GAME DEVELOPMENT PROJECT

Software Development – Game: Noughts and Crosses

JUNE 29, 2018

Student Name : Jason Fernandes
Tutor : Mizanur Rahman and Vy Tran

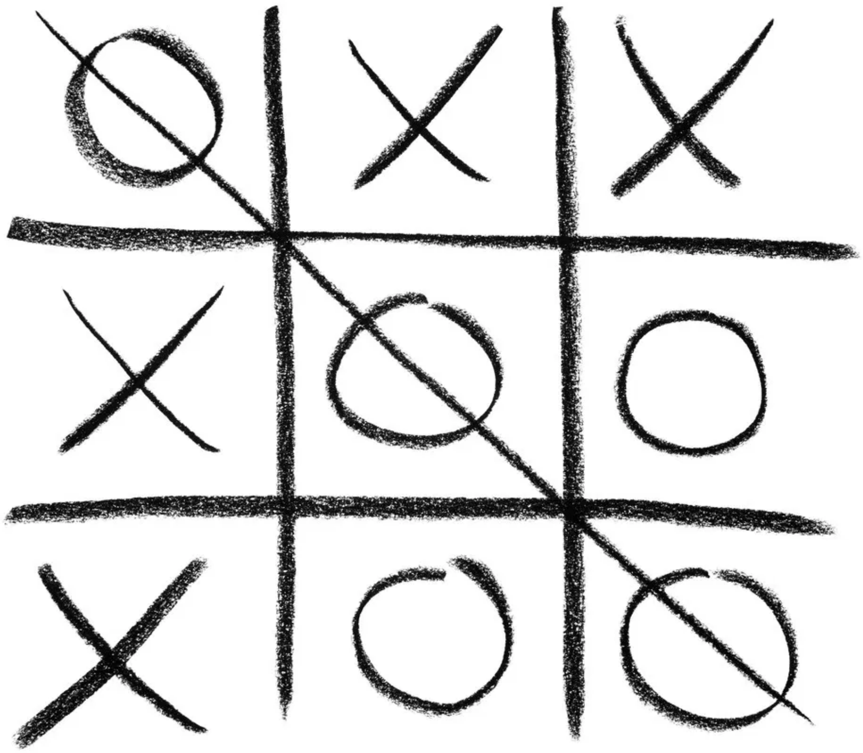
Game: Noughts and crosses

Design

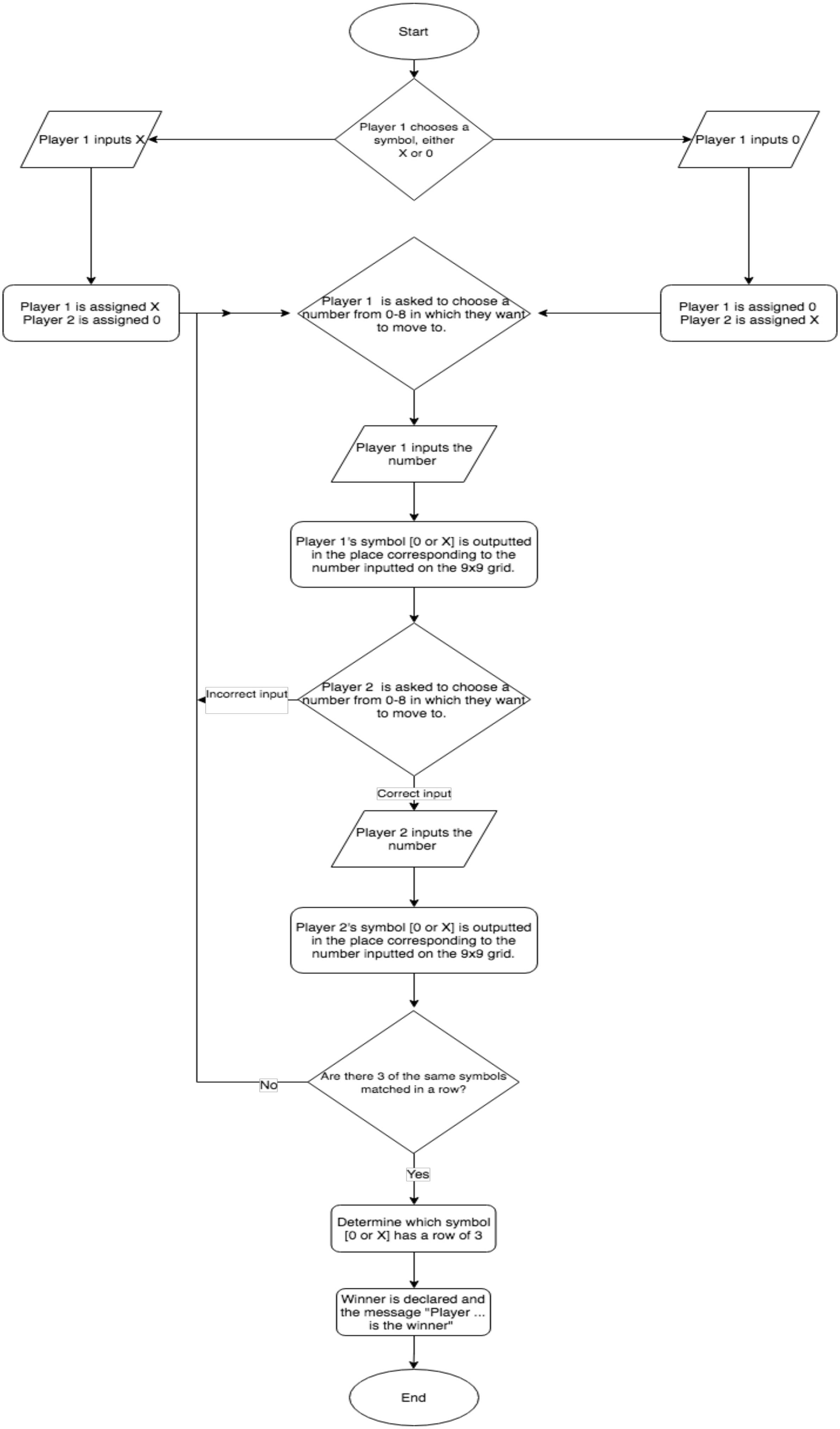
How the game works

For this project, I will be creating a game of noughts and crosses on python. How it works is that when the program is first loaded, a game board is drawn up, a 9x9 square to be precise. It will then ask the user a question as to either selecting X or 0. Once the user inputs either the X or 0, player 2's symbol will be determined automatically. For example, if player 1 selects X, then player 2 will be automatically allocated the 0 to play with and vice versa.

Once the symbol in which they are playing with are selected [X or 0], player 1 will be asked to input a number from 0-8 because this number will determine which square they want to place their chosen symbol in. Then the result will be outputted in the allocated place on the 9x9 grid. The program will then continue until a winner is able to achieve three of their symbols in a row and the message ''Player [1 or 2] is the winner!'' will be outputted from the program.



Flowchart


Start
Player 1 chooses a symbol, either X or 0
Player 1 inputs X
Player 1 inputs 0
Player 1 is assigned X
Player 2 is assigned 0
Player 1 is assigned 0
Player 2 is assigned X
Player 1 is asked to choose a number from 0-8 in which they want to move to.
Player 1 inputs the number
Player 1's symbol [0 or X] is outputted in the place corresponding to the number inputted on the 9x9 grid.
Player 2 is asked to choose a number from 0-8 in which they want to move to.
Incorrect input
Correct input
Player 2 inputs the number
Player 2's symbol [0 or X] is outputted in the place corresponding to the number inputted on the 9x9 grid.
Are there 3 of the same symbols matched in a row?
No
Yes
Determine which symbol [0 or X] has a row of 3
Winner is declared and the message "Player ... is the winner"
End

Variables used

In this program I would use the following variables with specific purposes:

board = This variable defines the empty spaces that the user can select shown as '_'. This then gets multiplied by 9 which allow for it to be indexed by python, where the user can choose a position from 0-8.

player1 = This is the variable which is where player 1 defines what symbol they are, either X or 0. This is the symbol in which player 1 will be playing with.

player2 = This is the variable that is defined by the opposite symbol that player 1 selects. This is the symbol in which player 2 will be playing with.

winner = This is the variable that determines whether the loop for players to play will continue on or not, if winner is equal to 1 then the while loop will break and declare a winner, and if it is 0 it will continue on. Winner will also turn to 1 when it is a draw.

print_board = This routine prints out the board after each player chooses their move. When a player chooses a grid position, the empty space becomes replaced with the player's marker.

Valid = The variable defines the valid inputs the user can choose from to input their symbol into the grid

Goes = This variable counts the number of turns both player 1 and 2 have had combined and increases by 1 after each valid input. The reason for counting the number of goes is how the program identifies when all of the grid positions have been taken up. If the number of goes is equal to 9 and there has not been any winning matches found, it means that it is a draw and the game will end

Chosenpositions = This variable contains a list of all of the chosen positions of each player which gets added to after each valid input. If a player tries to choose a position where the other player has already placed their marker, the game will recognize it and tell them that the position has already been taken and that they cannot choose it.

X = This variable contains the chosen position of player 1

Y = This variable contains the chosen position of player 2

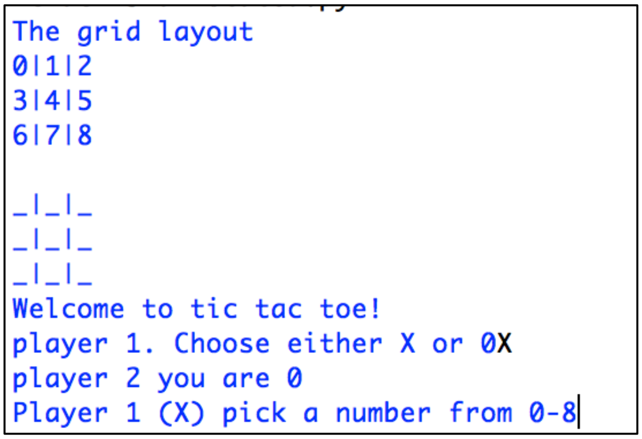
Test plan (Before carrying out testing)

Before testing the code that I made, I created the following test plan to test my game program.

Test number	What to do	Given input	Expected output	Actual output
1	Run the program and wait for the welcome message where the user will be required to choose a symbol	X	Player 2 will be assigned the symbol 0	
2	Run the program and wait for the welcome message where the user will be required to choose a symbol	0	Player 2 will be assigned the symbol X	
3	Run the program and wait for the welcome message where the user will choose their symbol	Z	Error	
4	Run the game until Player X wins	X,0,1,4,2,8	Player X wins!	
5	Run the program until Player 0 wins	0,0,1,4,2,8	Player 0 wins!	
6	Run the program and game until no one wins	X,0,1,2,3,4,5,6,7	It is a draw	
7	Run the program and enter the same position as the last player	X,0,0	That position has already been taken!	

