**Analysis and Design**

In this assignment I have been asked to design and write java program that uses an array to make a 2 player grid game. The users enter a number which corresponds to a position on the grid. The program then replaces the blank space (0) with their corresponding player number. The first player with a 2x2 grid of their number wins the game.

In this program I will create two classes. The user class which will ask the user what game type they want. It will then create instances of the Game class and call the methods that will run within a ‘for’ loop. The counter in this for loop will be set to 25 as that is the array size. This is so the user(s) can enter a value in all the array elements. The Game class will ask the player for that turn for their desired position. There will then be a method that checks that the chosen position is not already taken, if it is not taken it will place their player number there. Each time a user has entered their number it will check to see if there are any 2x2 squares for that specific player. If there is then it will display who won and stop the program.

For the extended requirements I was set with the take to implement a computer choice instead of a second player. To do this I asked the user at the start what game type they would like to play. If they chose the computer option it will go to the loop that calls a computer Choice method that generates a random number and then uses that number as if it was a second player.

|  |
| --- |
| Game |
| - int playerUser  - int array [][]  - int nextTurn  - int robotChoice  - boolean gameType |
| + Game (boolean)  + askPlayer(): void  + playerPoint(): void  + gameBoard(): void  + checkWin(): void  + getRobotChoice(): int  + changeTurn(): int  + printDraw(): void |

|  |
| --- |
| GameUser |
|  |
| Main(String[]) |

**Pseudocode for Methods**

METHOD checkWin ()

INPUT

OUTPUT

LOCAL DATA: INT x1, INT x2, INT y1, INT y2

FOR (INT sets=1 sets<5 INCREMENT sets)

FOR (INT count=1 count<5 INCREMENT count)

IF (array[x1][y1]==nextTurn){

IF (array[x1][y2]==nextTurn)

IF (array[x2][y1]==nextTurn)

IF (array[x2][y2]==nextTurn)

PRINT Player (Current Player) is the winner!

STOP Program

INCREMENT y1

INCREMENT y2

y1 = 0

y2 = 1

INCREMENT x1

INCREMENT x2

METHOD main

INPUT args

OUTPUT READ an integer from the keyboard. Make sure the input is either 1 or 2.

LOCAL DATA boolean robot

SET up Instance for Game class

FOR (INT count=1 count<=25 INCREMENT count)

CALL Game Class Methods

METHOD playerPoint()

INPUT

OUTPUT

LOCAL DATA INT playerRow, INT playerCol

playerRow = (playerUser -1)/5

playerCol = (playerUser -1)%5

WHILE (array[playerRow][playerCol]>0)

PRINT “This position has already been taken”

IF (gameType==true)

CALL getRobotChoice()

ELSE IF (gameType==false)

CALL askPlayer()

playerRow = (playerUser -1)/5

playerCol = (playerUser -1)%5

array[playerRow][playerCol] = nextTurn

METHOD gameBoard()

INPUT

OUTPUT

FOR (INT row=0; row<array.length; row++) {

FOR (INT col = 0; col < array[row] Length; col++) {

PRINT (array[row][col])

IF (col < array[row] Length - 1)

PRINT " "

}

PRINT New Line

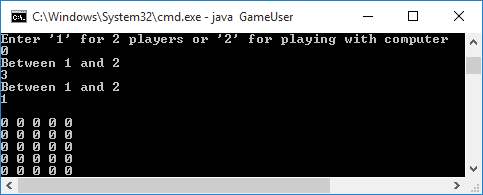
}

PRINT New Line

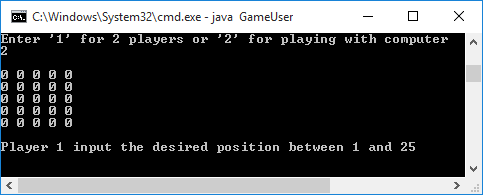
**Testing**

In order to make sure my java program works correctly I need to test each element of the program. I need to check if the input values are being retried if below 1 and above 25. And run through the game which has different events that occur.

*Test: The input value for game type can only be the values 1 and 2.*

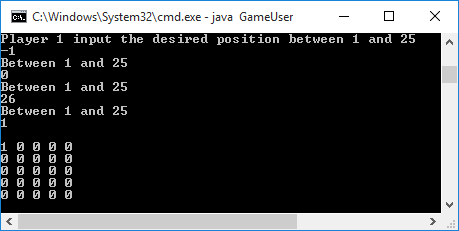


|  |  |
| --- | --- |
| Input | Expected |
| 0 | Prints invalid message |
| 3 | Prints invalid message |
| 1 | Continue |

