| Co-ordinate (2,0) | Integer | This variable will be used to determine who owns the co-ordinate. It is an integer because there are three numbers that could be assigned to it. The first is “0” to denote that the space is empty. The second is “1” to denote the user owns that space. Finally, the third is “2” to denote the AI owns that space. |
| Co-ordinate (2,1) | Integer | This variable will be used to determine who owns the co-ordinate. It is an integer because there are three numbers that could be assigned to it. The first is “0” to denote that the space is empty. The second is “1” to denote the user owns that space. Finally, the third is “2” to denote the AI owns that space. |
| Co-ordinate (2,2) | Integer | This variable will be used to determine who owns the co-ordinate. It is an integer because there are three numbers that could be assigned to it. The first is “0” to denote that the space is empty. The second is “1” to denote the user owns that space. Finally, the third is “2” to denote the AI owns that space. |
| Co-ordinate (2,3) | Integer | This variable will be used to determine who owns the co-ordinate. It is an integer because there are three numbers that could be assigned to it. The first is “0” to denote that the space is empty. The second is “1” to denote the user owns that space. Finally, the third is “2” to denote the AI owns that space. |
| Co-ordinate (2,4) | Integer | This variable will be used to determine who owns the co-ordinate. It is an integer because there are three numbers that could be assigned to it. The first is “0” to denote that the space is empty. The second is “1” to denote the user owns that space. Finally, the third is “2” to denote the AI owns that space. |
| Co-ordinate (2,5) | Integer | This variable will be used to determine who owns the co-ordinate. It is an integer because there are three numbers that could be assigned to it. The first is “0” to denote that the space is empty. The second is “1” to denote the user owns that space. Finally, the third is “2” to denote the AI owns that space. |
| Co-ordinate (2,6) | Integer | This variable will be used to determine who owns the co-ordinate. It is an integer because there are three numbers that could be assigned to it. The first is “0” to denote that the space is empty. The second is “1” to denote the user owns that space. Finally, the third is “2” to denote the AI owns that space. |
| Co-ordinate (3,0) | Integer | This variable will be used to determine who owns the co-ordinate. It is an integer because there are three numbers that could be assigned to it. The first is “0” to denote that the space is empty. The second is “1” to denote the user owns that space. Finally, the third is “2” to denote the AI owns that space. |
| Co-ordinate (3,1) | Integer | This variable will be used to determine who owns the co-ordinate. It is an integer because there are three numbers that could be assigned to it. The first is “0” to denote that the space is empty. The second is “1” to denote the user owns that space. Finally, the third is “2” to denote the AI owns that space. |
| Co-ordinate (3,2) | Integer | This variable will be used to determine who owns the co-ordinate. It is an integer because there are three numbers that could be assigned to it. The first is “0” to denote that the space is empty. The second is “1” to denote the user owns that space. Finally, the third is “2” to denote the AI owns that space. |
| Co-ordinate (3,3) | Integer | This variable will be used to determine who owns the co-ordinate. It is an integer because there are three numbers that could be assigned to it. The first is “0” to denote that the space is empty. The second is “1” to denote the user owns that space. Finally, the third is “2” to denote the AI owns that space. |
| Co-ordinate (3,4) | Integer | This variable will be used to determine who owns the co-ordinate. It is an integer because there are three numbers that could be assigned to it. The first is “0” to denote that the space is empty. The second is “1” to denote the user owns that space. Finally, the third is “2” to denote the AI owns that space. |
| Co-ordinate (3,5) | Integer | This variable will be used to determine who owns the co-ordinate. It is an integer because there are three numbers that could be assigned to it. The first is “0” to denote that the space is empty. The second is “1” to denote the user owns that space. Finally, the third is “2” to denote the AI owns that space. |
| Co-ordinate (3,6) | Integer | This variable will be used to determine who owns the co-ordinate. It is an integer because there are three numbers that could be assigned to it. The first is “0” to denote that the space is empty. The second is “1” to denote the user owns that space. Finally, the third is “2” to denote the AI owns that space. |
| Co-ordinate (4,0) | Integer | This variable will be used to determine who owns the co-ordinate. It is an integer because there are three numbers that could be assigned to it. The first is “0” to denote that the space is empty. The second is “1” to denote the user owns that space. Finally, the third is “2” to denote the AI owns that space. |
| Co-ordinate (4,1) | Integer | This variable will be used to determine who owns the co-ordinate. It is an integer because there are three numbers that could be assigned to it. The first is “0” to denote that the space is empty. The second is “1” to denote the user owns that space. Finally, the third is “2” to denote the AI owns that space. |
| Co-ordinate (4,2) | Integer | This variable will be used to determine who owns the co-ordinate. It is an integer because there are three numbers that could be assigned to it. The first is “0” to denote that the space is empty. The second is “1” to denote the user owns that space. Finally, the third is “2” to denote the AI owns that space. |
| Co-ordinate (4,3) | Integer | This variable will be used to determine who owns the co-ordinate. It is an integer because there are three numbers that could be assigned to it. The first is “0” to denote that the space is empty. The second is “1” to denote the user owns that space. Finally, the third is “2” to denote the AI owns that space. |
| Co-ordinate (4,4) | Integer | This variable will be used to determine who owns the co-ordinate. It is an integer because there are three numbers that could be assigned to it. The first is “0” to denote that the space is empty. The second is “1” to denote the user owns that space. Finally, the third is “2” to denote the AI owns that space. |
| Co-ordinate (4,4) | Integer | This variable will be used to determine who owns the co-ordinate. It is an integer because there are three numbers that could be assigned to it. The first is “0” to denote that the space is empty. The second is “1” to denote the user owns that space. Finally, the third is “2” to denote the AI owns that space. |
| Co-ordinate (4,6) | Integer | This variable will be used to determine who owns the co-ordinate. It is an integer because there are three numbers that could be assigned to it. The first is “0” to denote that the space is empty. The second is “1” to denote the user owns that space. Finally, the third is “2” to denote the AI owns that space. |
| Co-ordinate (5,0) | Integer | This variable will be used to determine who owns the co-ordinate. It is an integer because there are three numbers that could be assigned to it. The first is “0” to denote that the space is empty. The second is “1” to denote the user owns that space. Finally, the third is “2” to denote the AI owns that space. |
| Co-ordinate (5,1) | Integer | This variable will be used to determine who owns the co-ordinate. It is an integer because there are three numbers that could be assigned to it. The first is “0” to denote that the space is empty. The second is “1” to denote the user owns that space. Finally, the third is “2” to denote the AI owns that space. |
| Co-ordinate (5,2) | Integer | This variable will be used to determine who owns the co-ordinate. It is an integer because there are three numbers that could be assigned to it. The first is “0” to denote that the space is empty. The second is “1” to denote the user owns that space. Finally, the third is “2” to denote the AI owns that space. |
| Co-ordinate (5,3) | Integer | This variable will be used to determine who owns the co-ordinate. It is an integer because there are three numbers that could be assigned to it. The first is “0” to denote that the space is empty. The second is “1” to denote the user owns that space. Finally, the third is “2” to denote the AI owns that space. |
| Co-ordinate (5,4) | Integer | This variable will be used to determine who owns the co-ordinate. It is an integer because there are three numbers that could be assigned to it. The first is “0” to denote that the space is empty. The second is “1” to denote the user owns that space. Finally, the third is “2” to denote the AI owns that space. |
| Co-ordinate (5,5) | Integer | This variable will be used to determine who owns the co-ordinate. It is an integer because there are three numbers that could be assigned to it. The first is “0” to denote that the space is empty. The second is “1” to denote the user owns that space. Finally, the third is “2” to denote the AI owns that space. |
| Co-ordinate (5,6) | Integer | This variable will be used to determine who owns the co-ordinate. It is an integer because there are three numbers that could be assigned to it. The first is “0” to denote that the space is empty. The second is “1” to denote the user owns that space. Finally, the third is “2” to denote the AI owns that space. |

## Database Diagram

Password

Username

Total Wins

# 3.2.3 Test Plan Tables

Iterative Development Test Table (Testing whether something works during the creation of the program):

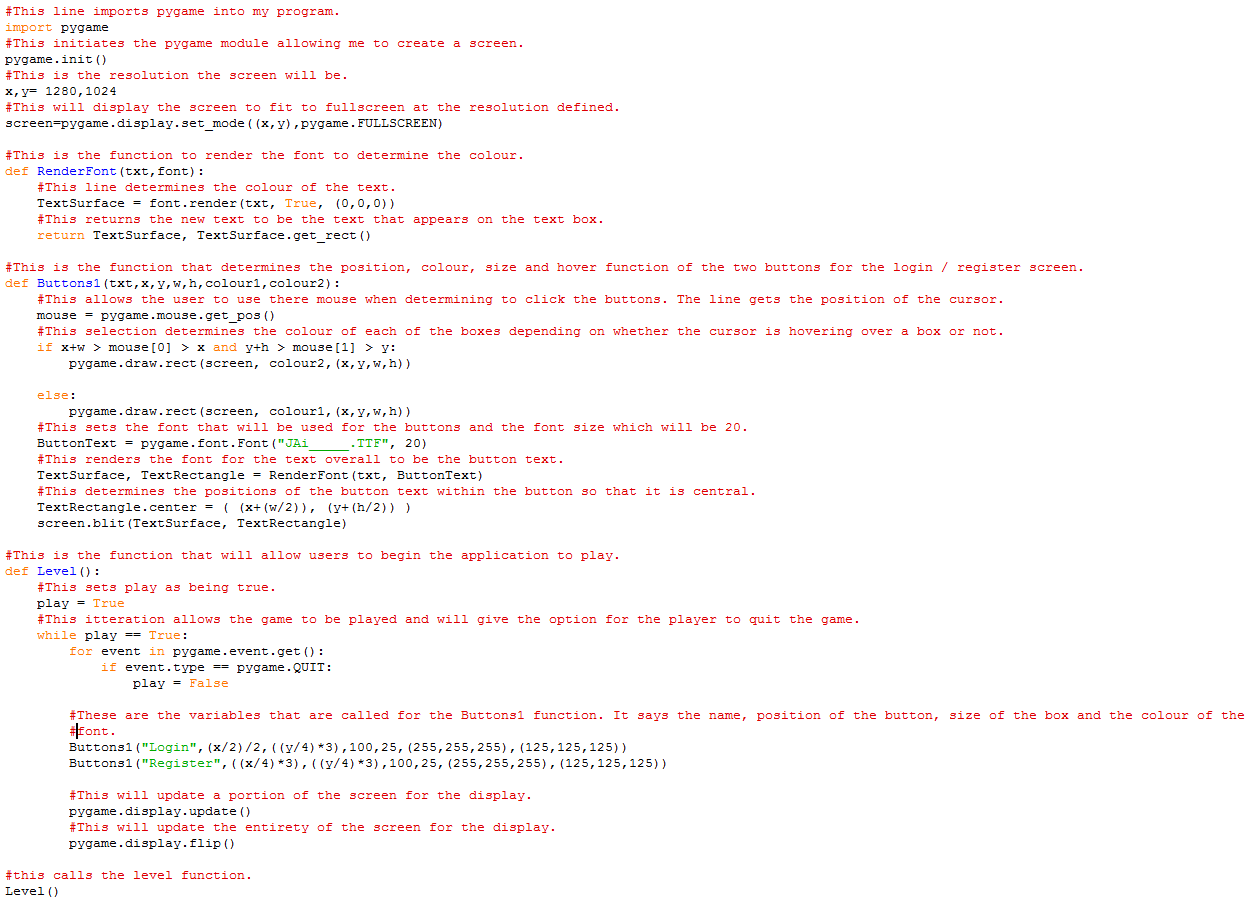
|  |  |  |  |
| --- | --- | --- | --- |
| Date | What am I testing? | How am I testing it? | Result |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Post Development Test Table (Tests to try to break the code):

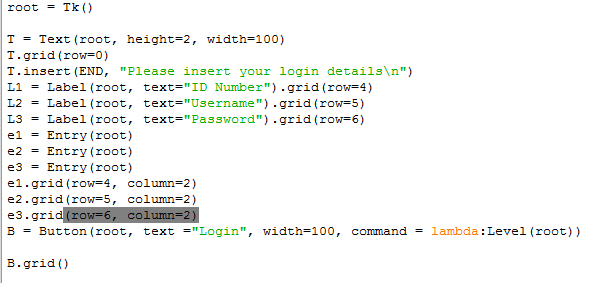
|  |  |  |  |
| --- | --- | --- | --- |
| Test Number | What is being tested? | Expected Outcome | What actually happened? |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# 3.3.1

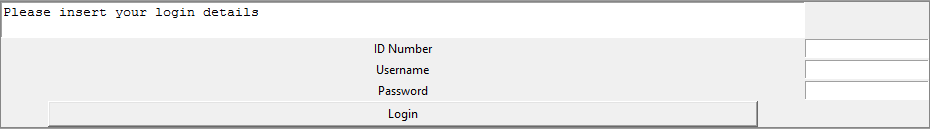
## Development Diary



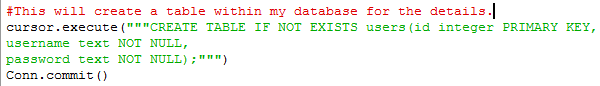
This is my first entry into my development diary. At this point, I can make a screen with two buttons that say “Register” and “Login”. The colour of the button will change when you hover over them so the user knows they could click it.



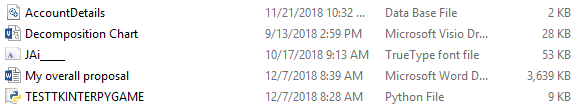
Above is the code I have used in order to create my login screen which is used to store the information of the user. It contains a simple text box with two input boxes and a button. You can see this below.