Test plan results (After carrying out testing)

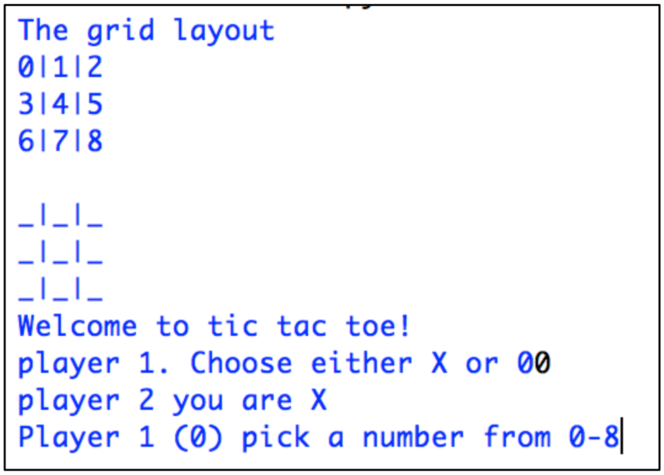
Test number	What to do	Expected input	Expected output	Actual output
1	Run the program and wait for the welcome message where the user will be required to choose a symbol	X	Player 2 will be assigned the symbol 0	“Player 2 you are 0”



```
The grid layout
0|1|2
3|4|5
6|7|8

_|_|_
_|_|_
_|_|_
Welcome to tic tac toe!
player 1. Choose either X or 0X
player 2 you are 0
Player 1 (X) pick a number from 0-8
```

Test Passed

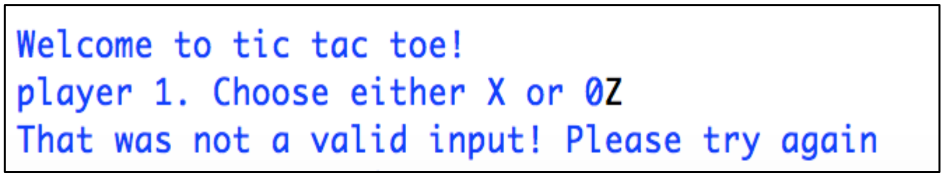
Test number	What to do	Expected input	Expected output	Actual output
2	Run the program and wait for the welcome message where the user will be required to choose a symbol	0	Player 2 will be assigned the symbol X	“Player 2 you are X”



```
The grid layout
0|1|2
3|4|5
6|7|8

_|_|_
_|_|_
_|_|_
Welcome to tic tac toe!
player 1. Choose either X or 00
player 2 you are X
Player 1 (0) pick a number from 0-8
```

Test Passed

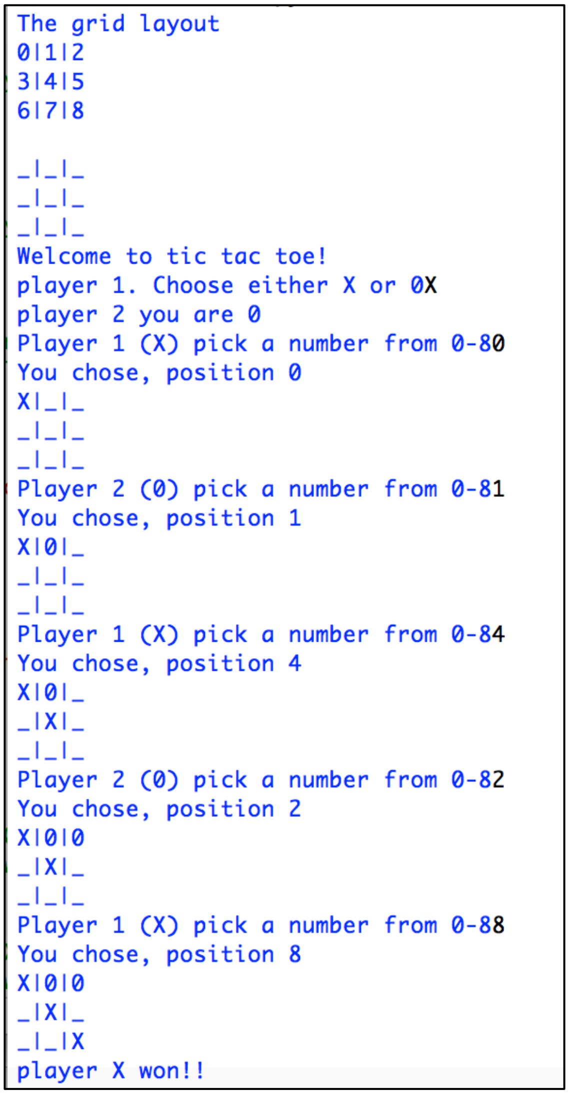
Test number	What to do	Expected input	Expected output	Actual output
3	Run the program and wait for the welcome message where the user will be required to choose a symbol	Z	‘That is not a valid input!’	An error message is outputted: “That was not a valid input! Please try again”



```
Welcome to tic tac toe!
player 1. Choose either X or 0Z
That was not a valid input! Please try again
```

Test Passed

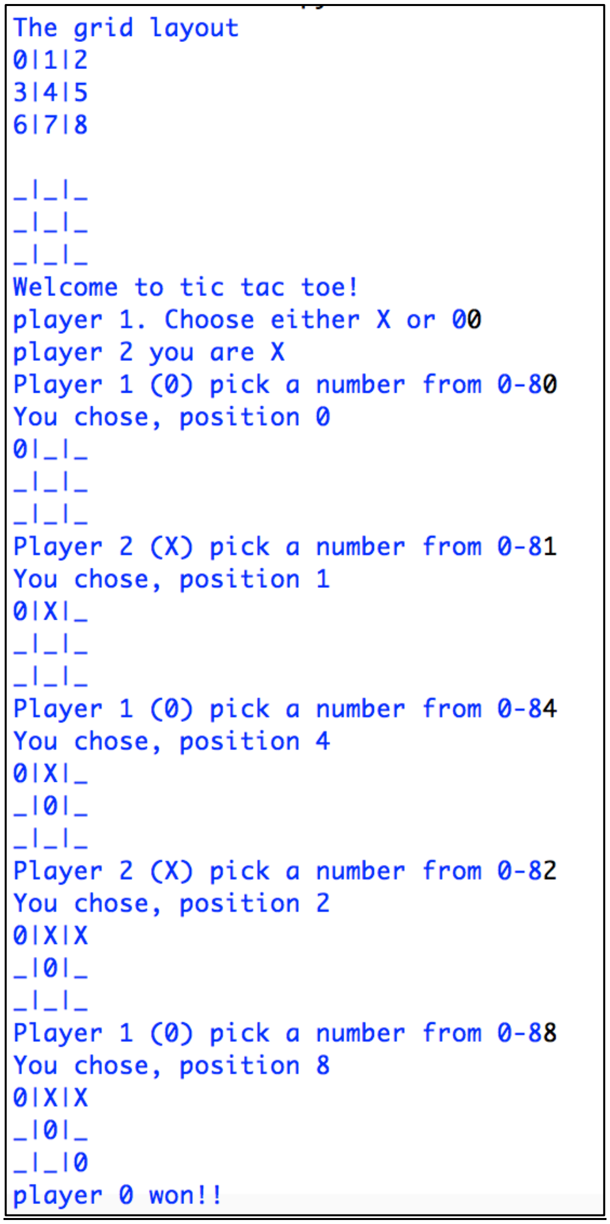
Test number	What to do	Expected input	Expected output	Actual output
4	Run the game until Player X wins	X,0,1,4,2,8	Player X wins!	‘Player X Won!!’



```
The grid layout
0|1|2
3|4|5
6|7|8

_|_|_
_|_|_
_|_|_
Welcome to tic tac toe!
player 1. Choose either X or 0X
player 2 you are 0
Player 1 (X) pick a number from 0-80
You chose, position 0
X|_|_
_|_|_
_|_|_
Player 2 (0) pick a number from 0-81
You chose, position 1
X|0|_
_|_|_
_|_|_
Player 1 (X) pick a number from 0-84
You chose, position 4
X|0|_
_|X|_
_|_|_
Player 2 (0) pick a number from 0-82
You chose, position 2
X|0|0
_|X|_
_|_|_
Player 1 (X) pick a number from 0-88
You chose, position 8
X|0|0
_|X|_
_|_|X
player X won!!
```

Test Passed

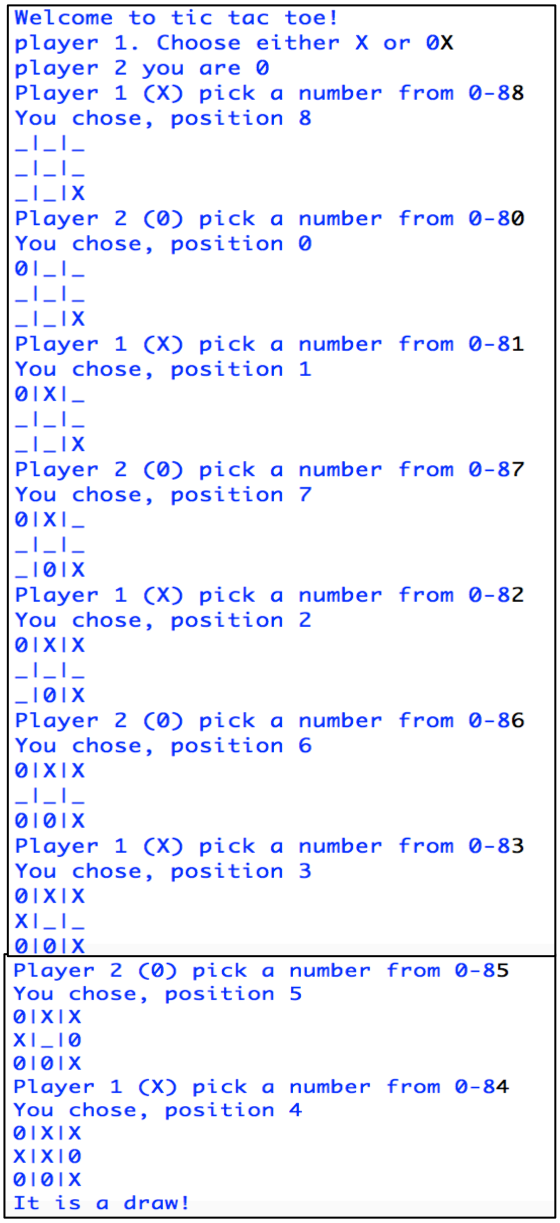
Test number	What to do	Expected input	Expected output	Actual output
5	Run the program until Player 0 wins	0,0,1,4,2,8	Player 0 wins!	‘Player 0 won!!’



```
The grid layout
0|1|2
3|4|5
6|7|8

_|_|_
_|_|_
_|_|_
Welcome to tic tac toe!
player 1. Choose either X or 00
player 2 you are X
Player 1 (0) pick a number from 0-80
You chose, position 0
0|_|_
_|_|_
_|_|_
Player 2 (X) pick a number from 0-81
You chose, position 1
0|X|_
_|_|_
_|_|_
Player 1 (0) pick a number from 0-84
You chose, position 4
0|X|_
_|0|_
_|_|_
Player 2 (X) pick a number from 0-82
You chose, position 2
0|X|X
_|0|_
_|_|_
Player 1 (0) pick a number from 0-88
You chose, position 8
0|X|X
_|0|_
_|_|0
player 0 won!!
```