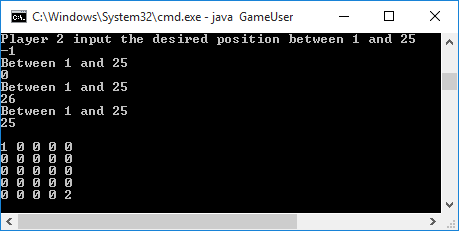


|  |  |
| --- | --- |
| Input | Expected |
| 2 | Continue |

*Test: The input value for player 1 on should be between 1 and 25.*

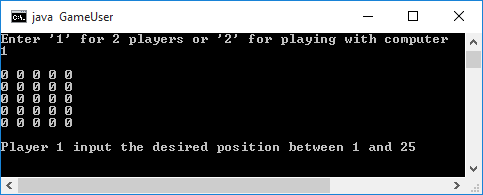


|  |  |
| --- | --- |
| Input | Expected |
| -1 | Prints invalid message |
| 0 | Prints invalid message |
| 26 | Prints invalid message |
| 1 | Continue |

*Test: The input value for player 2 on should be between 1 and 25.*

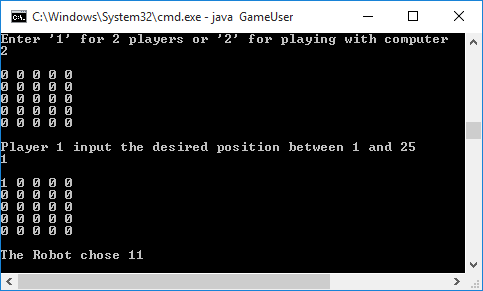
|  |  |
| --- | --- |
| Input | Expected |
| -1 | Prints invalid message |
| 0 | Prints invalid message |
| 26 | Prints invalid message |
| 25 | Continue |

*Test: To see if the 2 player game starts and the array is outputted correctly and the program asks for the first player to enter a value.*



|  |  |
| --- | --- |
| Input | Expected |
| 1 | Starts the 2 player gametype |

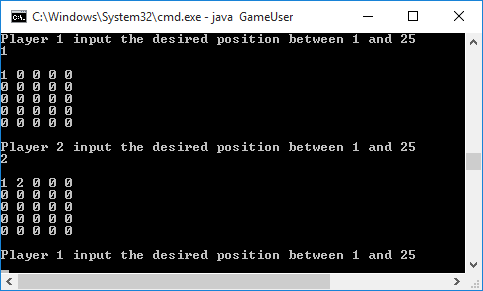
*Test: To see if the computer game starts and the array is outputted correctly and the program asks for the first player to enter a value.*



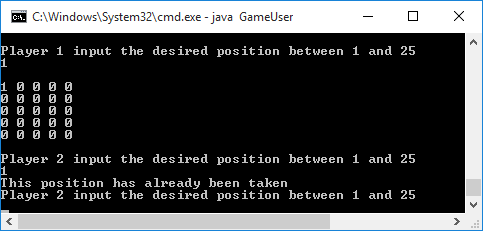
|  |  |
| --- | --- |
| Input | Expected |
| 1 | Starts the computer gametype |

*Test: To see if the output values are correct using the users inputs. (Player 1 positions replaced with a 1 and Player 2 positions replaced with a 2)*

|  |  |
| --- | --- |
| Input | Expected |
| 1 | Replaces that position with the number 1 |
| 2 | Continue |