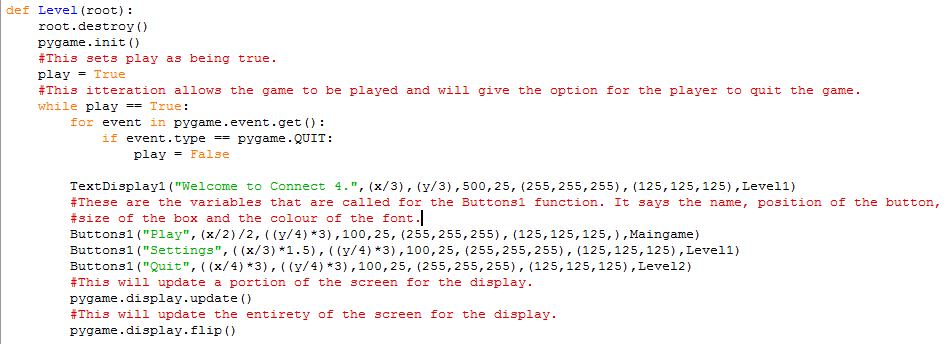






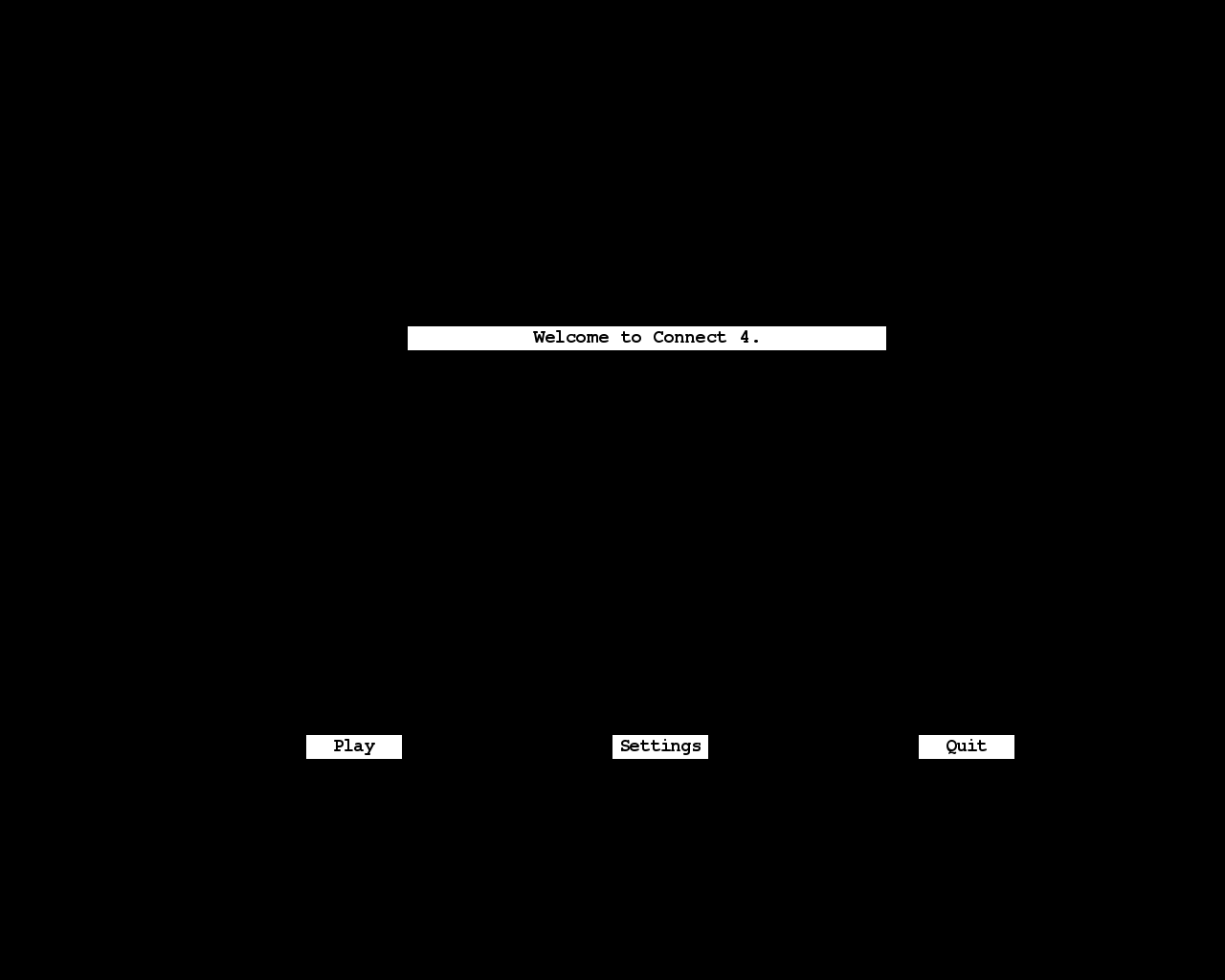
The next stage was to create my database. The code above shows I have used SQL in order to create a database called “AccountDetails”. This database will, as you can see, include a table with the fields “integer, username and password.” You can see below that the database has been created as a file in my folder as a Data Base file.



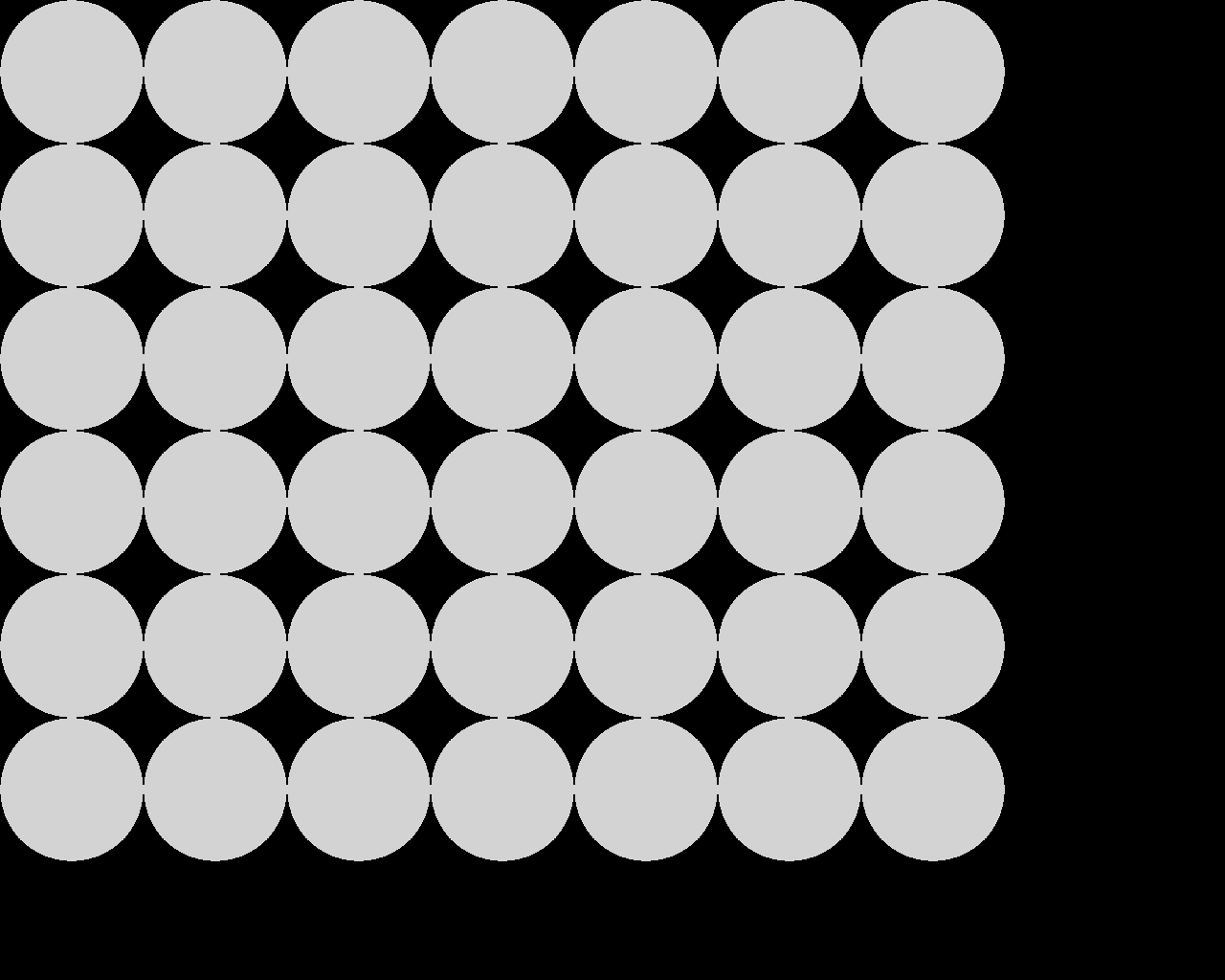


Above is the code I have used in order to create the main menu screen for the game. It has a TextDisplay line that displays “Welcome to Connect 4.” It also has three buttons that you can click. These are a “play” button, a “settings” button and a “quit” button. If the “Play” button is pressed then the game will load. If the “Settings” button is pressed then this will lead to a settings menu where the user will be able to change the colour of the background and the token colours. Finally, when the “Quit” button is pressed, the game will be closed. You can see my menu below as well as what happens after you click each button.

Main Menu:



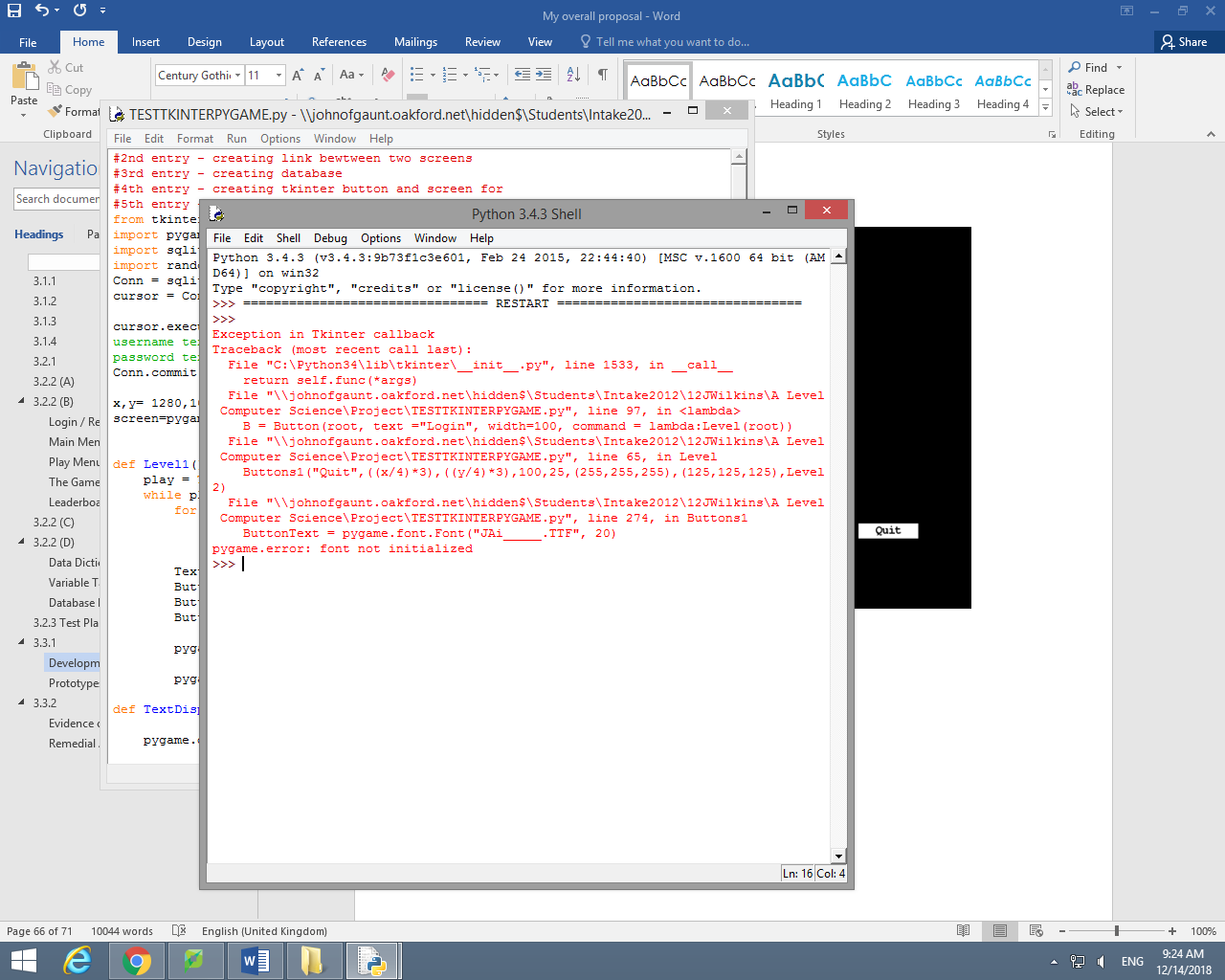
When the “Play” button is pressed:



When the “Settings” button is pressed:

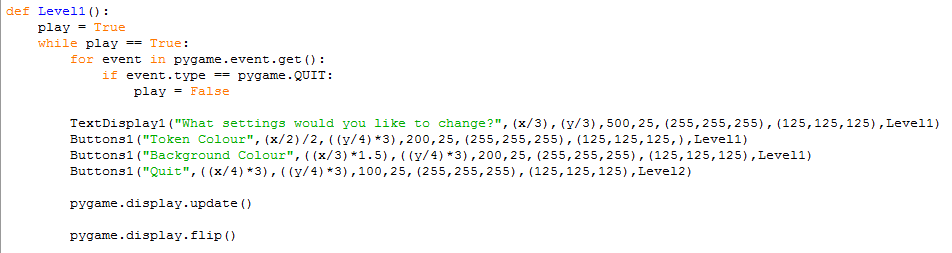


When the “Quit” button is pressed:



It is okay to have this error because the program is still trying to initialize the pygame material even though I have closed down the pygame section of the program. Therefore, this error doesn’t stop the program from closing down which means that the quit function is working as it should be.

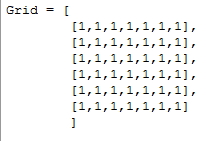
As you can see from above, when the “Settings” button is clicked, it will lead to a settings menu where you will be able to change the token colours as well as the background colour. Below is the code I used in order to create this menu. It is a very similar principle to the Main Menu as it consists of TextDisplay line which reads “What settings would you like to change?” as well as three buttons, one for “Token colour”, one for “Background colour” and one for “quit”, which will allow the user to quit the game.

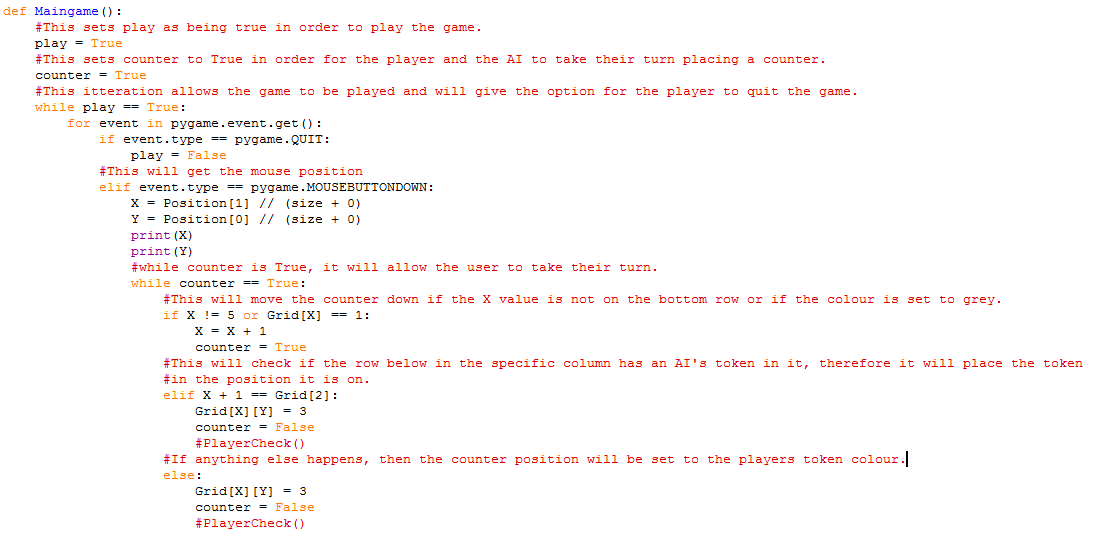


The code for the quit function is very simple as all that is needed when the function is called is pygame.quit(). This will make the pygame section of the program close.