Test Passed

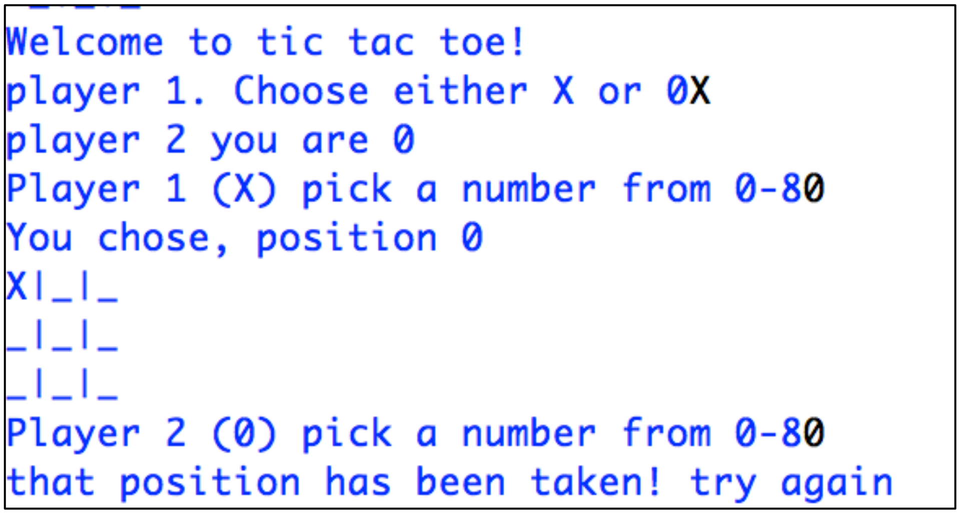
Test number	What to do	Expected input	Expected output	Actual output
6	Run the program and game until no one wins	X,8,0,1,7,2,6,3,5,4	It is a draw	A message is outputted stating: ‘It is a draw!’



```
Welcome to tic tac toe!
player 1. Choose either X or 0X
player 2 you are 0
Player 1 (X) pick a number from 0-88
You chose, position 8
_|_|_
_|_|_
_|_|X
Player 2 (0) pick a number from 0-80
You chose, position 0
0|_|_
_|_|_
_|_|X
Player 1 (X) pick a number from 0-81
You chose, position 1
0|X|_
_|_|_
_|_|X
Player 2 (0) pick a number from 0-87
You chose, position 7
0|X|_
_|_|_
_|0|X
Player 1 (X) pick a number from 0-82
You chose, position 2
0|X|X
_|_|_
_|0|X
Player 2 (0) pick a number from 0-86
You chose, position 6
0|X|X
_|_|_
0|0|X
Player 1 (X) pick a number from 0-83
You chose, position 3
0|X|X
X|_|_
0|0|X
Player 2 (0) pick a number from 0-85
You chose, position 5
0|X|X
X|_|0
0|0|X
Player 1 (X) pick a number from 0-84
You chose, position 4
0|X|X
X|X|0
0|0|X
It is a draw!
```

Test Passed

Test number	What to do	Expected input	Expected output	Actual output
7	Run the program and enter the same position as the last player	X,0,0	That position has already been taken!	The code outputted ‘That position has been taken! Try again



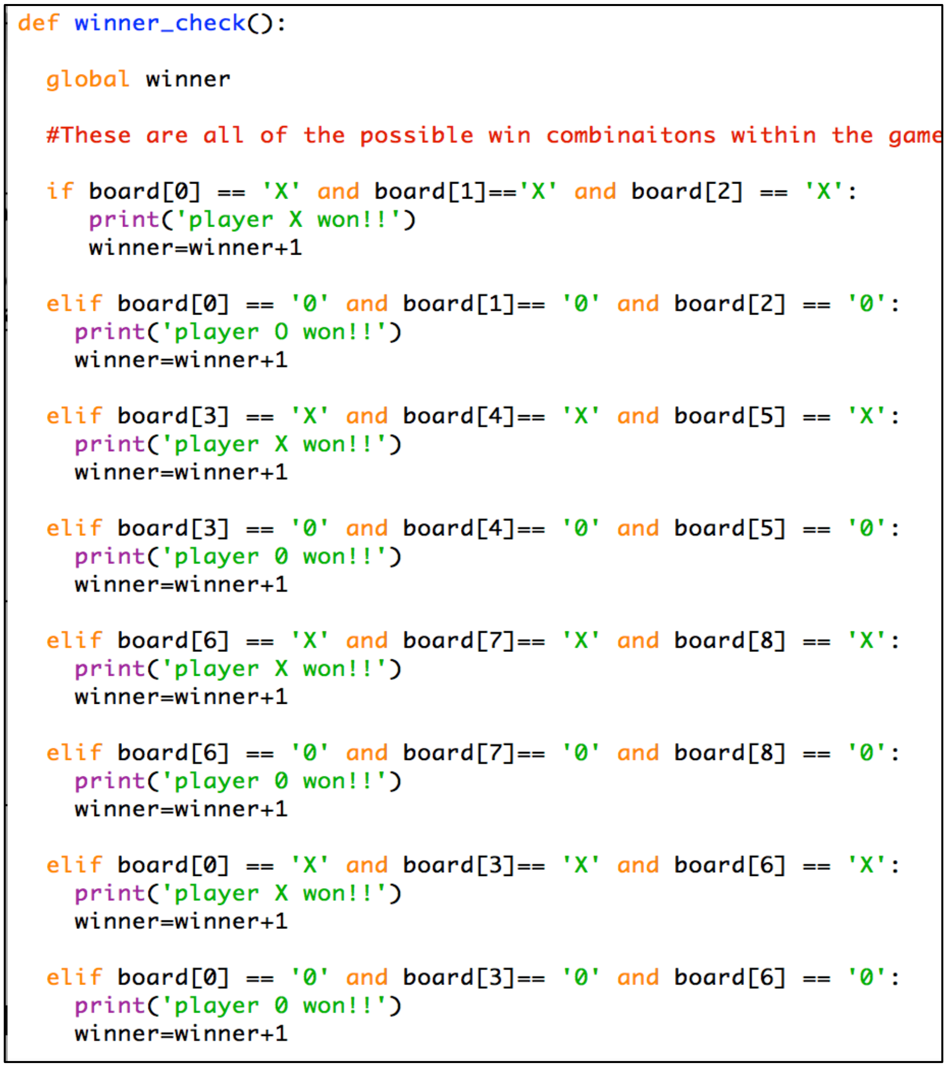
```
Welcome to tic tac toe!
player 1. Choose either X or 0X
player 2 you are 0
Player 1 (X) pick a number from 0-80
You chose, position 0
X|_|_
_|_|_
_|_|_
Player 2 (0) pick a number from 0-80
that position has been taken! try again
```

Test Passed

Evaluation of the results

Overall my code that I created has been able to pass all of the tests that I had created. However, it does not mean that it is perfect since my code consists of 219 lines and if I was given more time, I am convinced that this could be a lot shorter. Here are some potential ways that my code could be cut down:

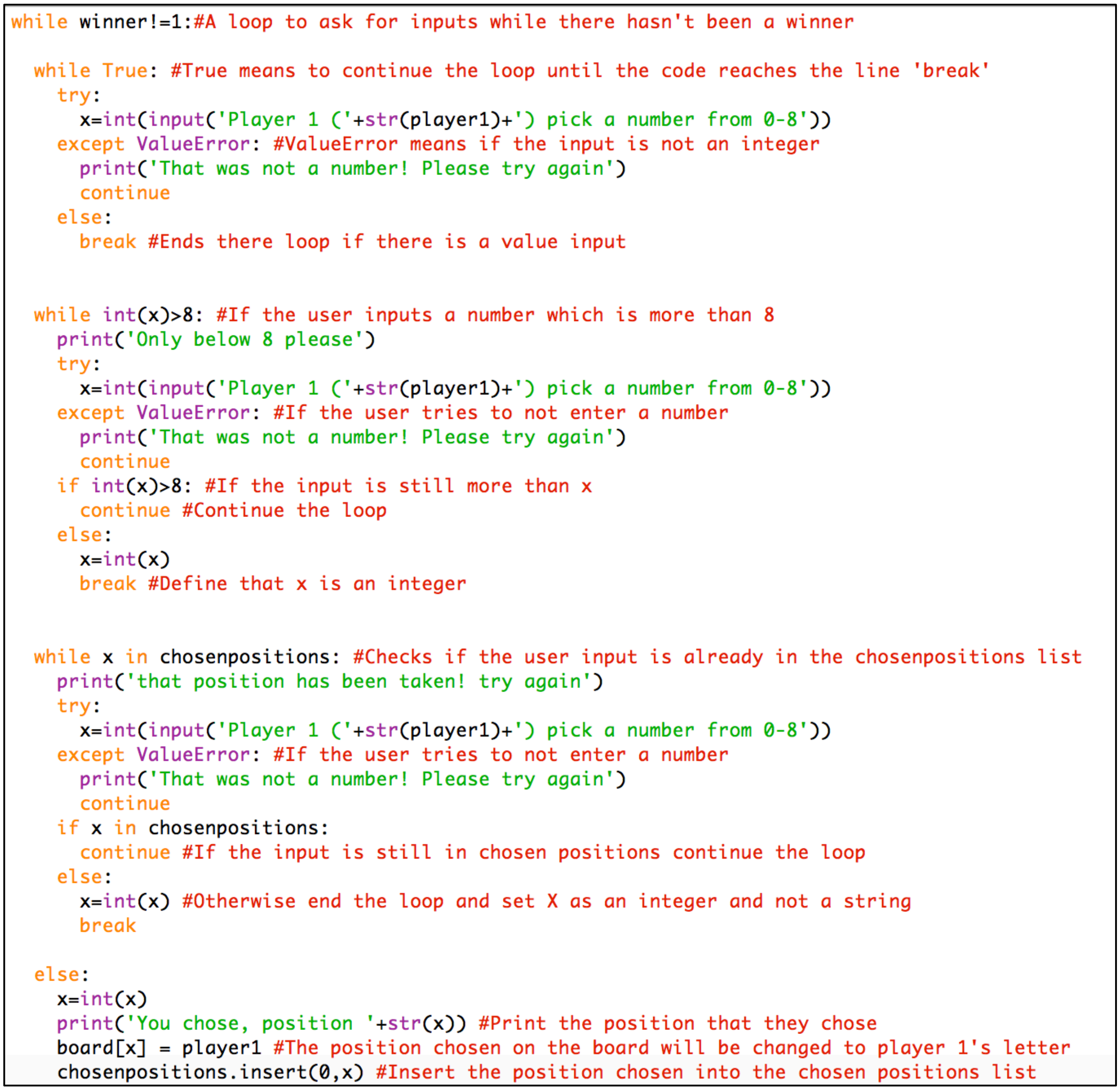
1. **Develop an efficient way of finding a winner**
   Currently the only way to find a winner is to manually define what a win is by checking the grid number positions. This is the longest part of the code and is repeated constantly, as seen in the screenshot below. The code of winner combinations consists 73 lines and could be significantly cut down.



```
def winner_check():

  global winner

  #These are all of the possible win combinaitons within the game

  if board[0] == 'X' and board[1]=='X' and board[2] == 'X':
     print('player X won!!')
     winner=winner+1

  elif board[0] == '0' and board[1]== '0' and board[2] == '0':
    print('player O won!!')
    winner=winner+1

  elif board[3] == 'X' and board[4]== 'X' and board[5] == 'X':
    print('player X won!!')
    winner=winner+1

  elif board[3] == '0' and board[4]== '0' and board[5] == '0':
    print('player 0 won!!')
    winner=winner+1

  elif board[6] == 'X' and board[7]== 'X' and board[8] == 'X':
    print('player X won!!')
    winner=winner+1

  elif board[6] == '0' and board[7]== '0' and board[8] == '0':
    print('player 0 won!!')
    winner=winner+1

  elif board[0] == 'X' and board[3]== 'X' and board[6] == 'X':
    print('player X won!!')
    winner=winner+1

  elif board[0] == '0' and board[3]== '0' and board[6] == '0':
    print('player 0 won!!')
    winner=winner+1
```

2. **Develop a way where code is not repeated for each player**

For each player to have their turn, my code is repeated twice and is exactly the same, except for the player variable. Player 1 is variable X and player 2 is variable Y. To improve the code, I could find a way to only have this portion of code repeated once and have each player alternate when it is their turn. In total, the whole section of code for one player consists of 46 lines and therefore if I was given more time, I am convinced that this could be more efficient.



```
while winner!=1:#A loop to ask for inputs while there hasn't been a winner

  while True: #True means to continue the loop until the code reaches the line 'break'
    try:
      x=int(input('Player 1 ('+str(player1)+') pick a number from 0-8'))
    except ValueError: #ValueError means if the input is not an integer
      print('That was not a number! Please try again')
      continue
    else:
      break #Ends there loop if there is a value input


  while int(x)>8: #If the user inputs a number which is more than 8
    print('Only below 8 please')
    try:
      x=int(input('Player 1 ('+str(player1)+') pick a number from 0-8'))
    except ValueError: #If the user tries to not enter a number
      print('That was not a number! Please try again')
      continue
    if int(x)>8: #If the input is still more than x
      continue #Continue the loop
    else:
      x=int(x)
      break #Define that x is an integer


  while x in chosenpositions: #Checks if the user input is already in the chosenpositions list
    print('that position has been taken! try again')
    try:
      x=int(input('Player 1 ('+str(player1)+') pick a number from 0-8'))
    except ValueError: #If the user tries to not enter a number
      print('That was not a number! Please try again')
      continue
    if x in chosenpositions:
      continue #If the input is still in chosen positions continue the loop
    else:
      x=int(x) #Otherwise end the loop and set X as an integer and not a string
      break

  else:
    x=int(x)
    print('You chose, position '+str(x)) #Print the position that they chose
    board[x] = player1 #The position chosen on the board will be changed to player 1's letter
    chosenpositions.insert(0,x) #Insert the position chosen into the chosen positions list
```

Documentation

Hardware requirements

According to the python website, below are the minimum requirements needed to run python on your computer:

- A keyboard
- A mouse
- A screen
- A CPU with minimum of 1Ghz
- 1GB of RAM
- A hard drive with minimum of 1GB of space for python installation

Software requirements

Also according to the python website, below are the minimum requirements needed to run python on your computer:

- Windows Vista, Mac OS 9 or Linux Operating system
- Python 2.0 or above

User Guide

How to prepare to play Noughts and Crosses?