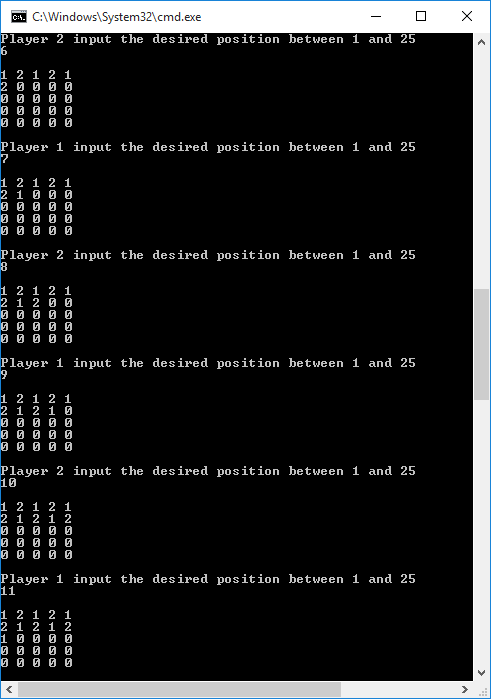
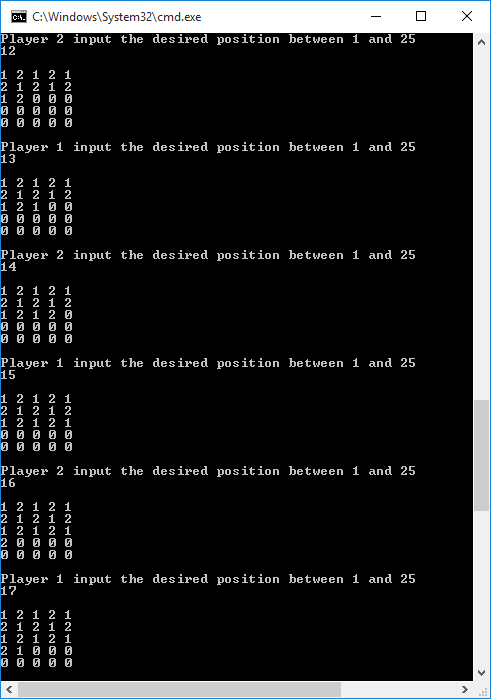
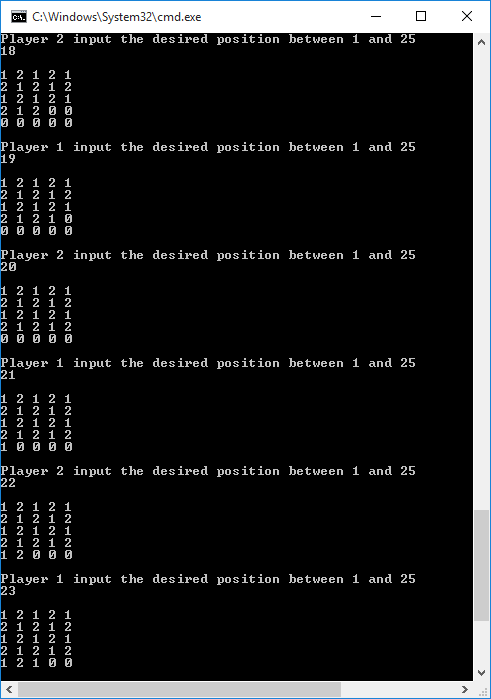
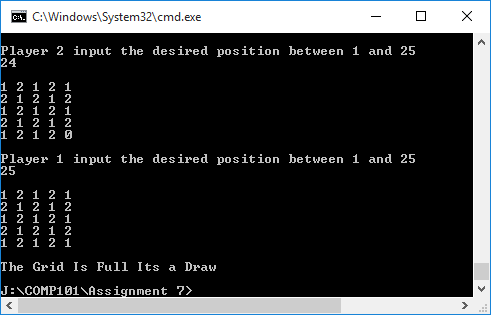
*Test: To see if the retry position method works for when a player enters a position that is already taken.*

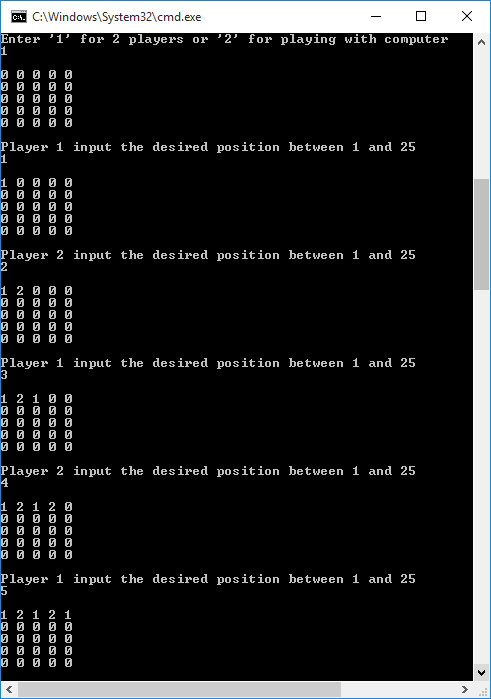


|  |  |
| --- | --- |
| Input | Expected |
| 1 | Replaces that position with the number 1 |
| 1 | Print invalid message and try again |

*Test: To see if the draw message prints when all the spaces are full.*

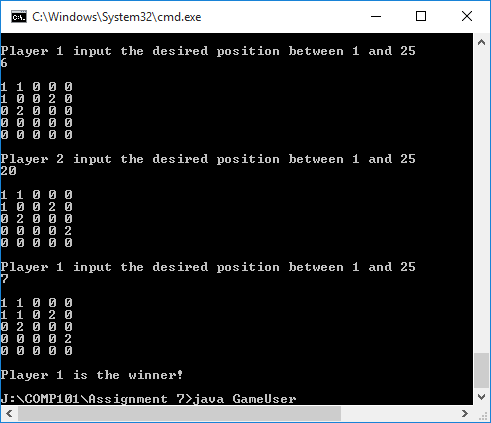
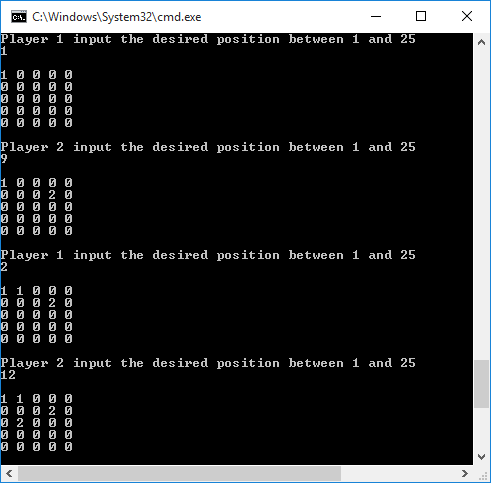
|  |  |
| --- | --- |
| Input | Expected |
| 1-25 | Prints the Draw message and stops the program |