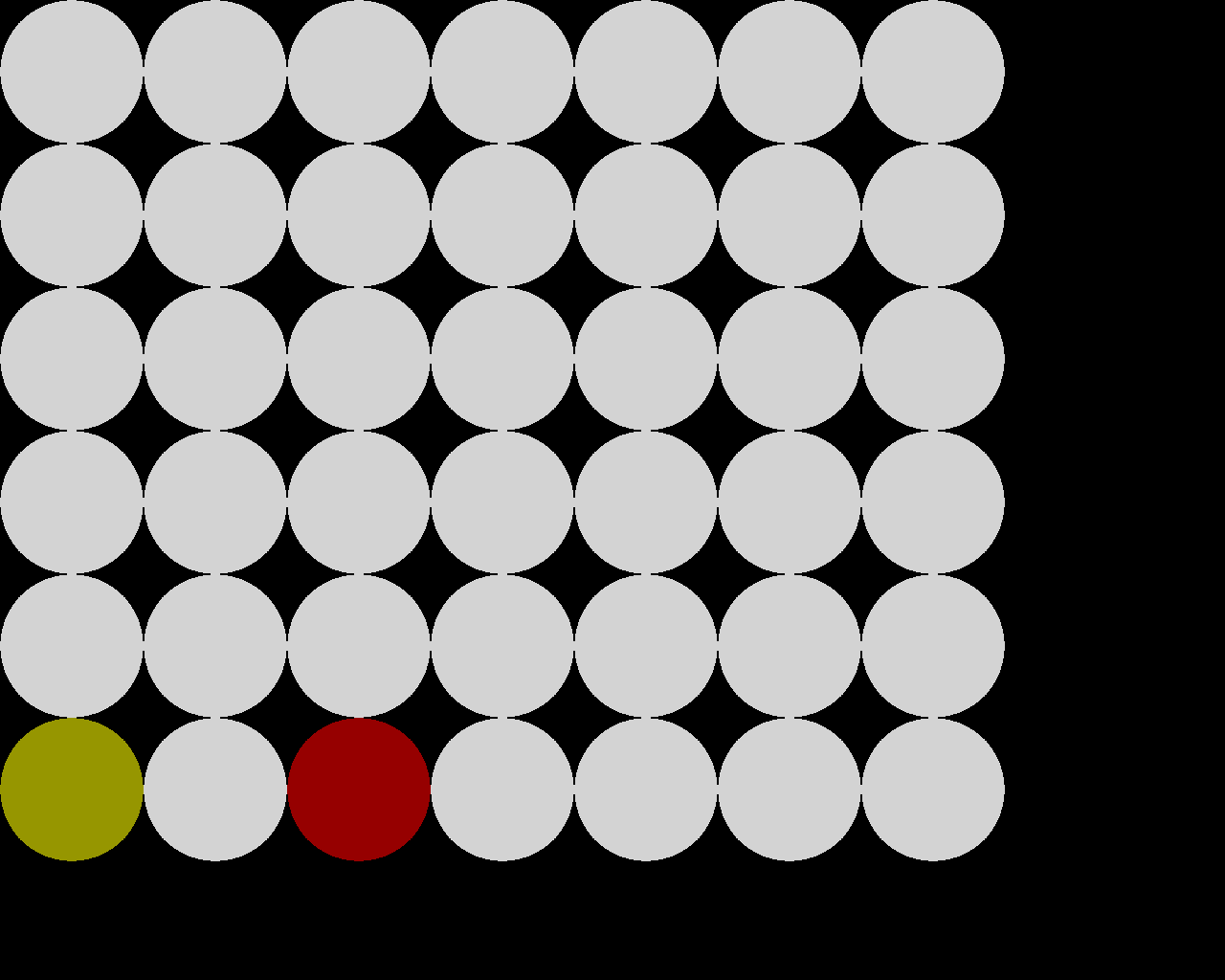


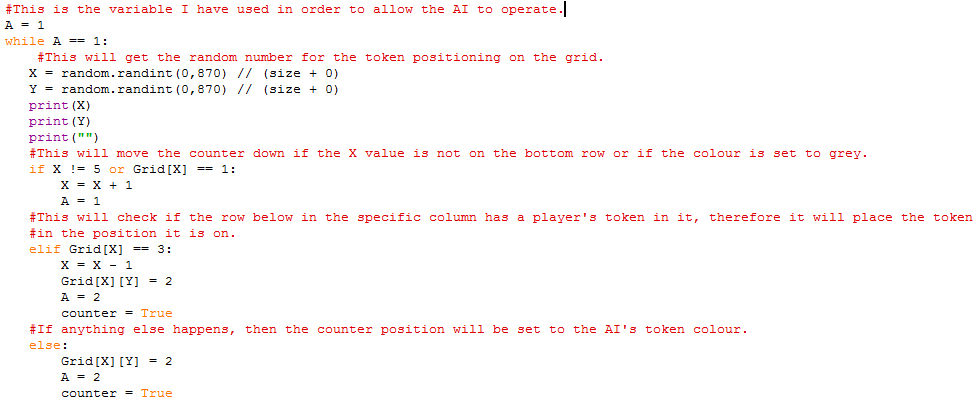
Above, you could see my grid for the connect 4 game. Below is the code I used in order to create the grid.



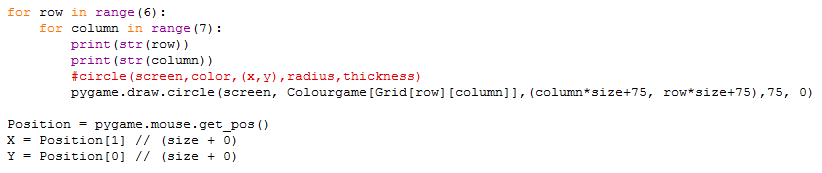


Above is the code I have used in order to allow the player to take their move. First, it will get the mouse position in order to allow the user to click to place the token. Then it will determine the correct variable when the user clicks on a column. The first scenario allows the token to fall down the rows in order to allow it to hit either the bottom row or the lowest row without hitting another token. The second scenario should place a token at the position it is checking if there is an AI token on the row below it in that column. Finally, the third scenario will place a token if there are any situations that have been unaccounted for. You can see below that a token is placed when I run my program.

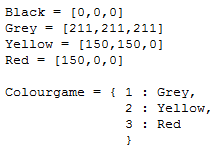




As you can see from above, when the player places their token, the AI will place theirs automatically. The code below explains how the AI will place their token. The AI will take place as long as A = 1. It will then determine a random position for the X and Y axis for its placement. Then it will determine the correct variable when the AI selects a column. The first scenario allows the token to fall down the rows in order to allow it to hit either the bottom row or the lowest row without hitting another token. The second scenario should place a token at the position it is checking if there is an AI token on the row below it in that column. Finally, the third scenario will place a token if there are any situations that have been unaccounted for. You can see below that a token is placed when I run my program.



Above is the code that will draw the tokens onto the grid. It will also draw them in the specific position in which the grid spot was clicked on. It determines the position by using pygame.mouse.get\_pos() and will place the token in the position it finds using that line of code.



Finally, here are the colours I have used throughout the code with their hex values in order to get the colour to appear on screen. I have put some in a dictionary called Colourgame in order to use the numbers throughout my code instead of writing the colours every time to make it easier for me write the program.

Below is my final code in full:

from tkinter import \*

import pygame

import sqlite3

import random

Conn = sqlite3.connect("AccountDetails.db")

cursor = Conn.cursor()

cursor.execute("""CREATE TABLE IF NOT EXISTS users(id integer PRIMARY KEY,

username text NOT NULL,

password text NOT NULL);""")

Conn.commit()

#This determines the size of the screen that will be created in the pygame section of the program.

x,y= 1280,1024

screen=pygame.display.set\_mode((x,y),pygame.FULLSCREEN)

def Level1():

play = True

while play == True:

for event in pygame.event.get():

if event.type == pygame.QUIT:

play = False

TextDisplay1("What settings would you like to change?",(x/3),(y/3),500,25,(255,255,255),(125,125,125),Level1)

Buttons1("Token Colour",(x/2)/2,((y/4)\*3),200,25,(255,255,255),(125,125,125,),Level1)

Buttons1("Background Colour",((x/3)\*1.5),((y/4)\*3),200,25,(255,255,255),(125,125,125),Level1)

Buttons1("Quit",((x/4)\*3),((y/4)\*3),100,25,(255,255,255),(125,125,125),Level2)

pygame.display.update()

pygame.display.flip()

def TextDisplay1(txt,x,y,w,h,colour1,colour2,command):

pygame.draw.rect(screen, colour1, (x,y,w,h))

DisplayText = pygame.font.Font("JAi\_\_\_\_\_.TTF", 20)

TextSurface, TextRectangle = RenderFont(txt, DisplayText)

TextRectangle.center = ( (x+(w/2)), (y+(h/2)) )

screen.blit(TextSurface, TextRectangle)

def Level(root):

root.destroy()

pygame.init()

#This sets play as being true.

play = True

#This itteration allows the game to be played and will give the option for the player to quit the game.

while play == True:

for event in pygame.event.get():

if event.type == pygame.QUIT:

play = False

TextDisplay1("Welcome to Connect 4.",(x/3),(y/3),500,25,(255,255,255),(125,125,125),Level1)

#These are the variables that are called for the Buttons1 function. It says the name, position of the button,

#size of the box and the colour of the font.

Buttons1("Play",(x/2)/2,((y/4)\*3),100,25,(255,255,255),(125,125,125,),Maingame)

Buttons1("Settings",((x/3)\*1.5),((y/4)\*3),100,25,(255,255,255),(125,125,125),Level1)

Buttons1("Quit",((x/4)\*3),((y/4)\*3),100,25,(255,255,255),(125,125,125),Level2)

#This will update a portion of the screen for the display.

pygame.display.update()

#This will update the entirety of the screen for the display.

pygame.display.flip()

#def dbupdate(root):

#global e1

#global e2

#global e3

#newid=input(e1)

#newusername=input(e2)

#newpassword=input(e3)

#cursor.execute("""INSERT INTO users (id,username,password)

#VALUES (?,?,?)""",(newid,newusername,newpassword))

#Conn.commit()

#level(dbupdate)

root = Tk()

T = Text(root, height=2, width=100)

T.grid(row=0)

T.insert(END, "Please insert your login details\n")

L1 = Label(root, text="ID Number").grid(row=4)

L2 = Label(root, text="Username").grid(row=5)

L3 = Label(root, text="Password").grid(row=6)

e1 = Entry(root)

e2 = Entry(root)

e3 = Entry(root)

e1.grid(row=4, column=2)

e2.grid(row=5, column=2)

e3.grid(row=6, column=2)

B = Button(root, text ="Login", width=100, command = lambda:Level(root))

B.grid()

def RenderFont(txt,font):

#This line determines the colour of the text.

TextSurface = font.render(txt, True, (0,0,0))

#This returns the new text to be the text that appears on the text box.

return TextSurface, TextSurface.get\_rect()

#http://programarcadegames.com/index.php?lang=en&chapter=array\_backed\_grids

#These are the colours I'll be using in my game.

Black = [0,0,0]

Grey = [211,211,211]

Yellow = [150,150,0]

Red = [150,0,0]

#Some colours have been stored in this dictionary.

Colourgame = { 1 : Grey,

2 : Yellow,

3 : Red

}

size = 150

Grid = [

[1,1,1,1,1,1,1],

[1,1,1,1,1,1,1],

[1,1,1,1,1,1,1],

[1,1,1,1,1,1,1],

[1,1,1,1,1,1,1],

[1,1,1,1,1,1,1]

]

def TextDisplay2(txt,x,y,w,h,colour1,colour2,command):

pygame.draw.rect(screen, colour1, (x,y,w,h))

DisplayText = pygame.font.Font("JAi\_\_\_\_\_.TTF", 20)

TextSurface, TextRectangle = RenderFont(txt, DisplayText)

TextRectangle.center = ( (x+(w/2)), (y+(h/2)) )

screen.blit(TextSurface, TextRectangle)

#def PlayerCheck():

for X in Grid:

for Y in Grid:

if X == [2,3,3,3,3,1,1]:

print ("Win")

print(Y)

#if Grid [X] in int(X == [3,3,3,3]):

#TextDisplay2("You win.",(x/3),(y/3),500,25,(255,255,255),(125,125,125),Maingame)

#else:

#Maingame()

def TextDisplay3(txt,x,y,w,h,colour1,colour2,command):

pygame.draw.rect(screen, colour1, (x,y,w,h))

DisplayText = pygame.font.Font("JAi\_\_\_\_\_.TTF", 20)

TextSurface, TextRectangle = RenderFont(txt, DisplayText)

TextRectangle.center = ( (x+(w/2)), (y+(h/2)) )

screen.blit(TextSurface, TextRectangle)

#def ComputerCheck():

if Grid [X] in int(X == [2,2,2,2]):

TextDisplay3("Computer win.",(x/3),(y/3),500,25,(255,255,255),(125,125,125),Maingame)

else:

Maingame()

def Maingame():

#This sets play as being true in order to play the game.

play = True

#This sets counter to True in order for the player and the AI to take their turn placing a counter.

counter = True

#This itteration allows the game to be played and will give the option for the player to quit the game.

while play == True:

for event in pygame.event.get():

if event.type == pygame.QUIT:

play = False

#This will get the mouse position

elif event.type == pygame.MOUSEBUTTONDOWN:

X = Position[1] // (size + 0)

Y = Position[0] // (size + 0)

#print(X)

#print(Y)

#while counter is True, it will allow the user to take their turn.

while counter == True:

#This will move the counter down if the X value is not on the bottom row or if the colour is set to grey.

if X != 5 or Grid[X] == 1:

X = X + 1

counter = True

#This will check if the row below in the specific column has an AI's token in it, therefore it will place the token

#in the position it is on.

elif X + 1 == Grid[2]:

Grid[X][Y] = 3

counter = False

#PlayerCheck()

#If anything else happens, then the counter position will be set to the player's token colour.

else:

Grid[X][Y] = 3

counter = False

#PlayerCheck()

print(Grid)

#0 1156 832

#This is the variable I have used in order to allow the AI to operate.