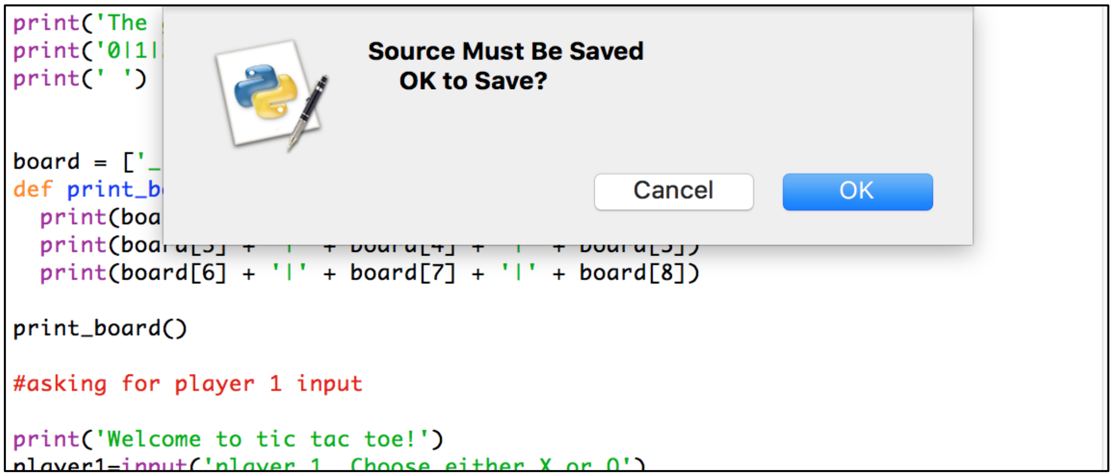
Make sure you have copied Simpleg1.py file to your chosen folder/directory of your hard drive or USB memory stick.

In order to prepare this game or open this game in Python, follow the steps:

- From desktop or start menu, select python 3.6 (or IDLE Python 3.6 32- bit)
- Click the File menu and Open option and select Tic Tac v2.py file from your chosen directory/folder
- Now you can see source code of the game program.

How to run Noughts and Crosses?

When your game program's source code is available on the screen, press F5 from keyboard or select Run Module/F5 from Run menu to run the game program.



```
print('The
print('0|1|
print(' ')


board = ['_
def print_b
  print(boa
  print(board[3] + '|' + board[4] + '|' + board[5])
  print(board[6] + '|' + board[7] + '|' + board[8])

print_board()

#asking for player 1 input

print('Welcome to tic tac toe!')
player1=input('player 1, Choose either X or O')
```

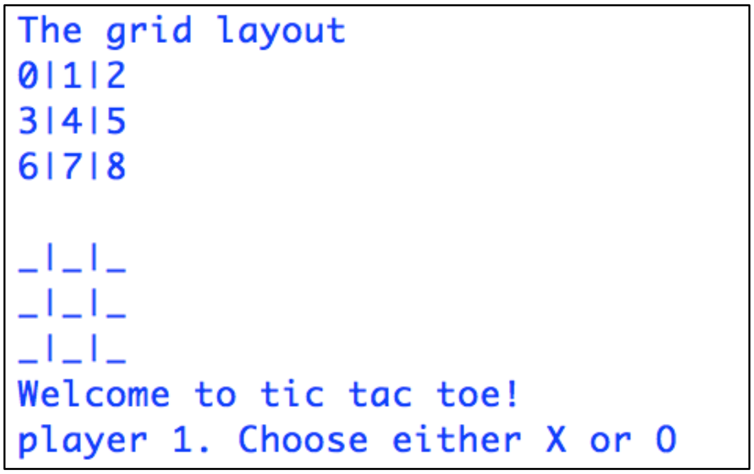
Source Must Be Saved
OK to Save?

Python may ask you that the source must be saved. Click OK and the program will be opened

How to play Noughts and Crosses?

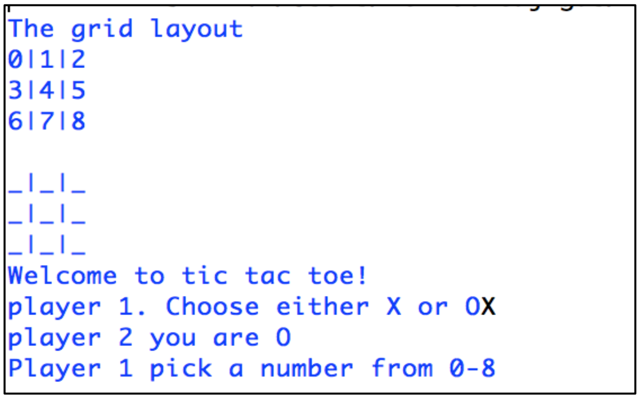
In order to play the game, follow the steps:

- Open TicTac v2.py file to the screen
- Press F5 from keyboard or select Run Module/F5 from Run menu
- The screen should look like the following:



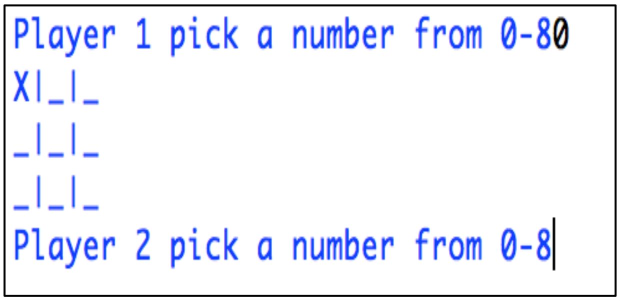
```
The grid layout
0|1|2
3|4|5
6|7|8

_|_|_
_|_|_
_|_|_
Welcome to tic tac toe!
player 1. Choose either X or O
```

1. Player 1 should input either X or 0, and once entered player 2’s symbol will be determined



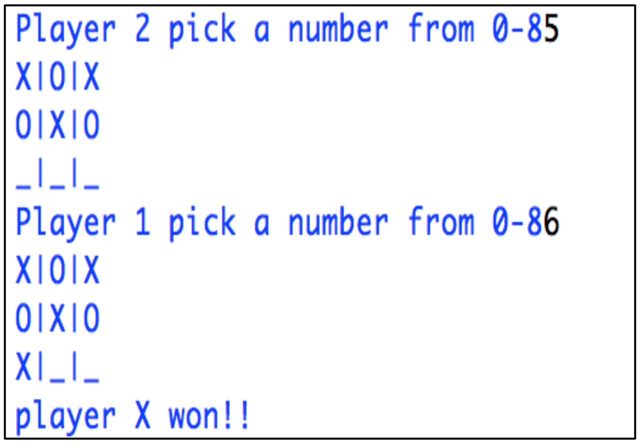
```
The grid layout
0|1|2
3|4|5
6|7|8

_|_|_
_|_|_
_|_|_
Welcome to tic tac toe!
player 1. Choose either X or OX
player 2 you are O
Player 1 pick a number from 0-8
```

2. The grid layout is seen above. Player 1 will have to first turn to input a number from 0 to 8 and their symbol will be outputted on the grid. Player 2 will have their turn and then this cycle continues, until a winner is found.



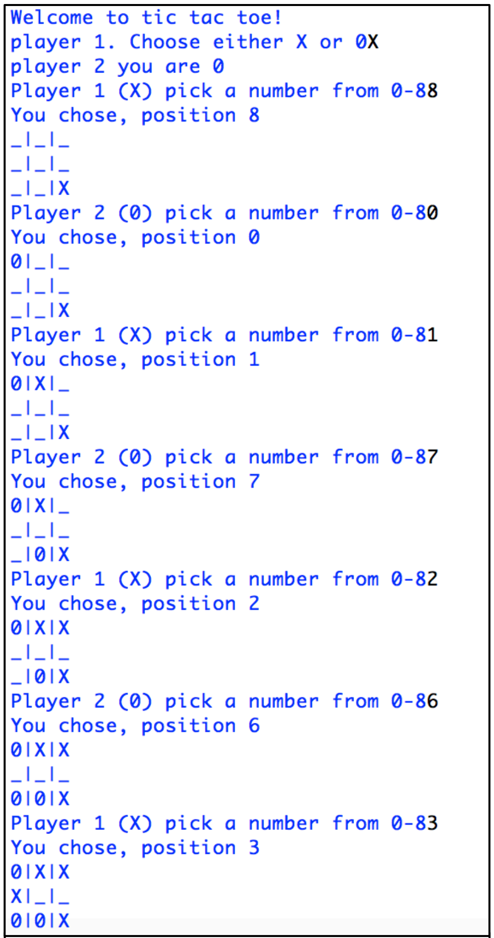
```
Player 1 pick a number from 0-80
X|_|_
_|_|_
_|_|_
Player 2 pick a number from 0-8
```

3. When the same symbol is displayed 3 times in a row, a winner is found.

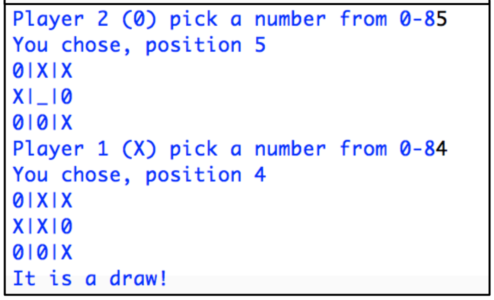


```
Player 2 pick a number from 0-85
X|O|X
O|X|O
_|_|_
Player 1 pick a number from 0-86
X|O|X
O|X|O
X|_|_
player X won!!
```

However, if all symbols have been inputted on the graph, and there have not been a match of 3 symbols in a row, then it is a draw as seen on the right.



```
Welcome to tic tac toe!
player 1. Choose either X or 0X
player 2 you are 0
Player 1 (X) pick a number from 0-88
You chose, position 8
_|_|_
_|_|_
_|_|X
Player 2 (0) pick a number from 0-80
You chose, position 0
0|_|_
_|_|_
_|_|X
Player 1 (X) pick a number from 0-81
You chose, position 1
0|X|_
_|_|_
_|_|X
Player 2 (0) pick a number from 0-87
You chose, position 7
0|X|_
_|_|_
_|0|X
Player 1 (X) pick a number from 0-82
You chose, position 2
0|X|X
_|_|_
_|0|X
Player 2 (0) pick a number from 0-86
You chose, position 6
0|X|X
_|_|_
0|0|X
Player 1 (X) pick a number from 0-83
You chose, position 3
0|X|X
X|_|_
0|0|X
```



```
Player 2 (0) pick a number from 0-85
You chose, position 5
0|X|X
X|_|0
0|0|X
Player 1 (X) pick a number from 0-84
You chose, position 4
0|X|X
X|X|0
0|0|X
It is a draw!
```