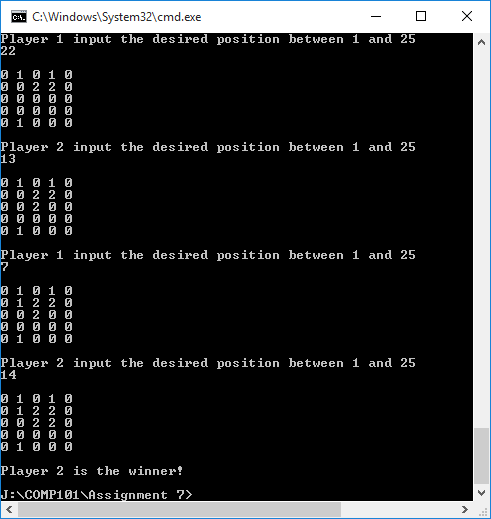
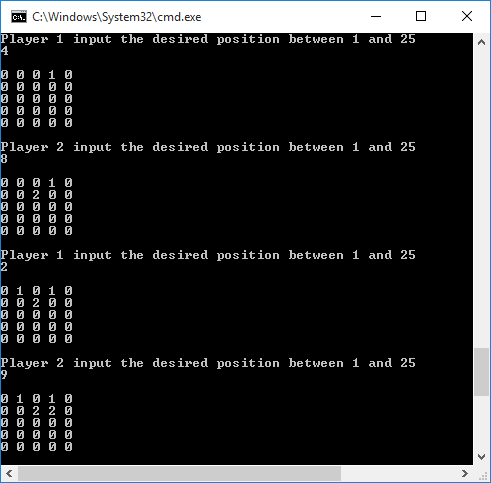
*Test: To see if the win message appears when there is a 2x2 grid. (With Player 1)*

|  |  |
| --- | --- |
| Input | Expected |
| 1,9,2,12,6,20,7 | Prints the win message and stops the program |



*Test: To see if the win message appears when there is a 2x2 grid. (With Player 2)*

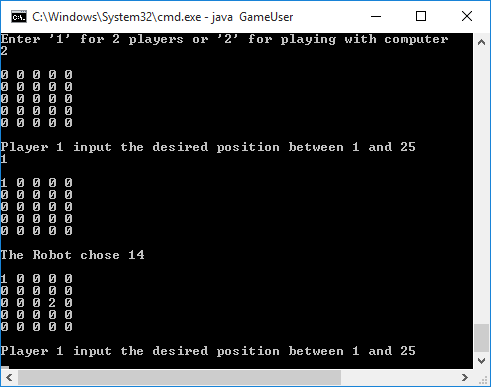
|  |  |
| --- | --- |
| Input | Expected |
| 4,8,2,9,22,13,7,14 | Prints the win message and stops the program |



*Extended Tests: For the computer choices I used a simple random number generator with a min and max value of the grid size. This was then used as the computers choice and inputted as player 2.*

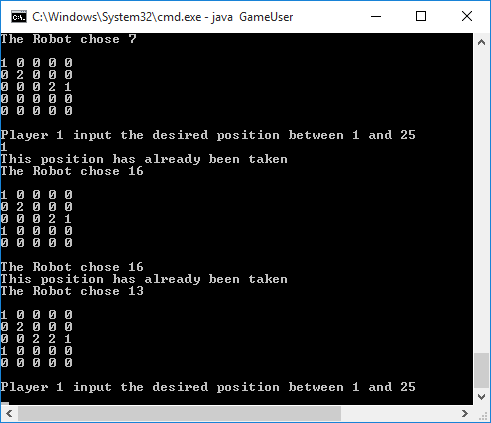
*Test: To see if the computer choice is displayed and the position is replaced with a 2*

|  |  |
| --- | --- |
| Input | Expected |
| 1 | *computers choice is then displayed and the position is replaced with a 2* |



|  |  |
| --- | --- |
| Input | Expected |
| 15,1 | *Computer method is called again and the and the position is replaced with a 2* |

*Test: To see if a message appears when the random number position has already been taken and then regenerates another number.*



*In this test it regenerated the number 16 twice. This is because it randomly generates a number and does not take into account the positions that are occupied. I did not have enough time to look into this however the program still carries on when it eventually chooses another number.*