A = 1

while A == 1:

#This will get the random number for the token positioning on the grid.

X = random.randint(0,870) // (size + 0)

Y = random.randint(0,870) // (size + 0)

#print(X)

#print(Y)

#print("")

#This will move the counter down if the X value is not on the bottom row or if the colour is set to grey.

if X != 5 or Grid[X] == 1:

X = X + 1

A = 1

#This will check if the row below in the specific column has a player's token in it, therefore it will place the token

#in the position it is on.

elif Grid[X] == 3:

X = X - 1

Grid[X][Y] = 2

A = 2

counter = True

#If anything else happens, then the counter position will be set to the AI's token colour.

else:

Grid[X][Y] = 2

A = 2

counter = True

screen.fill(Black)

for row in range(6):

for column in range(7):

#circle(screen,color,(x,y),radius,thickness)

pygame.draw.circle(screen, Colourgame[Grid[row][column]],(column\*size+75, row\*size+75),75, 0)

Position = pygame.mouse.get\_pos()

X = Position[1] // (size + 0)

Y = Position[0] // (size + 0)

#This will update a portion of the screen for the display.

pygame.display.update()

#This will update the entirety of the screen for the display.

pygame.display.flip()

def Level2():

pygame.quit()

#This is the function that determines the position, colour, size and hover function of the two buttons for the login / register screen.

def Buttons1(txt,x,y,w,h,colour1,colour2,command):

#This allows the user to use there mouse when determining to click the buttons. The line gets the position of the cursor.

mouse = pygame.mouse.get\_pos()

click = pygame.mouse.get\_pressed()

#This selection determines the colour of each of the boxes depending on whether the cursor is hovering over a box or not.

if x+w > mouse[0] > x and y+h > mouse[1] > y:

pygame.draw.rect(screen, colour2,(x,y,w,h))

if click[0] == 1:

command()

else:

pygame.draw.rect(screen, colour1,(x,y,w,h))

#This sets the font that will be used for the buttons and the font size which will be 20.

ButtonText = pygame.font.Font("JAi\_\_\_\_\_.TTF", 20)

#This renders the font for the text overall to be the button text.

TextSurface, TextRectangle = RenderFont(txt, ButtonText)

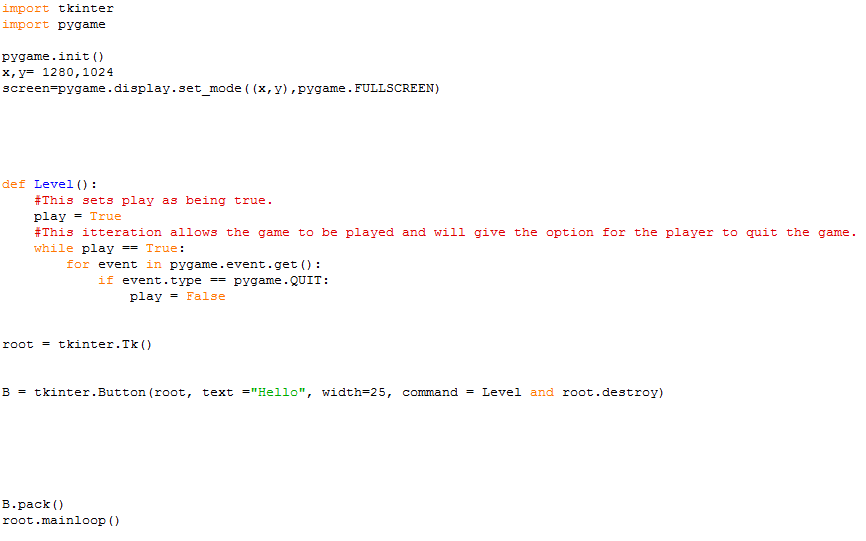
#This determines the positions of the button text within the button so that it is central.

TextRectangle.center = ( (x+(w/2)), (y+(h/2)) )

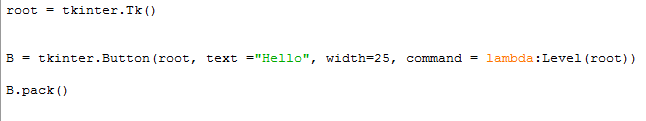
screen.blit(TextSurface, TextRectangle)

root.mainloop()

## Prototypes



Here is a test I did to try merge Tkinter and pygame together in order to fulfil the database requires. This was a simple test which meant I could click a button and then it will load a pygame screen.

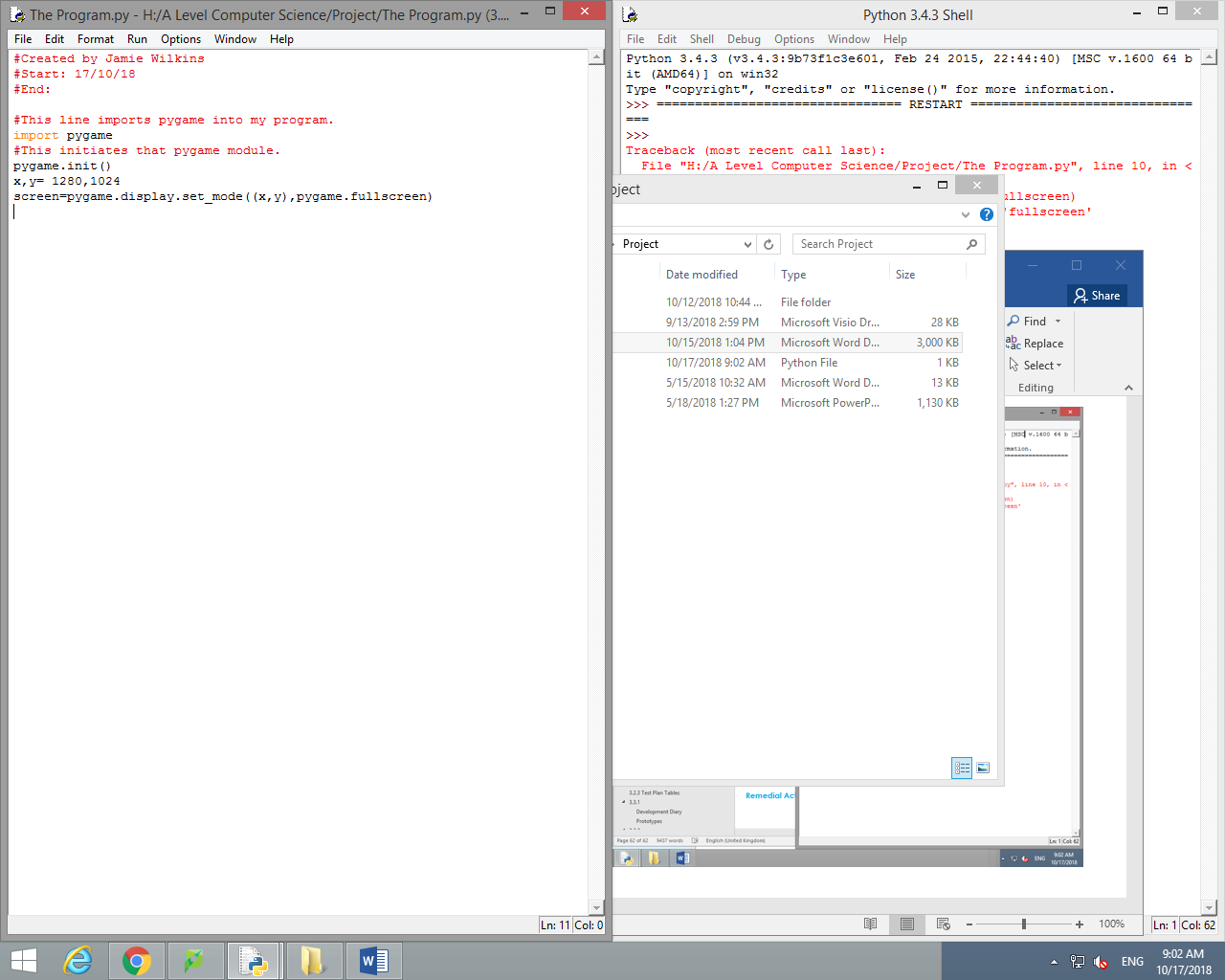


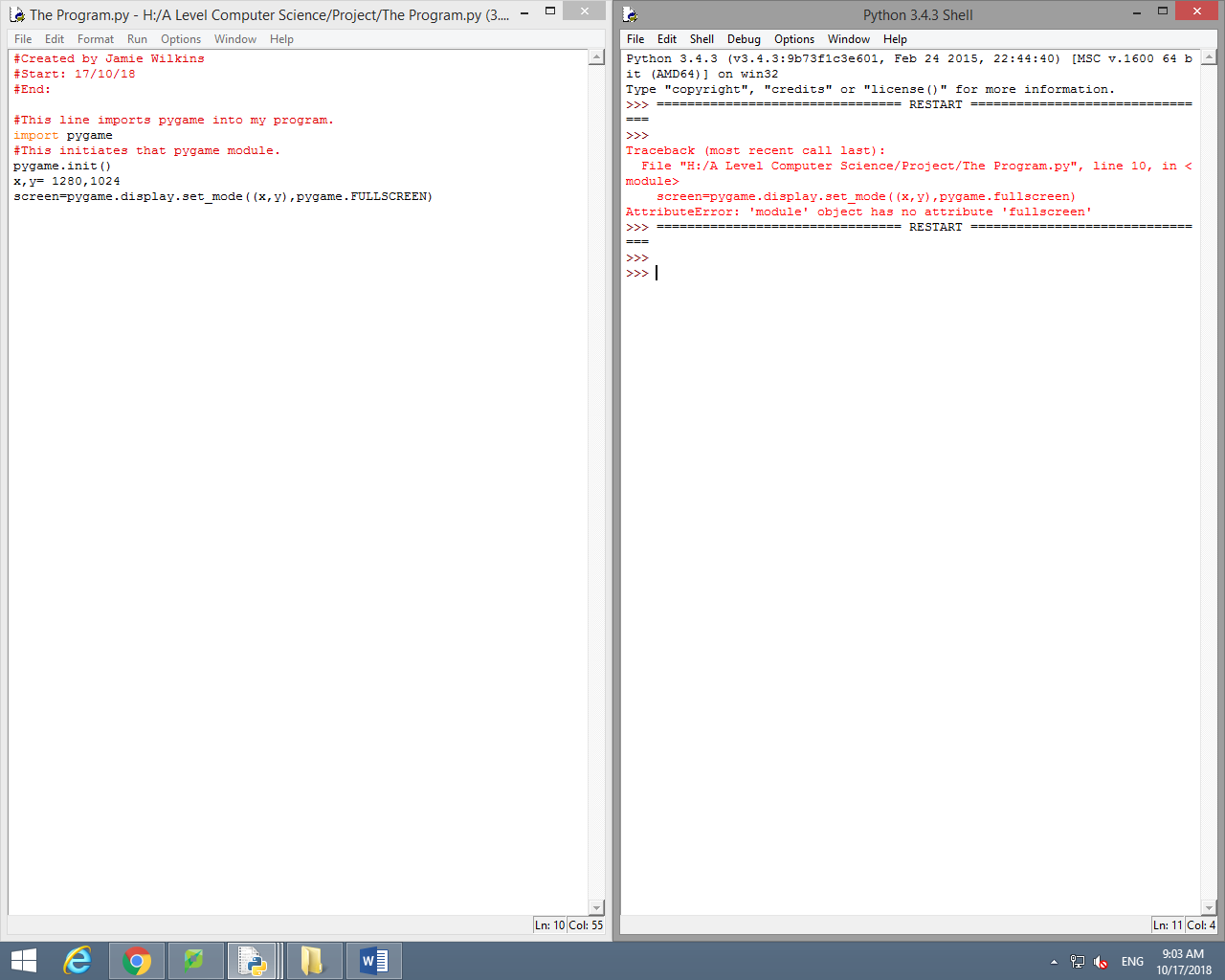
Below is a prototype to test whether I could create a button for the database in Tkinter rather than trying to confuse the pygame code.

# 3.3.2

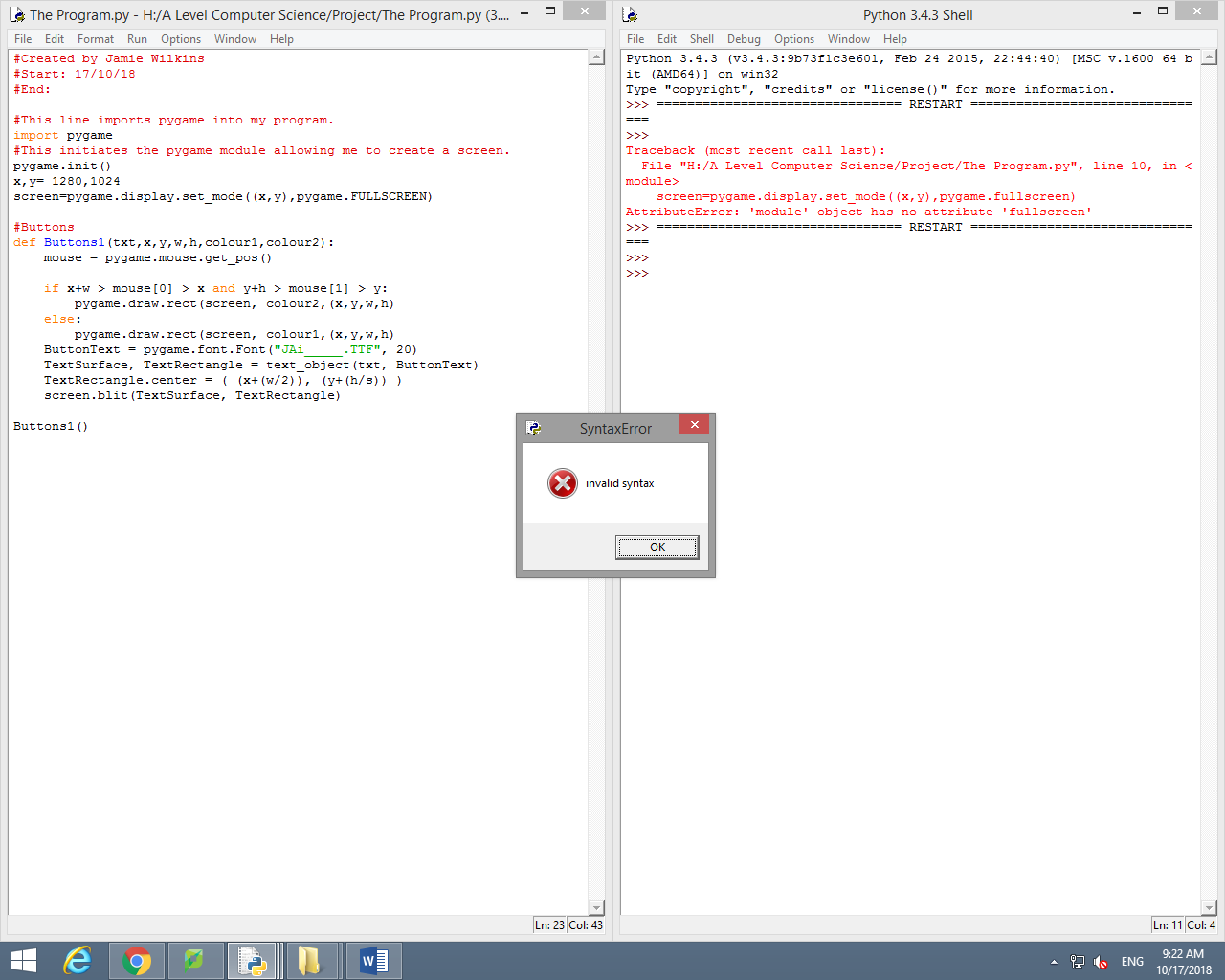
## Evidence of Testing

The error above states that there is an attribute error with the ‘fullscreen’ attribute. As you can see from my code below, this is because the attribute ‘fullscreen’ was written in lower case where as I must be written in upper case.