Troubleshooting

1. If you get an error to play this game, what should you do?

If you use recommended Python 3.6, you should not get any error message to run this game. If you use any other version of Python, you may get error message or may not be able to play this game. If you still use Python 3.6 version and get error to run this game, open and run original backup copy. Assume that you have duplicate copy of this game as backup in your hard drive

2. Do you need to save this game after playing?

After playing this game, you should not save this game. Somehow if this program code is changed or modified, during closing this game, a message would come on the screen to save the program file and click on “No” button to close program file without saving

3. How to update/modify this game program?

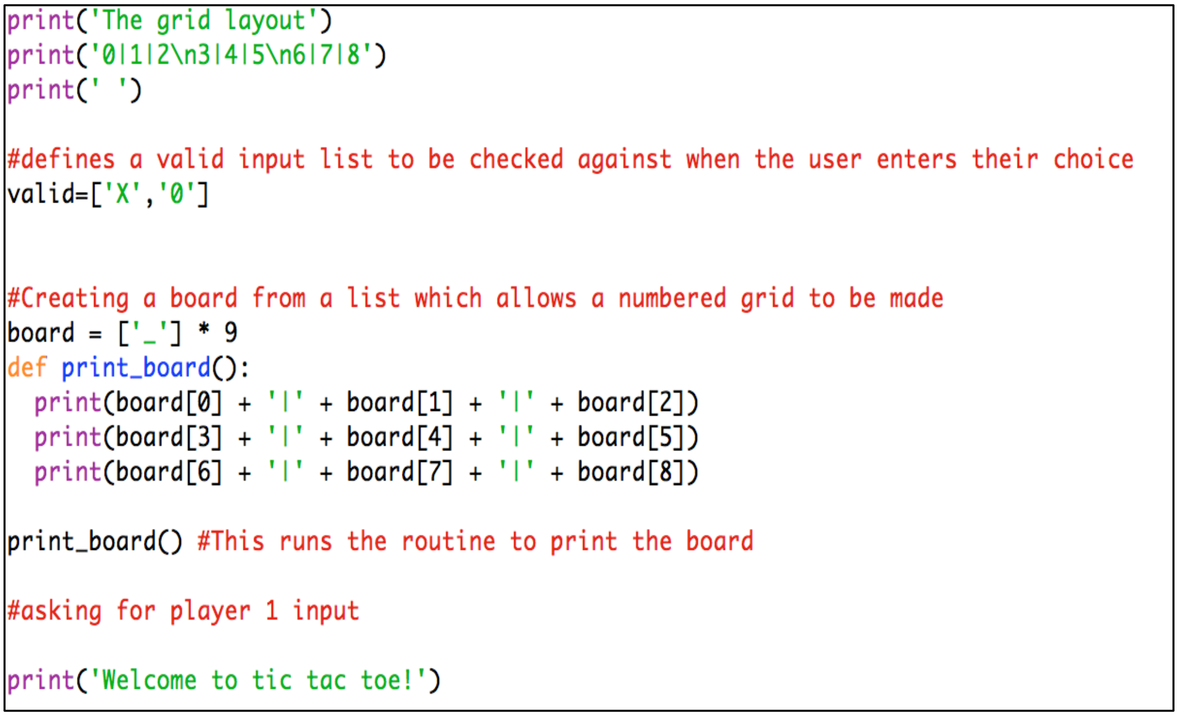
You should be familiar with Python program and its syntax. For further information, visit www.python.org and https://www.w3schools.com/python/

The Final Python Code explained [with comments]

Here I will explain my code in sections, for my python game “Noughts and Crosses” and show how it works

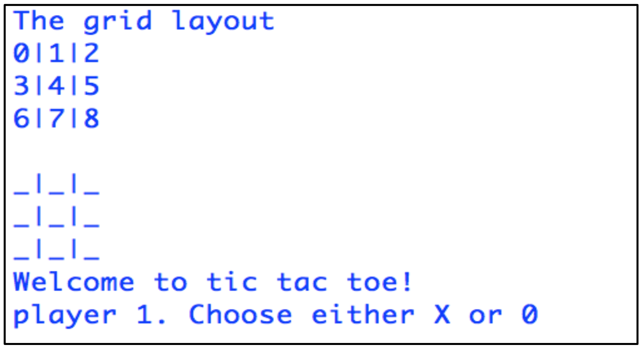
Section 1 - This is where the board is defined and board as a variable is set to ‘_’ to show the user an empty space. It is multiplied by 9 to create a list that can be indexed from 0-8. The board is printed and shown to the user. Indexing allows the user to have the choice to choose a position that will be printed out on the board.

Code:



```
print('The grid layout')
print('0|1|2\n3|4|5\n6|7|8')
print(' ')

#defines a valid input list to be checked against when the user enters their choice
valid=['X','0']


#Creating a board from a list which allows a numbered grid to be made
board = ['_'] * 9
def print_board():
  print(board[0] + '|' + board[1] + '|' + board[2])
  print(board[3] + '|' + board[4] + '|' + board[5])
  print(board[6] + '|' + board[7] + '|' + board[8])

print_board() #This runs the routine to print the board

#asking for player 1 input

print('Welcome to tic tac toe!')
```

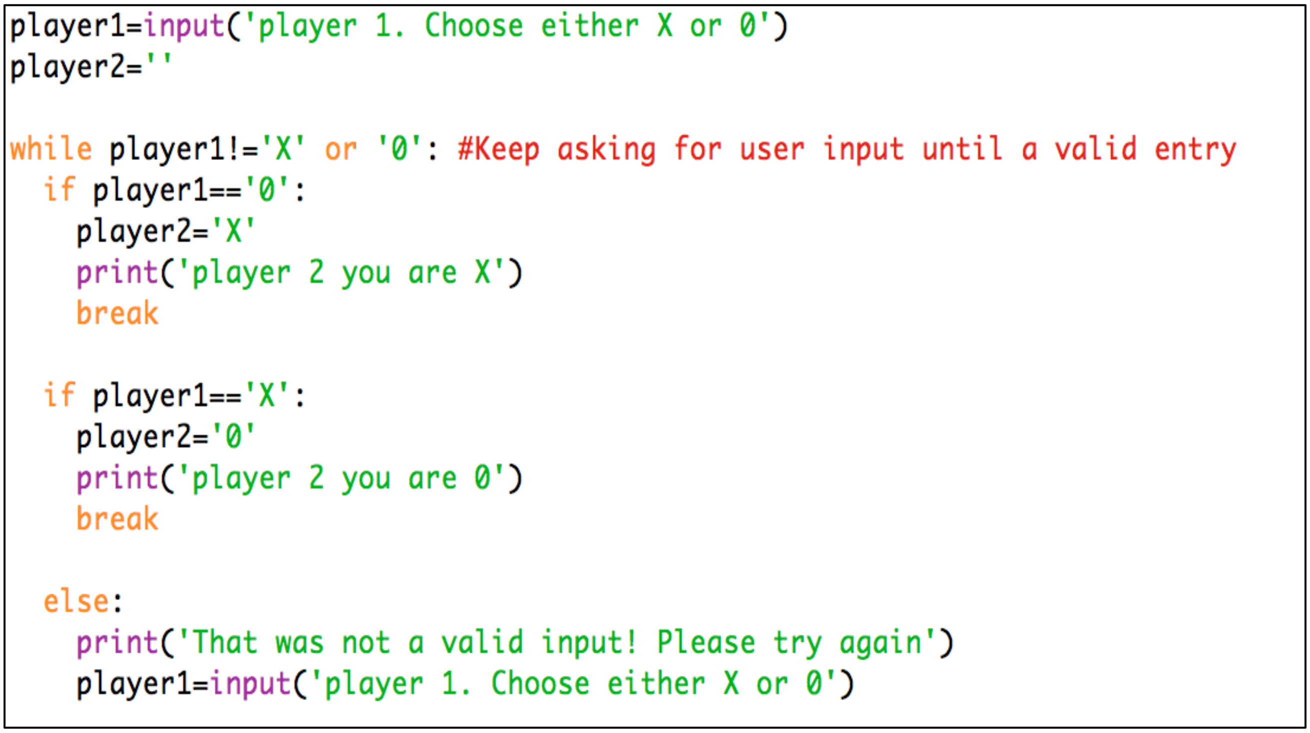
Output:



```
The grid layout
0|1|2
3|4|5
6|7|8

_|_|_
_|_|_
_|_|_
Welcome to tic tac toe!
player 1. Choose either X or 0
```

Section 2 - This is the first user input where the user is asked to choose either X or O. If they don’t put the right input, the code will output saying it was not a valid input!

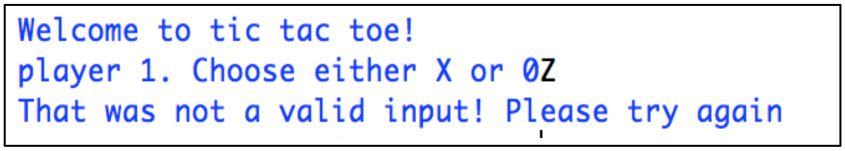
Code:



```
player1=input('player 1. Choose either X or 0')
player2=''

while player1!='X' or '0': #Keep asking for user input until a valid entry
  if player1=='0':
    player2='X'
    print('player 2 you are X')
    break

  if player1=='X':
    player2='0'
    print('player 2 you are 0')
    break

  else:
    print('That was not a valid input! Please try again')
    player1=input('player 1. Choose either X or 0')
```

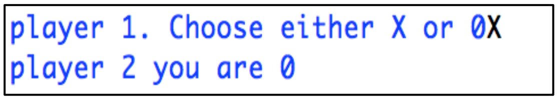
Possible Outputs

Output 1:



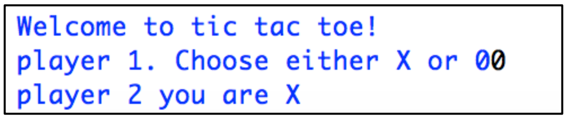
```
Welcome to tic tac toe!
player 1. Choose either X or 0Z
That was not a valid input! Please try again
```