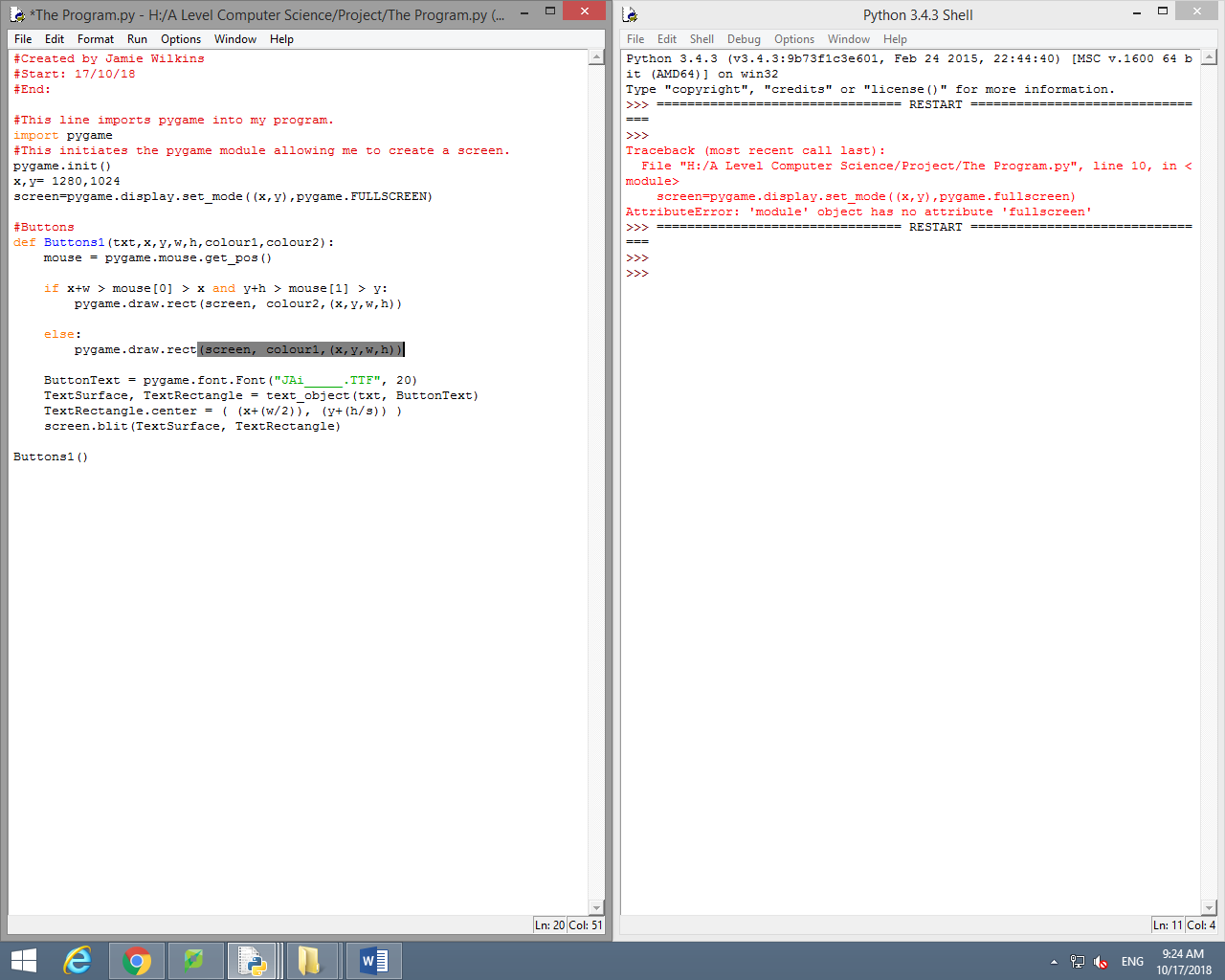




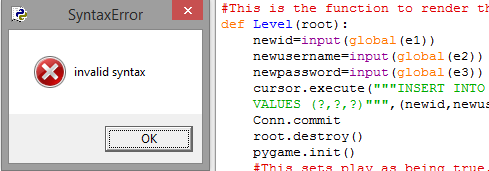
As you can now see I have changed the attribute ‘fullscreen’ to read ‘FULLSCREEN’ and it now does not show an error.



The error above states that the object has no ‘fullscreen’. This occurred because I had not closed all of my brackets.



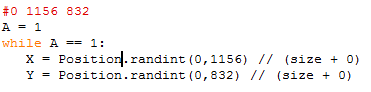
As you can see, after closing all of the brackets, the program will now run without any syntax errors.



Above, you can see that I had a syntax error. This is because I have tried to implement global variables but I have put them inside the input. To change this, I will move them out of the input so it works as two separate function, the first to pull the variable then the second to input it.



This error occurred when I was trying to create the settings button for my home pygame screen. This was because I forgot to break down the section of the name from the ‘x’ co-ordinate. In order to fix it used a comer to break apart the ‘x’ co-ordinate and the string of the name.



The error above is that I have tried to use a determined variable and make it random. By making it be called “random” instead of “Position” then it will determine the random position the AI will place their token.

# 3.4.1

## Testing to inform evaluation

(A)

In this section I will be testing my code in order to try and break it and therefore find problems at the end of the coding process. If am going to be using the table, I have created earlier:

Post Development Test Table (Tests to try to break the code):

|  |  |  |  |
| --- | --- | --- | --- |
| Test Number | What is being tested? | Expected Outcome | What actually happened? |
| 1 | In this test I will be checking to make sure it will allow a new user to be added to the database using their own account details. | I expect it to allow it to not lead to an error and add the account details to the database file. | As you can see when I click login it leads to the main menu and will therefore not error trying to add the account details into the database. |
| 2 | In this test I will be checking to make sure it will allow an existing user to be able to access their account using their own account details stored in the database. | I expect it to allow it to access the account and lead to the main menu. | As you can see when I click login, it leads to the main menu and will therefore not error trying to access the account details into the database. |
| 3 | In this test I will be checking to see whether the validation criteria that I mentioned earlier for both the username and password (having one capital letter, one number and one special character ad under 12 characters). I shall complete this test by inputting an invalid username and password without capital letters or special characters; the password will also be 13 characters long. | I expect this to error and not lead to the main menu. | As you can see when I click login, it leads to the main menu. This means that it isn’t checking the validation of the username or password as it leads to the main menu. |
| 4 | In this test, I will be checking to see what will happen if the user clicks out of the grid when placing their token. | I expect it to error as no valid position on the grid has been selected. | As you can see, an error occurs stating that the positioning is out of range. This shows that it will not allow the user to place their tokens outside of the grid. |
| 5 | In this test I will be checking to see what will happen if the user clicks rapidly on their turn and to see how that will affect the AI. | I expect it to do nothing until the AI takes its turn and then allow the user to click. | As you can see, it does not wait for the AI to complete its move before allowing the user to then take their. You can tell this because there have been four player moves to only two AI so it is struggling to recognise that it must wait for the AI. |
| 6 | In this test I will be checking to see what will happen during the period of transitioning between Tkinter and Pygame and whether the main menu will load after logging in. | I expect it to lead to the main menu (the pygame section) after logging in (the tkinter section). | As you can see it does lead to the main menu (pygame section) when I click the login button on the tkinter screen |
| 7 | In this test, I will be checking to see whether all of the buttons I have created will allow the user to lead to those sections. | I expect them to lead to the necessary function when clicked. | As you can see when I click the play button it leads to the grid for the game.      As you can see when I click the settings button, it leads to the settings menu.    As you can see when I click the token colour button, it does not lead to anything. This is because I have not added in the function to change the token colours.    Like with the token colour button, when I click the background colour button, it does not lead to anything. This is because I have not yet added the function to allow the user to change the background colour.      As you can see when I click the quit button it leads to this error. However, it does close the game down and therefore completes what it needs to do. The reason why this error exists is that it is trying to initialize the font even though the pygame feature of the code has been closed down. |
| 8 | In this test, I will be testing to see whether the tokens will fall on top of one another when they are placed in the same column. | I expect it to place the token above the spot that is already taken. | As you can see, it does not recognise that a token is present and it will overlap the previous token. The example above shows that is not just the user’s token that are suffering this problem but also the AI’s tokens. |

## (B) – Usability

I order to allow for usability function I have took into consideration many different factors when creating the UI. To begin with, I have made my buttons and text boxes a larger size so that they are easier to read and allow more text to be placed in them. I have also chosen to have the font as a simple font to read and at a size 20. With it being larger, it means that people are not squinting in order to try to read the text. Unfortunately, I have not added in the feature to change the token and background colours yet as I have simply ran out of time. If I had more time in order to complete this then it would also cater more for colour-blind users.

## [C] – Questionnaire

Here is a table for my questionnaire that I need to be completed by a test group from the clients and shareholders. It will be used in order to check whether they enjoy it or not and see what needs changing as well as whether it meets their expectations. The results shall also be shown in the table.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Subjects | What did you find good about the game? | What did you not enjoy about the game? | What do you think needs improving with the game? | What sections have met your expectations? |
| Subject 1 | The game loaded fast and looked acceptable, as well as the login screen being useful. | The game did not recognise when I won, and I was able to place tokens over my opponent’s. | Being able to place tokens on top of other tokens. | Login screen and display of the game. |
| Subject 2 | The concept of the game is good so is the logging screen. | The game code not being fully complex and only being able to use the bottom line. | The database comparison system. | The main menu system. |
| Subject 3 | Quick loading.  Good main screen. | Opponent could place over my token.  Settings had nothing/No way out. | Gameplay (opponent being able to place tokens over yours). | The home screen and login. |

# 3.4.2

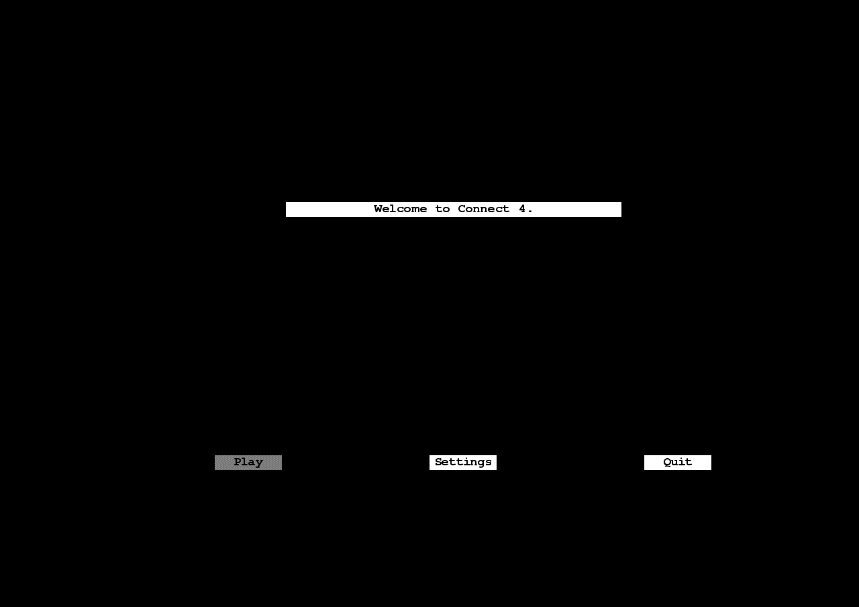
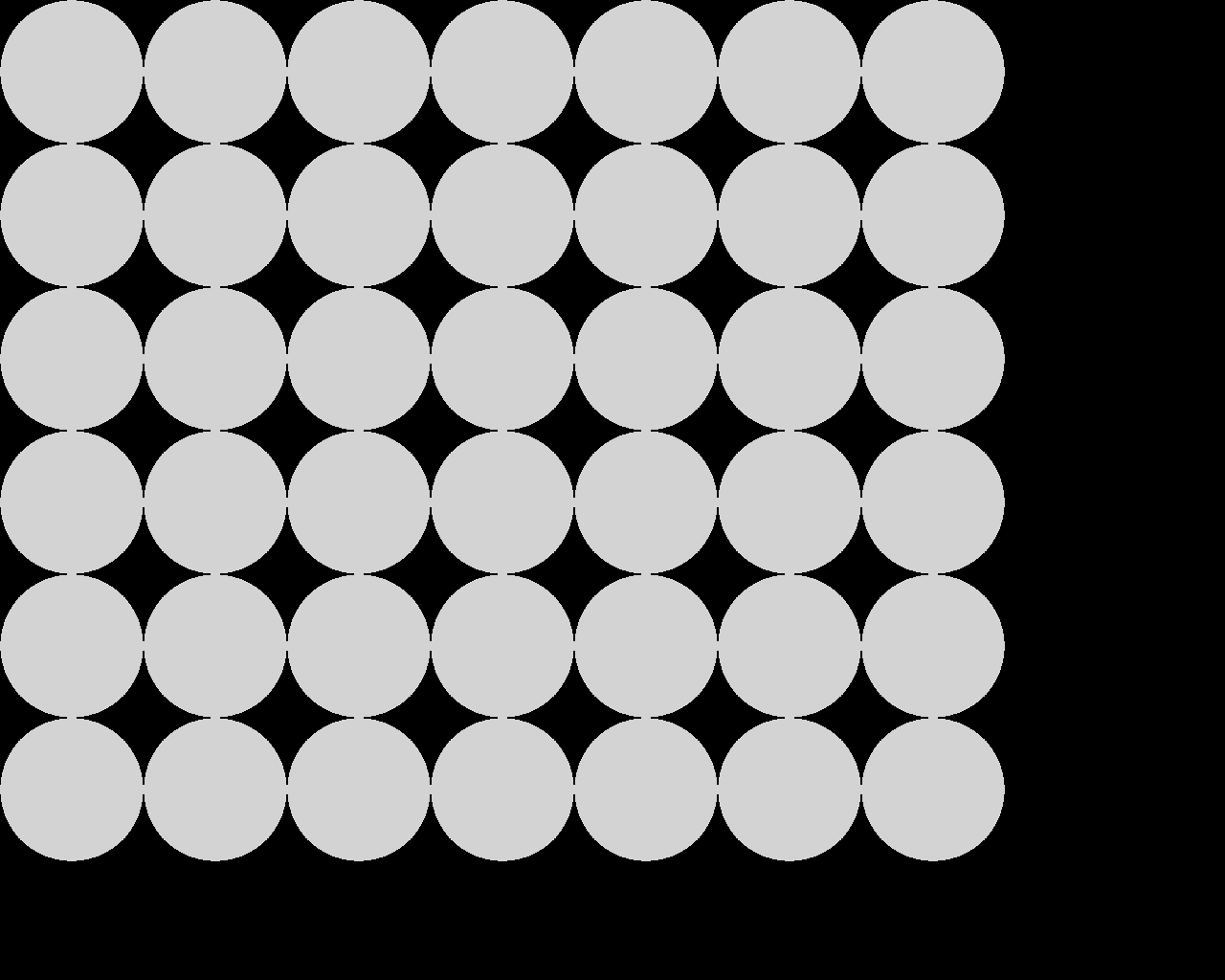
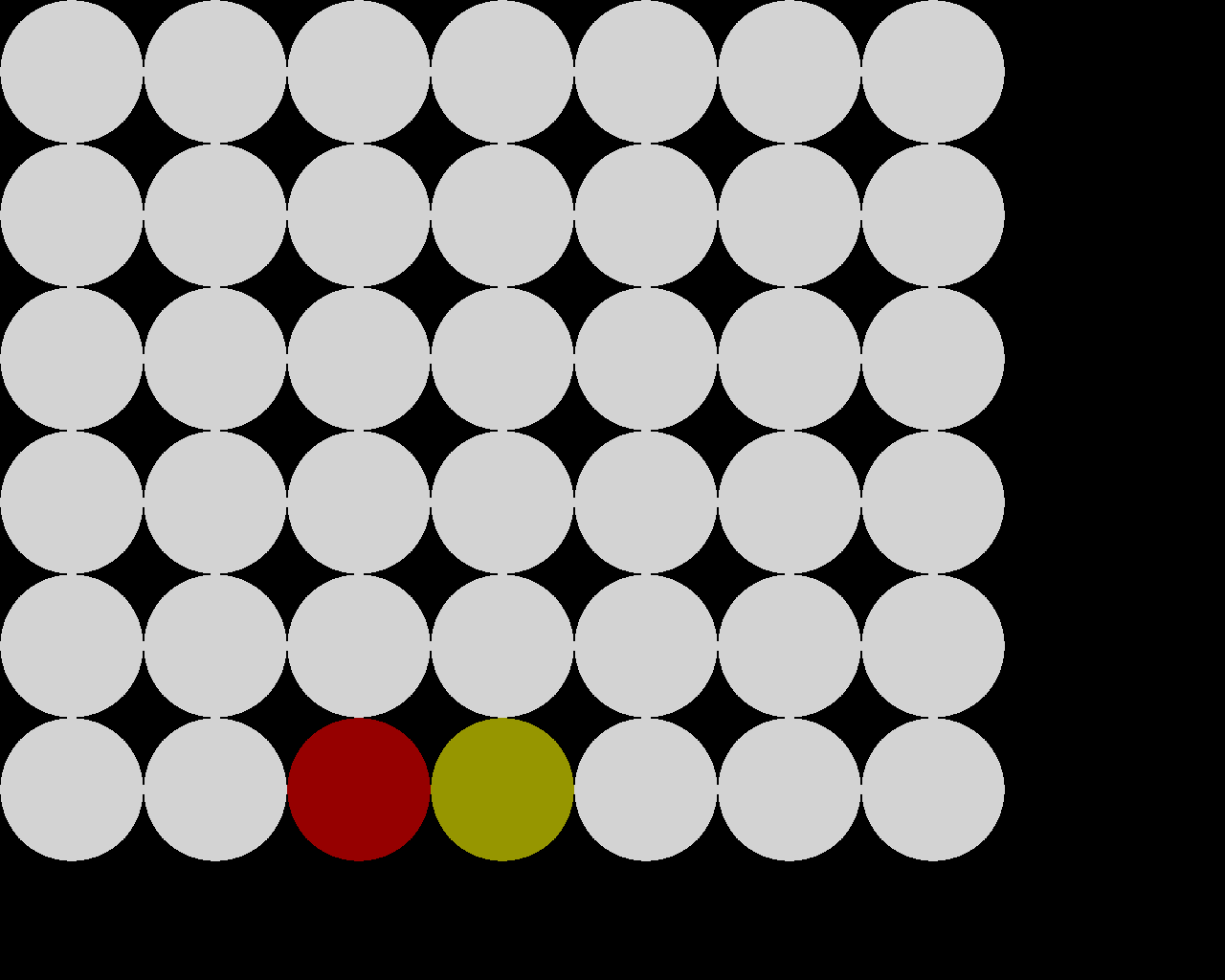
## Success of the Solution

Here is a copy of my success criteria:

* Whether the game will run.
* Whether the AI will respond well.
* Whether I can login.
* Whether the colours are correct for every user (ability to change the colours).

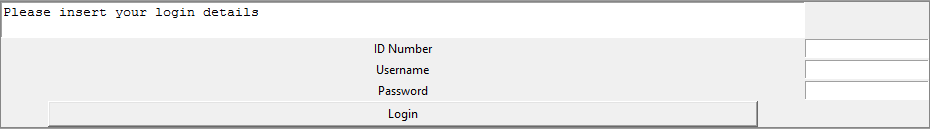
Criteria 1 and 2 (Will the game run and will the AI respond):

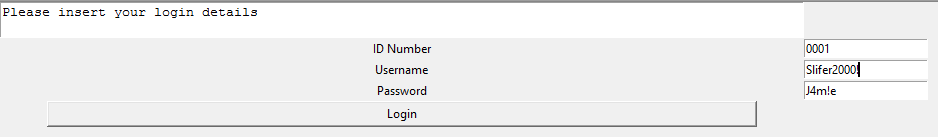
Here you can see that the game will run as my success criteria says. It will display the grid once the play button has been selected and will allow the user to place their tokens in the grid with the AI responding by placing theirs. However, as I have previously mentioned, the tokens do overlap each other at the moment which is therefore causing a problem. This is because of the fact that it isn’t reverting the position before it correctly back to the neutral value I have as “1” in order to have the token to be placed. There this does somewhat meet my success criteria in the fact that it allows tokens to be placed on a grid and allowing four in a row but it fails in the major section where the tokens do not stack.

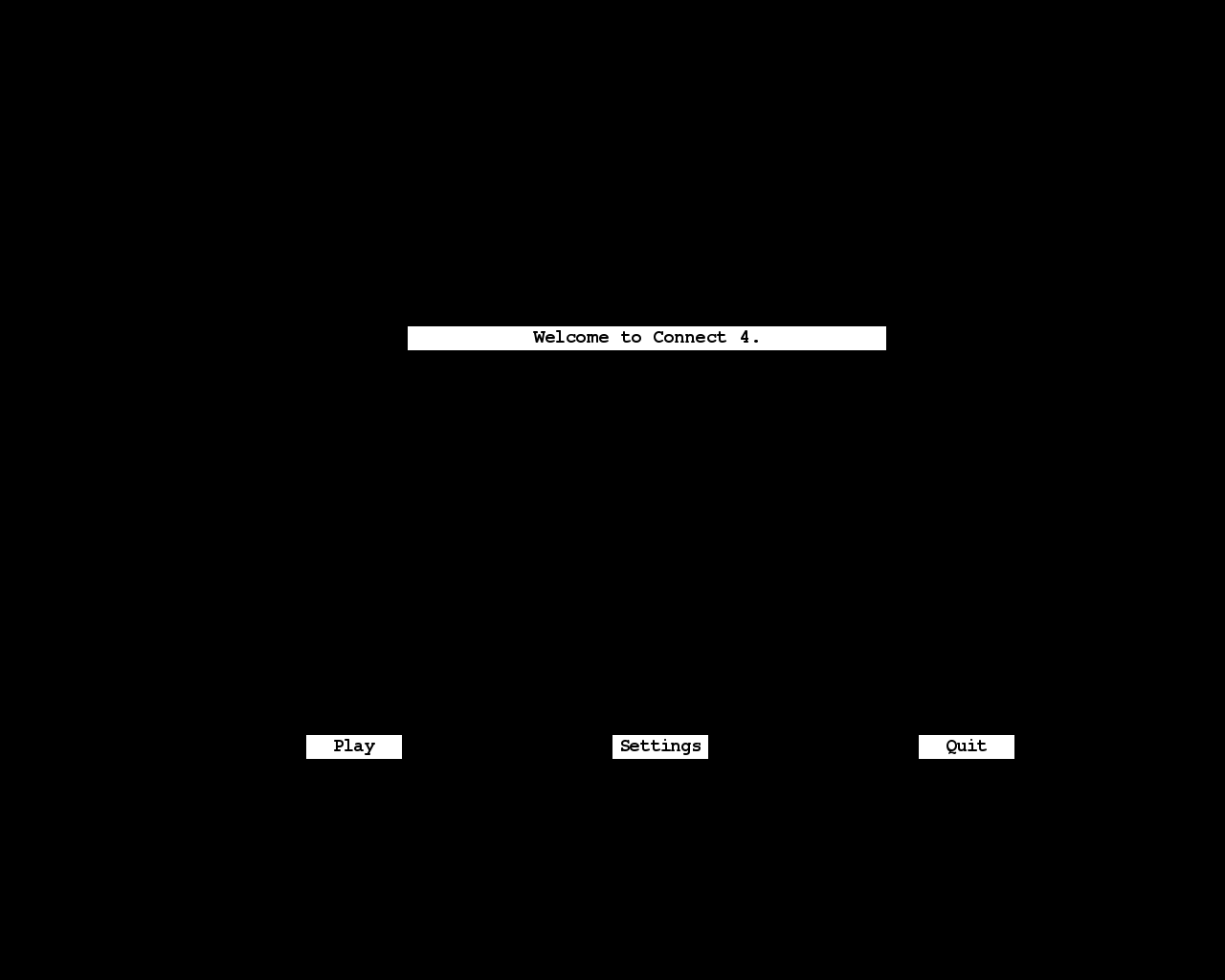
  


Criteria 3 (Can the user login / register):

Here you can see that my login screen does work as it allows new users to register and existing users to login. Once they have done this you can see that it leads straight to the main menu of the game. There this section of the success criteria has succeeded.