Output 2:



```
player 1. Choose either X or 0X
player 2 you are 0
```

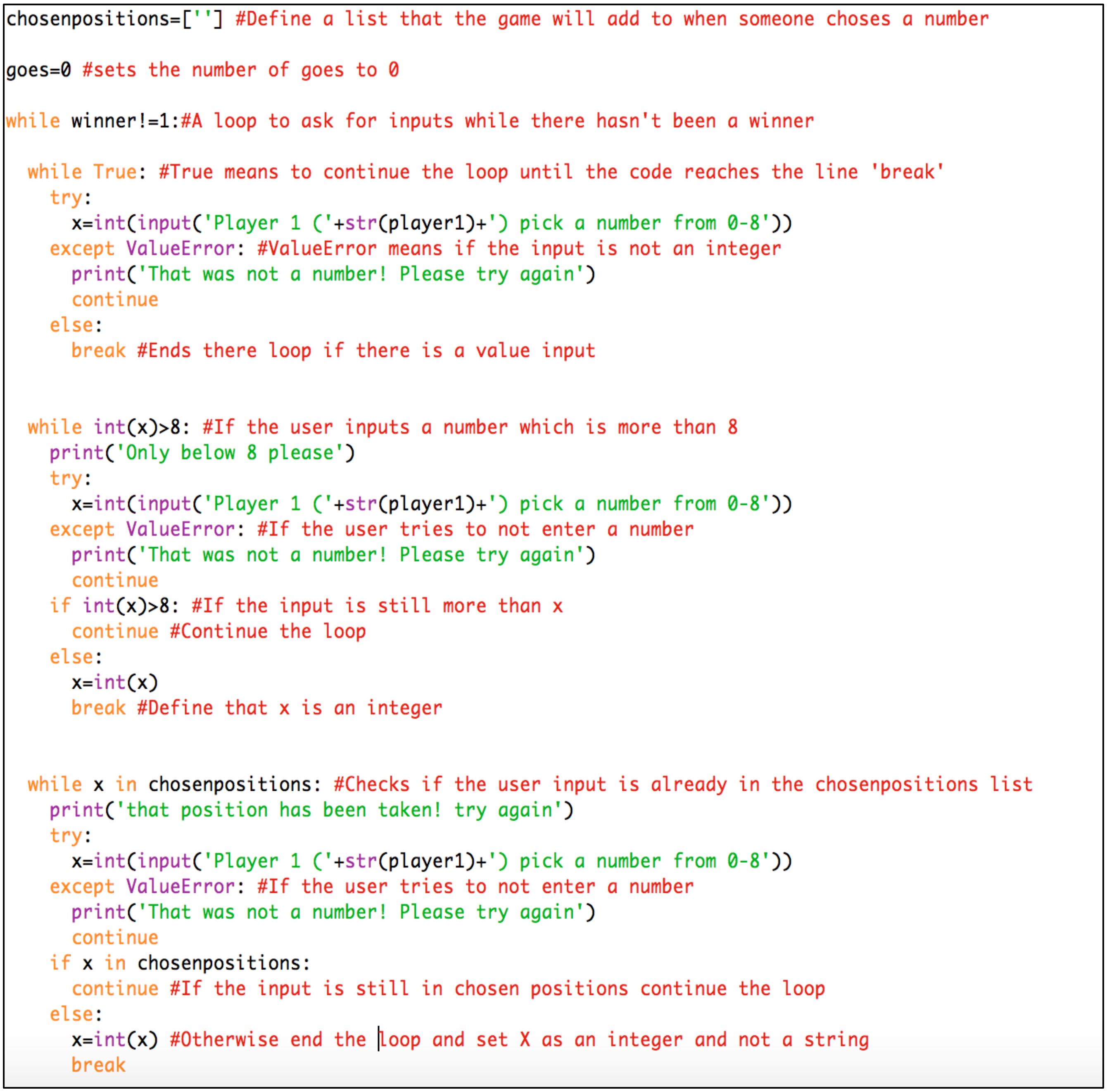
Output 3:



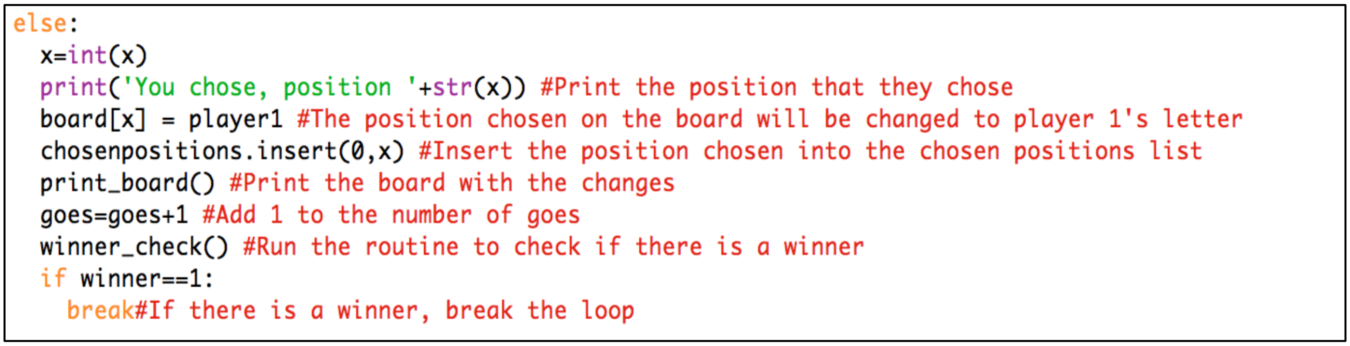
```
Welcome to tic tac toe!
player 1. Choose either X or 00
player 2 you are X
```

Section 3 - The user will be asked to select their position on the grid, and before it gets placed, the user input will need to go through a number of validation while there is no winner. (1) Ensuring it is a number (2) It is less than 8 (3) It hasn't been chosen yet by anyone. This is achieved by the code creating a list and recording to it the positions that have already been selected. Once all of the validation is done and the input is correct, it will print out the position chosen, redefine the board grid number with the player's chosen letter and add the number chosen to the chosenpositions list. It will also then run the winner check routine which is defined before the while loops and will be explained.

Code:



```
chosenpositions=[''] #Define a list that the game will add to when someone choses a number

goes=0 #sets the number of goes to 0

while winner!=1:#A loop to ask for inputs while there hasn't been a winner

  while True: #True means to continue the loop until the code reaches the line 'break'
    try:
      x=int(input('Player 1 ('+str(player1)+') pick a number from 0-8'))
    except ValueError: #ValueError means if the input is not an integer
      print('That was not a number! Please try again')
      continue
    else:
      break #Ends there loop if there is a value input


  while int(x)>8: #If the user inputs a number which is more than 8
    print('Only below 8 please')
    try:
      x=int(input('Player 1 ('+str(player1)+') pick a number from 0-8'))
    except ValueError: #If the user tries to not enter a number
      print('That was not a number! Please try again')
      continue
    if int(x)>8: #If the input is still more than x
      continue #Continue the loop
    else:
      x=int(x)
      break #Define that x is an integer


  while x in chosenpositions: #Checks if the user input is already in the chosenpositions list
    print('that position has been taken! try again')
    try:
      x=int(input('Player 1 ('+str(player1)+') pick a number from 0-8'))
    except ValueError: #If the user tries to not enter a number
      print('That was not a number! Please try again')
      continue
    if x in chosenpositions:
      continue #If the input is still in chosen positions continue the loop
    else:
      x=int(x) #Otherwise end the loop and set X as an integer and not a string
      break
```

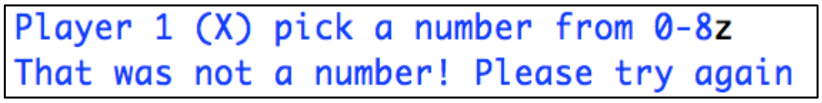


```
else:
  x=int(x)
  print('You chose, position '+str(x)) #Print the position that they chose
  board[x] = player1 #The position chosen on the board will be changed to player 1's letter
  chosenpositions.insert(0,x) #Insert the position chosen into the chosen positions list
  print_board() #Print the board with the changes
  goes=goes+1 #Add 1 to the number of goes
  winner_check() #Run the routine to check if there is a winner
  if winner==1:
    break#If there is a winner, break the loop
```

ChosenPositions is the variable used to define the list that will be added to when a user enters a number. Goes is the variable that defines the number of goes that each Player 1 and 2 have had combined and is increased by 1 after each correct input.

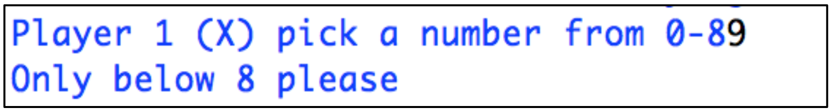
Possible Outputs

Output 1:



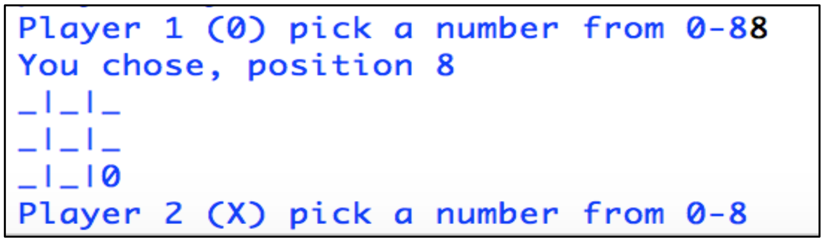
```
Player 1 (X) pick a number from 0-8z
That was not a number! Please try again
```

Output 2:



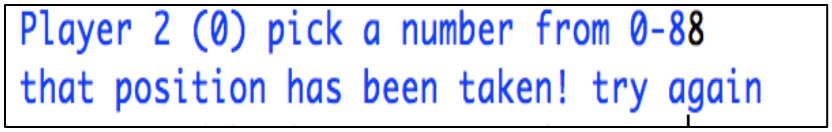
```
Player 1 (X) pick a number from 0-89
Only below 8 please
```

Output 3:



```
Player 1 (0) pick a number from 0-88
You chose, position 8
_|_|_
_|_|_
_|_|0
Player 2 (X) pick a number from 0-8
```

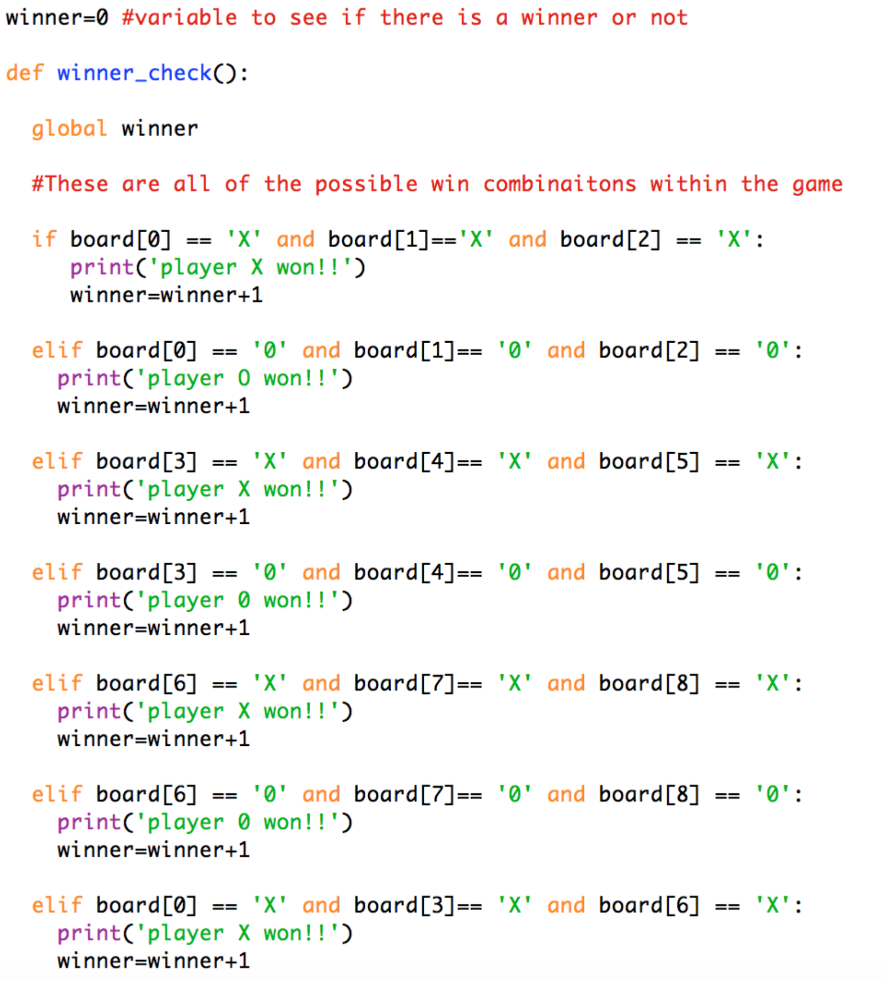
Output 4:



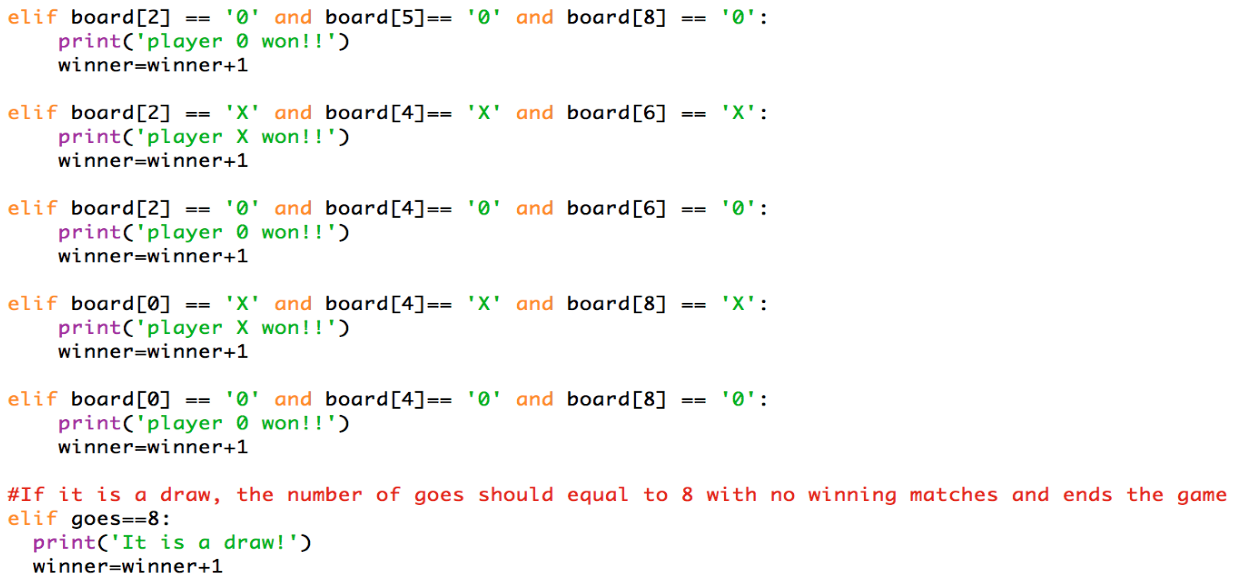
```
Player 2 (0) pick a number from 0-88
that position has been taken! try again
```

Section 4 - This section defines all of the possible combinations for wins within the game. If the game does not find any match, then it will go to a draw if the number of goes is equal to 9

Code:



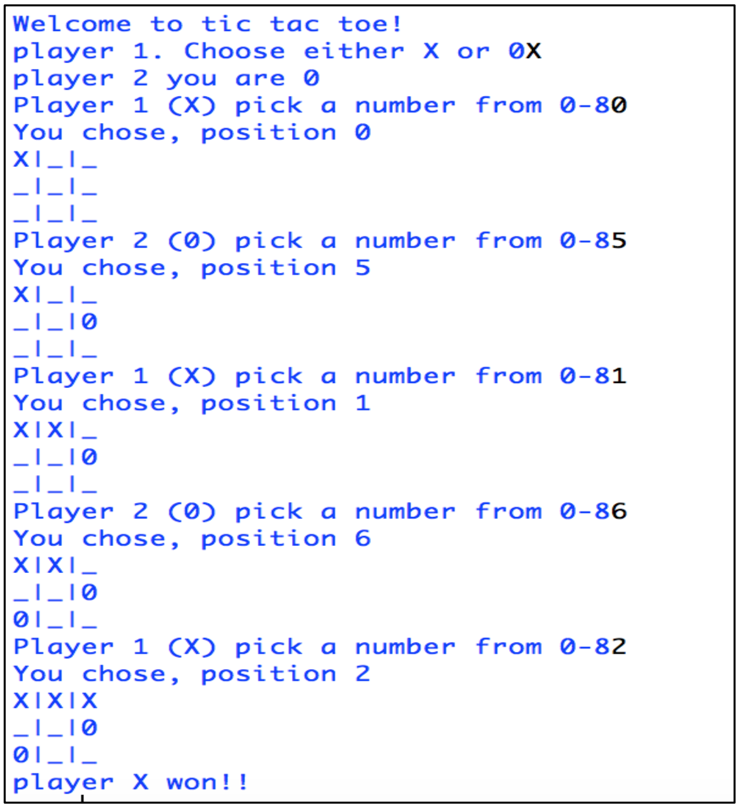
```
winner=0 #variable to see if there is a winner or not

def winner_check():

  global winner

  #These are all of the possible win combinaitons within the game

  if board[0] == 'X' and board[1]=='X' and board[2] == 'X':
     print('player X won!!')
     winner=winner+1

  elif board[0] == '0' and board[1]== '0' and board[2] == '0':
    print('player O won!!')
    winner=winner+1

  elif board[3] == 'X' and board[4]== 'X' and board[5] == 'X':
    print('player X won!!')
    winner=winner+1

  elif board[3] == '0' and board[4]== '0' and board[5] == '0':
    print('player 0 won!!')
    winner=winner+1

  elif board[6] == 'X' and board[7]== 'X' and board[8] == 'X':
    print('player X won!!')
    winner=winner+1

  elif board[6] == '0' and board[7]== '0' and board[8] == '0':
    print('player 0 won!!')
    winner=winner+1

  elif board[0] == 'X' and board[3]== 'X' and board[6] == 'X':
    print('player X won!!')
    winner=winner+1
```



```
  elif board[2] == '0' and board[5]== '0' and board[8] == '0':
      print('player 0 won!!')
      winner=winner+1

  elif board[2] == 'X' and board[4]== 'X' and board[6] == 'X':
      print('player X won!!')
      winner=winner+1

  elif board[2] == '0' and board[4]== '0' and board[6] == '0':
      print('player 0 won!!')
      winner=winner+1

  elif board[0] == 'X' and board[4]== 'X' and board[8] == 'X':
      print('player X won!!')
      winner=winner+1

  elif board[0] == '0' and board[4]== '0' and board[8] == '0':
      print('player 0 won!!')
      winner=winner+1

  #If it is a draw, the number of goes should equal to 8 with no winning matches and ends the game
  elif goes==8:
    print('It is a draw!')
    winner=winner+1
```

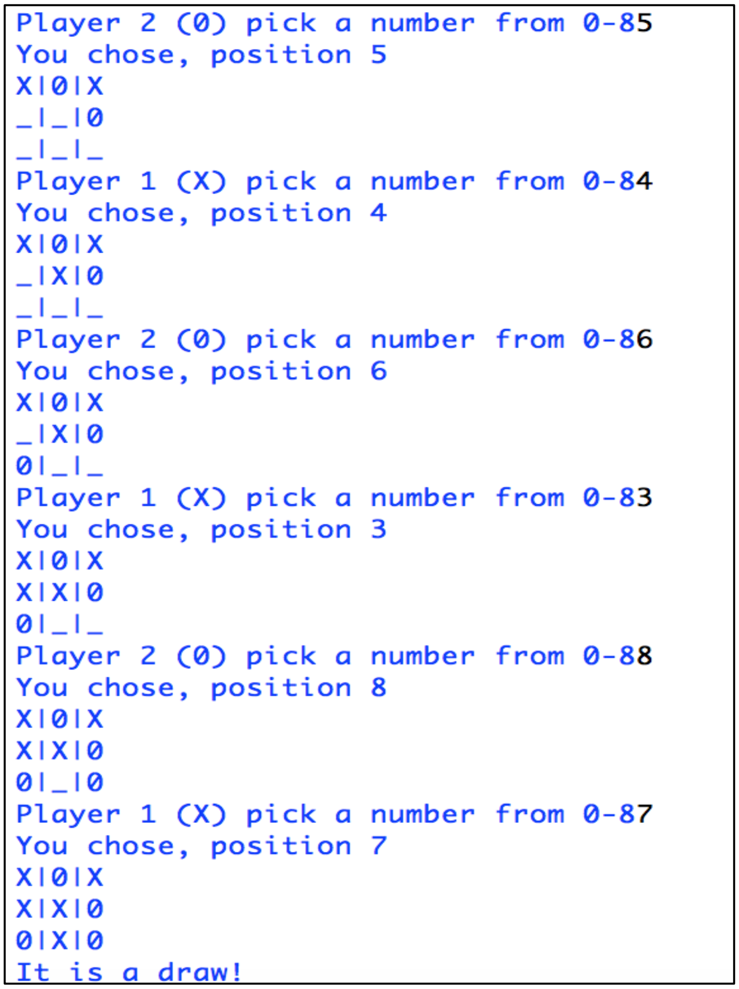
Possible Outputs

Output 1: For a win:



```
Welcome to tic tac toe!
player 1. Choose either X or 0X
player 2 you are 0
Player 1 (X) pick a number from 0-80
You chose, position 0
X|_|_
_|_|_
_|_|_
Player 2 (0) pick a number from 0-85
You chose, position 5
X|_|_
_|_|0
_|_|_
Player 1 (X) pick a number from 0-81
You chose, position 1
X|X|_
_|_|0
_|_|_
Player 2 (0) pick a number from 0-86
You chose, position 6
X|X|_
_|_|0
0|_|_
Player 1 (X) pick a number from 0-82
You chose, position 2
X|X|X
_|_|0
0|_|_
player X won!!
```

Output 2: For a draw:



```
Player 2 (0) pick a number from 0-85
You chose, position 5
X|0|X
_|_|0
_|_|_
Player 1 (X) pick a number from 0-84
You chose, position 4
X|0|X
_|X|0
_|_|_
Player 2 (0) pick a number from 0-86
You chose, position 6
X|0|X
_|X|0
0|_|_
Player 1 (X) pick a number from 0-83
You chose, position 3
X|0|X
X|X|0
0|_|_
Player 2 (0) pick a number from 0-88
You chose, position 8
X|0|X
X|X|0
0|_|0
Player 1 (X) pick a number from 0-87
You chose, position 7
X|0|X
X|X|0
0|X|0
It is a draw!
```