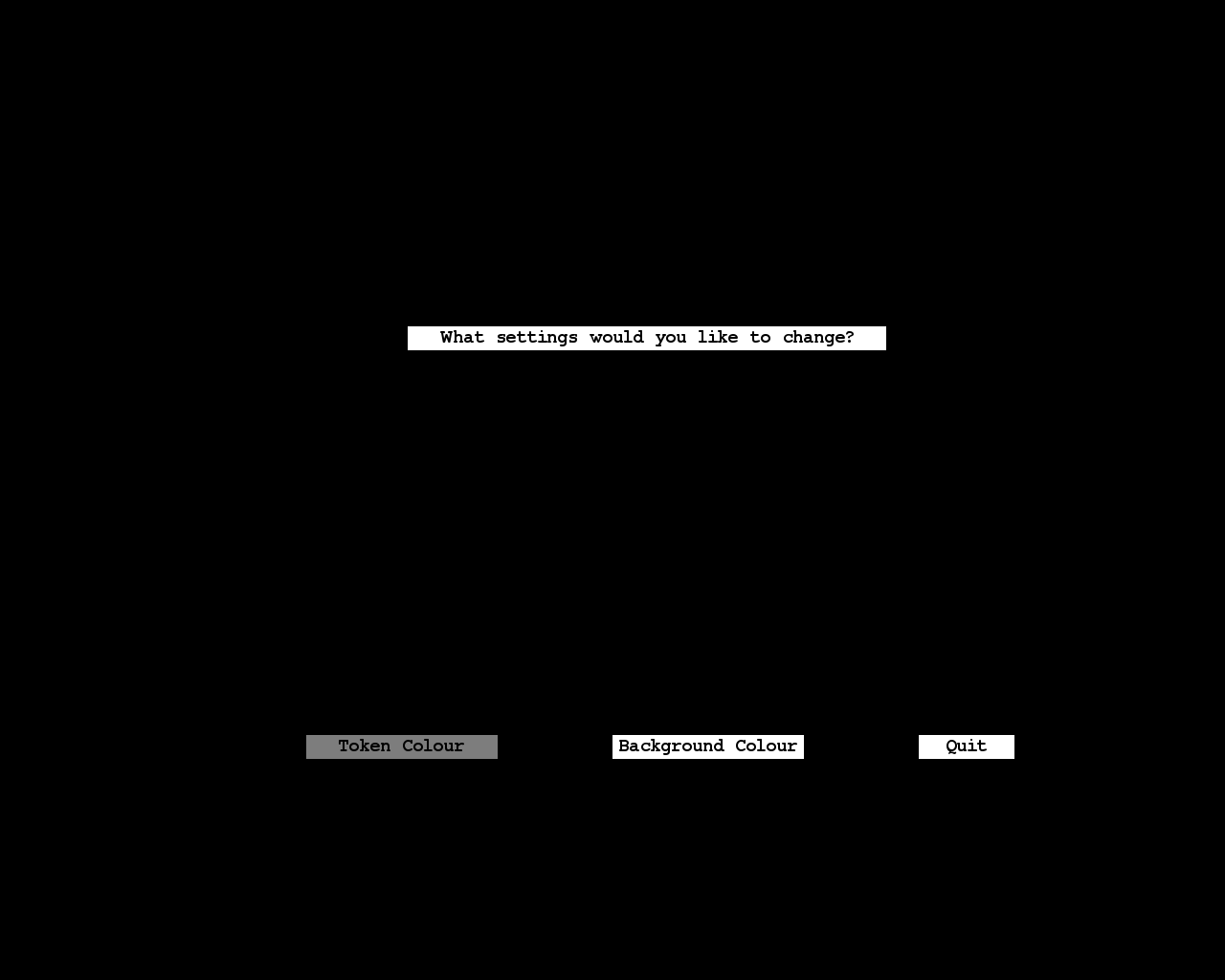


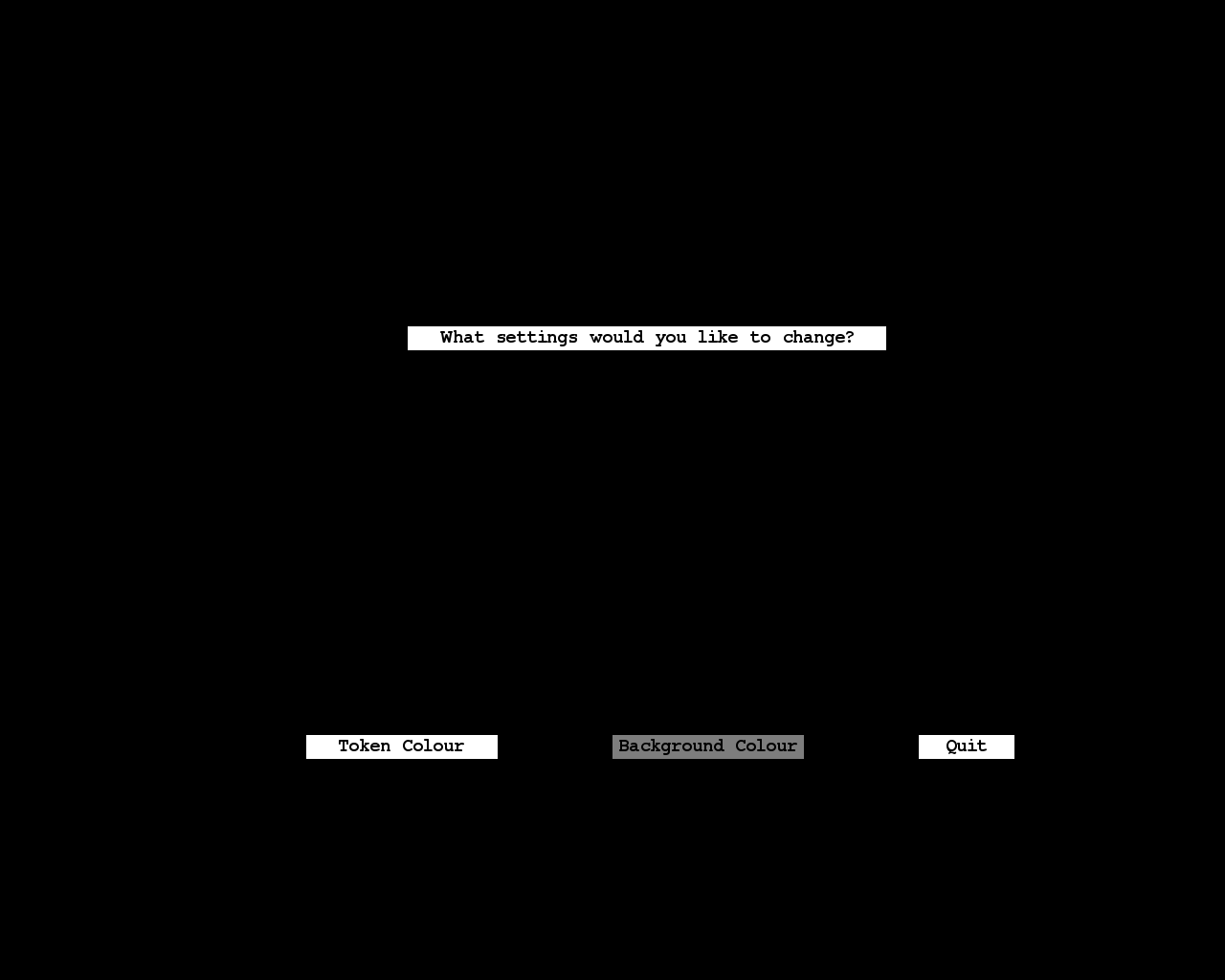


Criteria 4 (Are there user accessibility features related to the colours):



Here you can see that I have facilitated in the settings menu to give the option to the user to allow them to change the user’s token colour, the AI’s token colour as well as the background colour. However, as you can see from the screenshots, when the user clicks these buttons, it doesn’t lead to anything. This is because I ran out of time and therefore I couldn’t add this feature in. This therefore means that I have failed in this section of the success criteria for not catering for usability for all users.

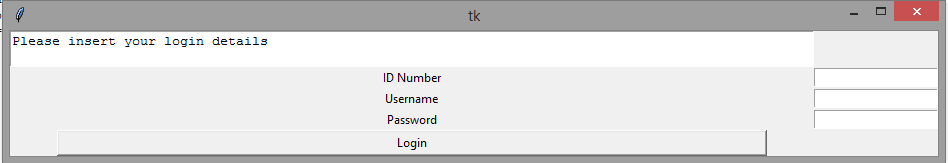




# 3.4.3

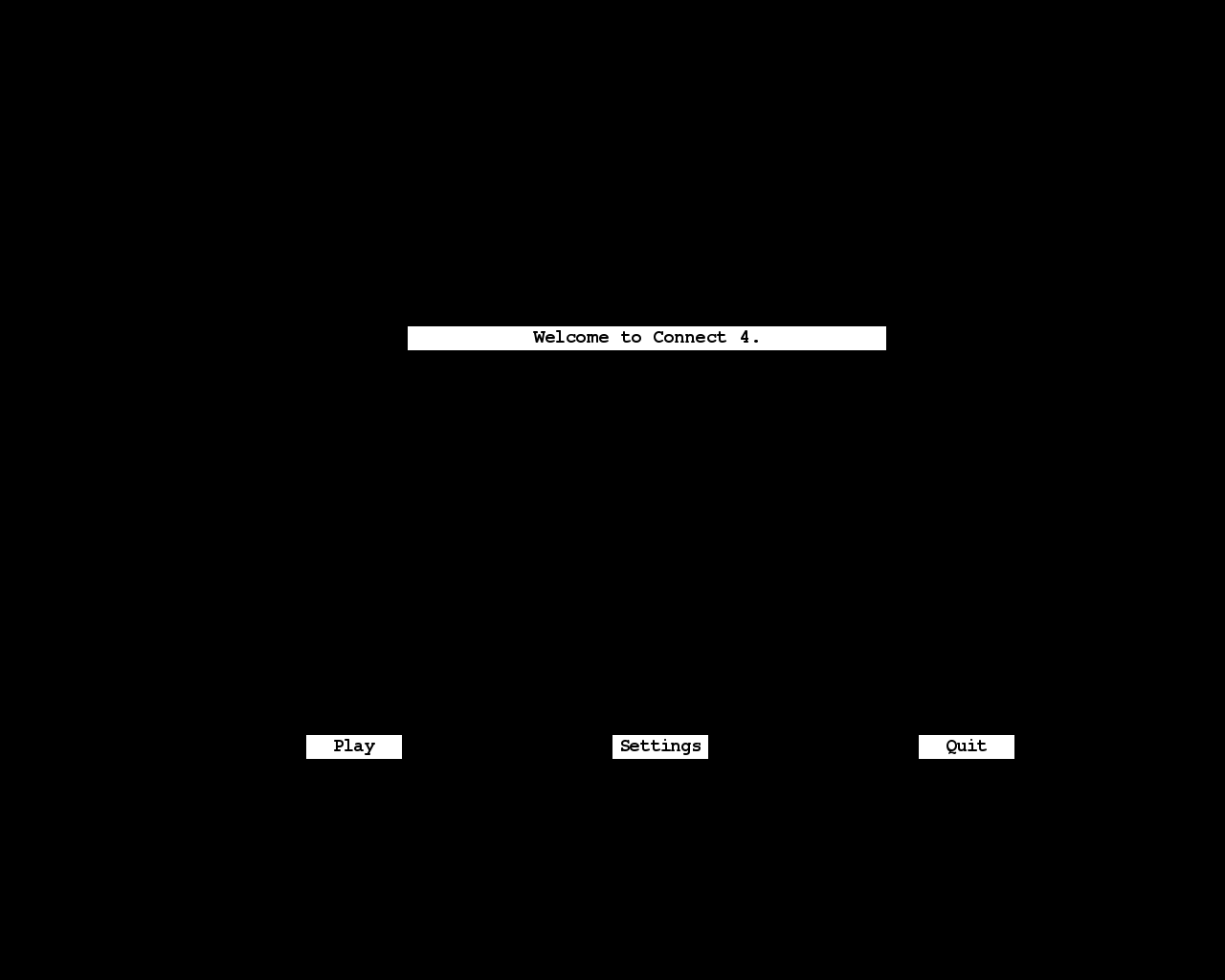
## Describe the final product

This screen has a text box with the text “Please input your login details”. It also has three text input boxes called “ID Number”, “Username” and “Password”. It also has a button called “Login” which, once clicked, will add the login details to the database and the then load the Level() function in order to display the Main Menu.



I think this is a simple design to read as it is clearly laid out and separate from each section. This means that it is easier for the user to navigate when inputting their login details. I also believe that the colour of the text is appropriate with the background colour that is being used. This is because it stands out having a dark colour for the text and a light colour for the background, the contrast makes it easier to read. However, if I would change anything it would probably be the size of the screen and therefore the size of the text. This would be a necessary change in the future because if the screen and text is to small then it may be hard for some users to see what each section is asking for.

Once the user has logged in it leads to the Main Menu. There is a text display box saying “Welcome to Connect 4” as a header of the menu. Then there are three buttons, one which says “Play”, one that says “Settings”, and one that says “Quit”. When the “Play” button is clicked it will load the Maingame() function which will lead to the game. When the “Settings” button is clicked it will load the Level1() function which will lead to the settings menu and when the “Quit” button is clicked it will load the Level2() function which will exit the game.

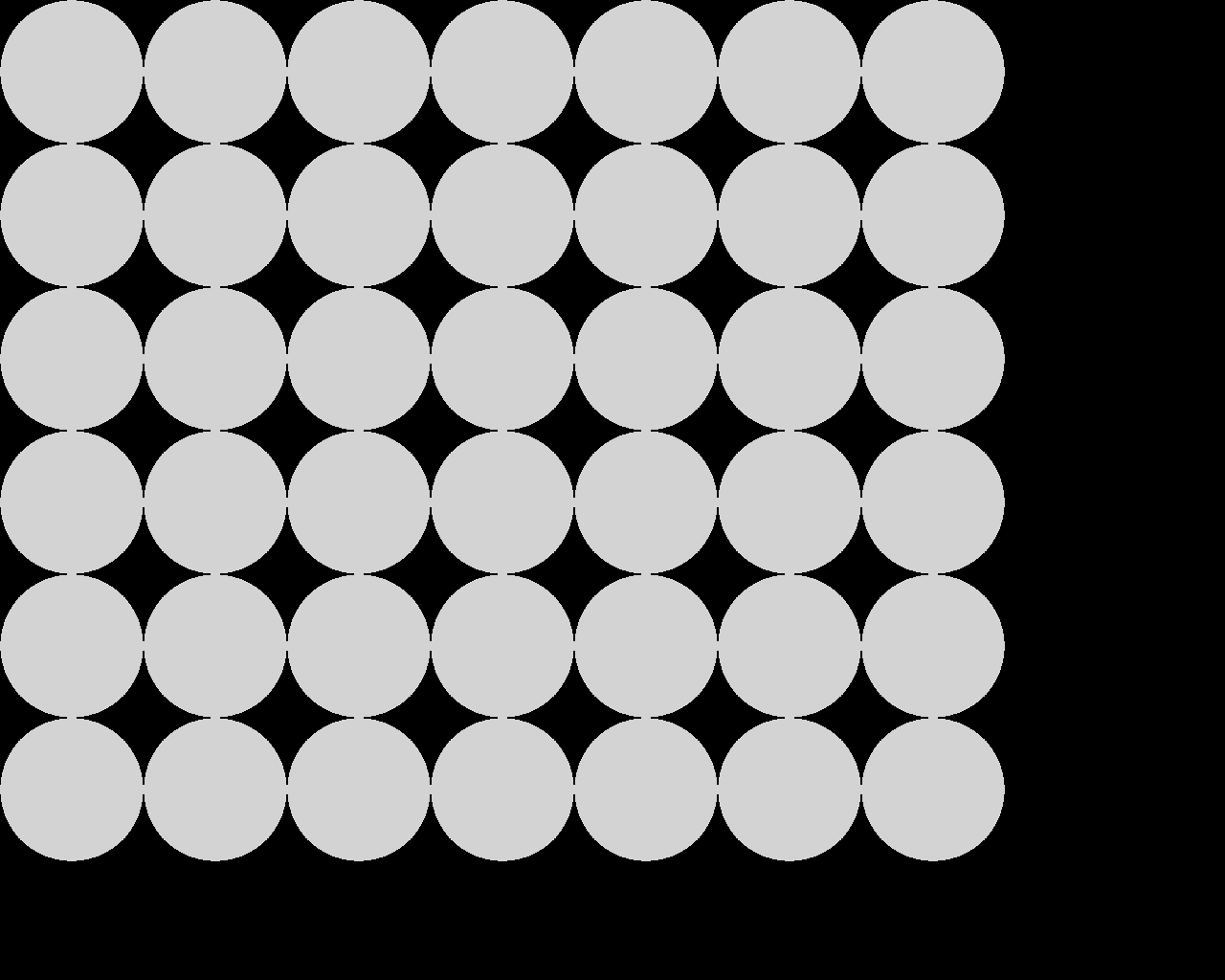


I think this is a basic design that is clearly laid out and separate from each section. This means that it is easier for the user to navigate when deciding what section to of the program to go to. However, I believe that the colour of the background could be changed possibly to white. This is because, in my opinion, it looks very messy and not smoothly put together. I also believe I would change the size of the screen and therefore the size of the text. This would be a necessary change in the future because if the screen and text is to small then it may be hard for some users to see what each section is asking for.

Once the user has clicked the settings button it leads to the Settings Menu. There is a text display box saying “What settings would you like to change?” as a header of the menu. Then there are three buttons, one which says “Token Colour”, one that says “Background Colour”, and one that says “Quit”. At the moment, when the “Token Colour” and “Background Colour” buttons are clicked it won’t lead to anything as I haven’t had the time to add this feature into the game yet. Finally, when the “quit” button is clicked it will load the Level2() function which will exit the game.

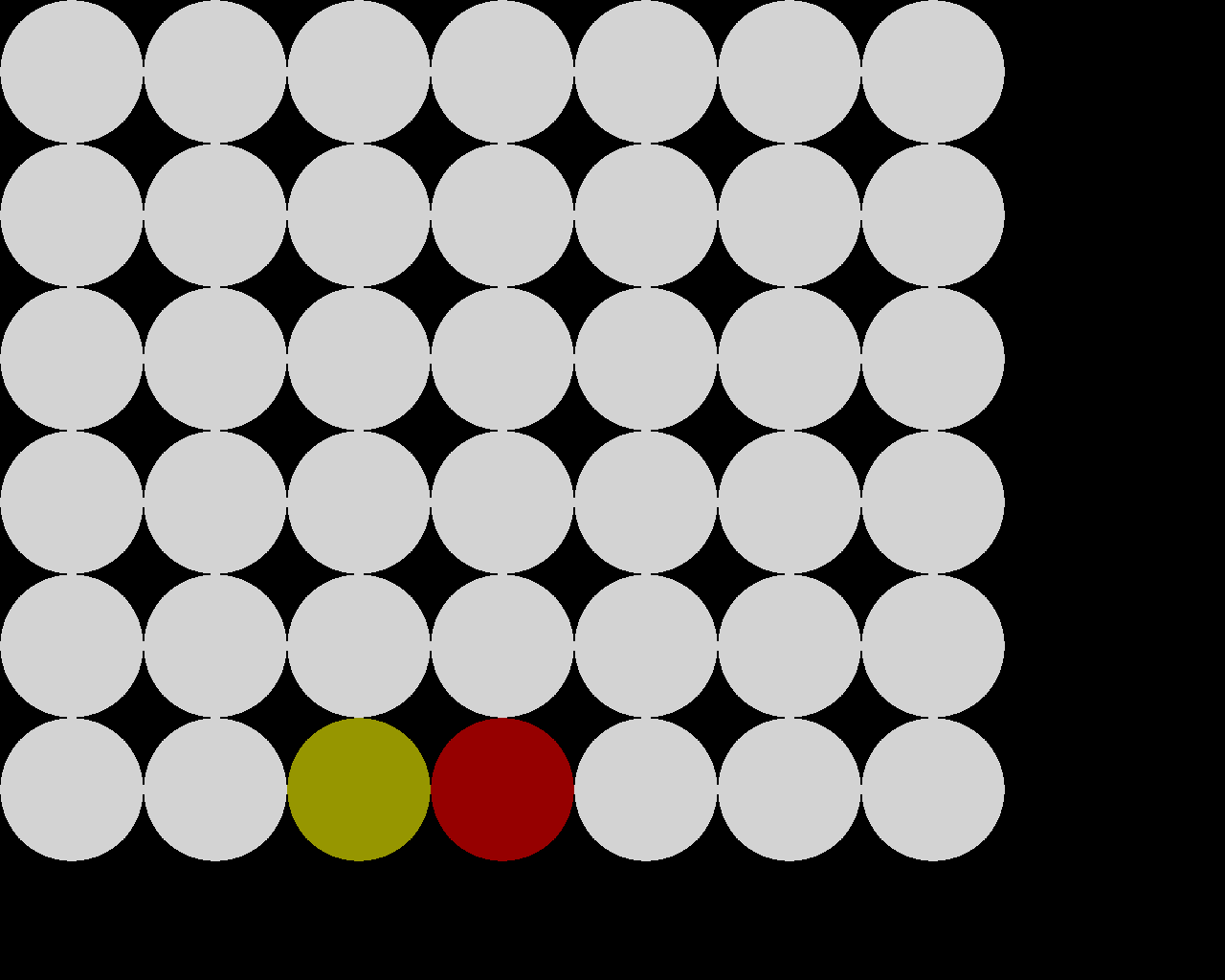


I think this is a basic design that is clearly laid out and separate from each section. This means that it is easier for the user to navigate when deciding what section of the program to go to. However, I believe that the colour of the background could be changed possibly to white. This is because, in my opinion, it looks very messy and not smoothly put together. I also believe I would change the size of the screen and therefore the size of the text. This would be a necessary change in the future because if the screen and text is to small then it may be hard for some users to see what each section is asking for.



This screen is the grid once the play button has been clicked. It contains a 7 x 6 grid (the standard size) with selections able to be pressed in order to drop the tokens into it.

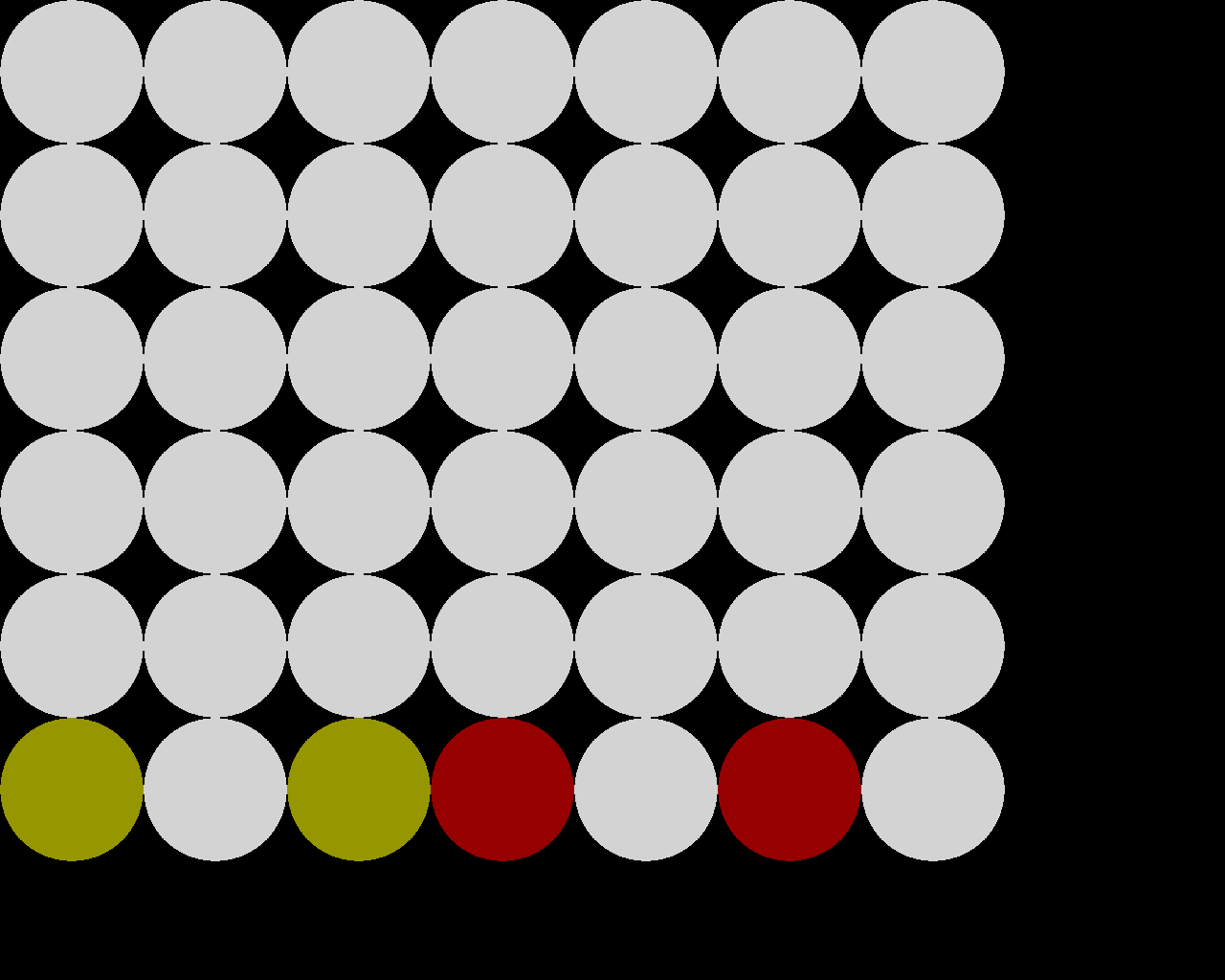
I think this is a basic design that is clearly laid out with each grid section separate from one another. This means that it is easier for the user to navigate when deciding what section to place their token in. However, I believe that the colour of the background could be changed possibly to white. This is because, in my opinion, it looks very messy and not smoothly put together. If I had the time to add the feature of changing the background colour, then it may appeal to more users.



This screen is the grid once the user has placed a token. As you can see when the user has placed their token, the AI has also placed their token.

As you can see I have used two different colours for both the user and AI’s tokens. Red for the user and yellow for the AI. This means that it is easy to establish who owns what space. If I had more time, then I would add the feature to change the token colours to make the user accessibility a lot better for specific users.

This screen is the grid once the user has placed their second token. As you can see when the user has placed their token, the AI has also placed their token.



As you can see I have used two different colours for both the user and AI’s tokens. Red for the user and yellow for the AI. This means that it is easy to establish who owns what space. If I had more time, then I would add the feature to change the token colours to make the user accessibility a lot better for specific users. It is clear to see that the colours don’t get impacted when the user takes a second turn which makes it easier to plan a strategy.

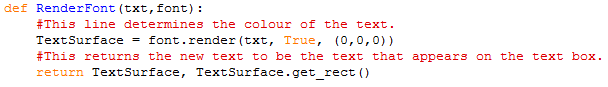
# 3.4.4

## Maintenance and development

I have tried to keep my program well maintained by commenting and indenting my program when needed in order to keep it easier to operate with.

Here are some examples of where my program is commented:

(1)



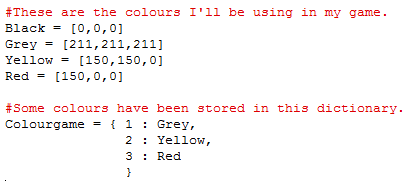
I have commented my code here in order to remind me exactly what each line does. The first line determines the colour the text will be as it says. The second will return this factor to each line of text within the program.

(2)



I have commented my code here in order to remind me that this is the section that tells me what the screen size is as well as how much space it takes up on the screen.

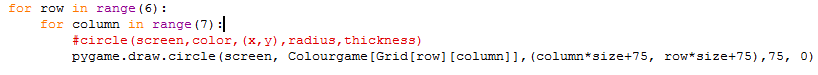
(3)



I have commented my code here to remind me that this section of code involves the colours used in the game. It also reminds me that the dictionary I have used includes the colours I have used and the number they are assigned to.

Here are some examples of where I have indented my code:

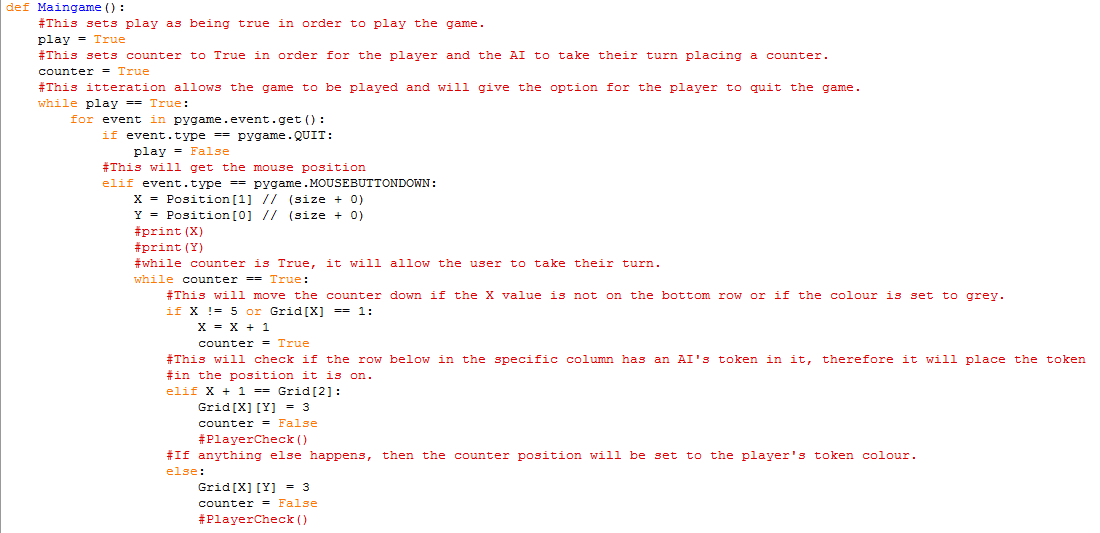
(1)



I have indented my code here. This specific section will draw the circles for the grid. It needs to be indented so that it:

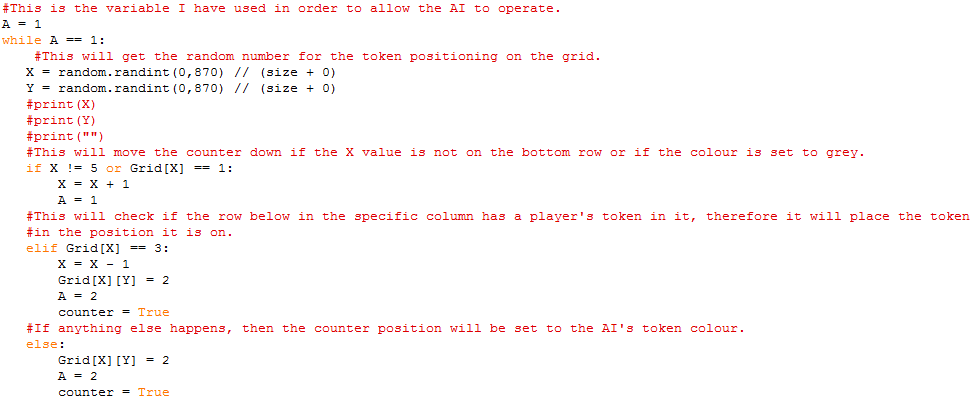
* First creates the rows,
* Then the columns,
* Finally, it draws the circles.

(2)



I have indented my code here in order to allow the different selections that can happen depending on the specific section of the grid that the token is on when the user selects a position. It will either fall once more, place above an AI’s token or just place a user token.

(3)



I have indented my code here in order to allow the different selections that can happen depending on the specific section of the grid that the token is on when the AI selects a position. It will either fall once more, place above a user’s token or just place an AI token.

## (B) – Changes I would make in the future

If I had more time I would firstly change the fact that the tokens cannot stack on top of one another in the grid. The reason this isn’t already a part of it is because I didn’t know how to alter the fact that it wasn’t resetting the column above properly and it therefore was getting confused. Had I resolved this then I would have been able to allow the token would stack on top of one another. I would also add the leaderboard function into the game so that the program will track the number of wins a user has as well as their settings and port that information to be stored in the database. This wasn’t done because I had run out of time. The next thing that I would add would be the ability to change the colours of both the user’s and the AI’s tokens as well as the background colour. With this being in the program it would allow people with sight difficulties to access the game a lot easier as they can change the colours to what is suitable for them. Therefore, this means that the game would appeal to a larger audience. Finally, if I had more money and skill I would re write this program in java script. I think this would be a sensible decision as it would allow me to develop my program with more complex algorithms that are hard to make in python. It would also allow me to use a UI (User Interface) instead of just basic screens for the navigation for the menus. For a final comment, I have enjoyed working on the program. However, I think it is clear to see that I was very ambitious at the beginning of the project but with extra time and skill I believe that this project would be completed to the success criteria and design set out in the beginning of the project.