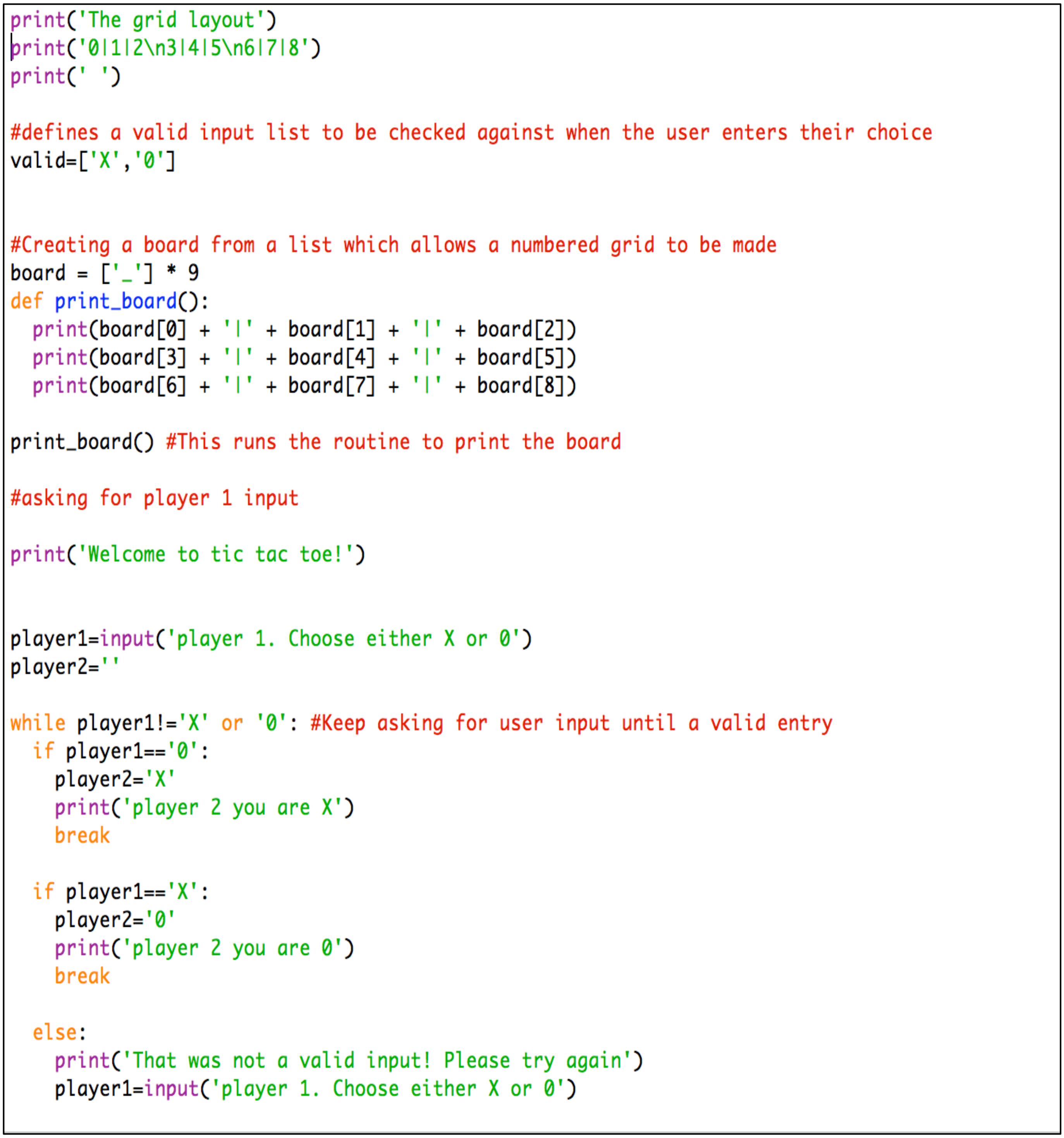
Full Python code with comments.

```
print('The grid layout')
print('0|1|2\n3|4|5\n6|7|8')
print(' ')

#defines a valid input list to be checked against when the user enters their choice
valid=['X','0']


#Creating a board from a list which allows a numbered grid to be made
board = ['_'] * 9
def print_board():
  print(board[0] + '|' + board[1] + '|' + board[2])
  print(board[3] + '|' + board[4] + '|' + board[5])
  print(board[6] + '|' + board[7] + '|' + board[8])

print_board() #This runs the routine to print the board

#asking for player 1 input

print('Welcome to tic tac toe!')


player1=input('player 1. Choose either X or 0')
player2=''

while player1!='X' or '0': #Keep asking for user input until a valid entry
  if player1=='0':
    player2='X'
    print('player 2 you are X')
    break

  if player1=='X':
    player2='0'
    print('player 2 you are 0')
    break

  else:
    print('That was not a valid input! Please try again')
    player1=input('player 1. Choose either X or 0')
```

```
winner=0 #variable to see if there is a winner or not

def winner_check():

  global winner

  #These are all of the possible win combinaitons within the game

  if board[0] == 'X' and board[1]=='X' and board[2] == 'X':
     print('player X won!!')
     winner=winner+1

  elif board[0] == '0' and board[1]== '0' and board[2] == '0':
    print('player O won!!')
    winner=winner+1

  elif board[3] == 'X' and board[4]== 'X' and board[5] == 'X':
    print('player X won!!')
    winner=winner+1

  elif board[3] == '0' and board[4]== '0' and board[5] == '0':
    print('player 0 won!!')
    winner=winner+1

  elif board[6] == 'X' and board[7]== 'X' and board[8] == 'X':
    print('player X won!!')
    winner=winner+1

  elif board[6] == '0' and board[7]== '0' and board[8] == '0':
    print('player 0 won!!')
    winner=winner+1

  elif board[0] == 'X' and board[3]== 'X' and board[6] == 'X':
    print('player X won!!')
    winner=winner+1

  elif board[0] == '0' and board[3]== '0' and board[6] == '0':
    print('player 0 won!!')
    winner=winner+1

  elif board[1] == 'X' and board[4]== 'X' and board[7] == 'X':
      print('player X won!!')
      winner=winner+1
  elif board[1] == '0' and board[4]== '0' and board[7] == '0':
      print('player 0 won!!')
      winner=winner+1

  elif board[2] == 'X' and board[5]== 'X' and board[8] == 'X':
      print('player X won!!')
      winner=winner+1

  elif board[2] == '0' and board[5]== '0' and board[8] == '0':
      print('player 0 won!!')
      winner=winner+1

  elif board[2] == 'X' and board[4]== 'X' and board[6] == 'X':
      print('player X won!!')
      winner=winner+1

  elif board[2] == '0' and board[4]== '0' and board[6] == '0':
      print('player 0 won!!')
      winner=winner+1

  elif board[0] == 'X' and board[4]== 'X' and board[8] == 'X':
      print('player X won!!')
      winner=winner+1

  elif board[0] == '0' and board[4]== '0' and board[8] == '0':
      print('player 0 won!!')
      winner=winner+1

  #If it is a draw, the number of goes should equal to 8 with no winning matches and ends the game
  elif goes==9:
    print('It is a draw!')
    winner=winner+1

chosenpositions=[''] #Define a list that the game will add to when someone choses a number

goes=0 #sets the number of goes to 0

while winner!=1:#A loop to ask for inputs while there hasn't been a winner

  while True: #True means to continue the loop until the code reaches the line 'break'
    try:
      x=int(input('Player 1 ('+str(player1)+') pick a number from 0-8'))
    except ValueError: #ValueError means if the input is not an integer
      print('That was not a number! Please try again')
      continue
    else:
      break #Ends there loop if there is a value input


  while int(x)>8: #If the user inputs a number which is more than 8
    print('Only below 8 please')
    try:
      x=int(input('Player 1 ('+str(player1)+') pick a number from 0-8'))
    except ValueError: #If the user tries to not enter a number
      print('That was not a number! Please try again')
      continue
    if int(x)>8: #If the input is still more than x
      continue #Continue the loop
    else:
      x=int(x)
      break #Define that x is an integer


  while x in chosenpositions: #Checks if the user input is already in the chosenpositions list
    print('that position has been taken! try again')
    try:
      x=int(input('Player 1 ('+str(player1)+') pick a number from 0-8'))
    except ValueError: #If the user tries to not enter a number
      print('That was not a number! Please try again')
      continue
    if x in chosenpositions:
      continue #If the input is still in chosen positions continue the loop
    else:
      x=int(x) #Otherwise end the loop and set X as an integer and not a string
      break

  else:
    x=int(x)
    print('You chose, position '+str(x)) #Print the position that they chose
    board[x] = player1 #The position chosen on the board will be changed to player 1's letter
    chosenpositions.insert(0,x) #Insert the position chosen into the chosen positions list
    print_board() #Print the board with the changes
    goes=goes+1 #Add 1 to the number of goes
    winner_check() #Run the routine to check if there is a winner
    if winner==1:
      break#If there is a winner, break the loop
```

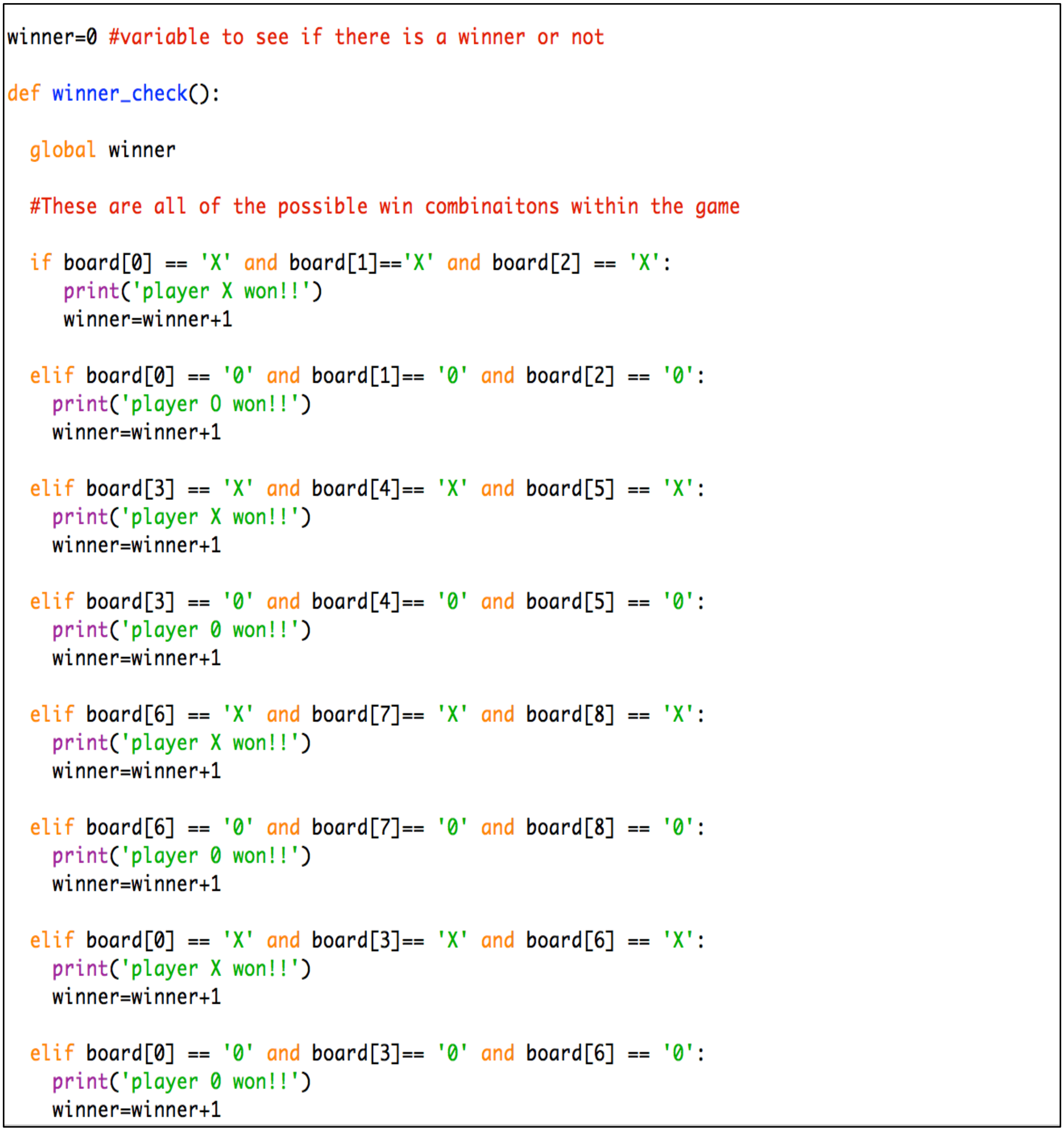
```
while True:
  try:
    y=int(input('Player 2 ('+str(player2)+') pick a number from 0-8'))
  except ValueError:
    print('That was not a number! Please try again')
    continue
  else:
    break


while int(y)>8:
  print('Only below 8 please')
  try:
    y=int(input('Player 1 ('+str(player2)+') pick a number from 0-8'))
  except ValueError:
    print('That was not a number! Please try again')
    continue
  if int(y)>8:
    continue
  else:
    y=int(y)
    break

while y in chosenpositions:
  print('that position has been taken! try again')
  try:
    y=int(input('Player 2 ('+str(player2)+') pick a number from 0-8'))
  except ValueError:
    print('That was not a number! Please try again')
    continue
  else:
    y=int(y)
    break

else:
  y=int(y)
  print('You chose, position '+str(y))
  board[y] = player2
  chosenpositions.insert(0,y)
  print_board()
  goes=goes+1
  winner_check()
  if winner==1:
    break
```

Full Python code with comments in python. [Part 1]



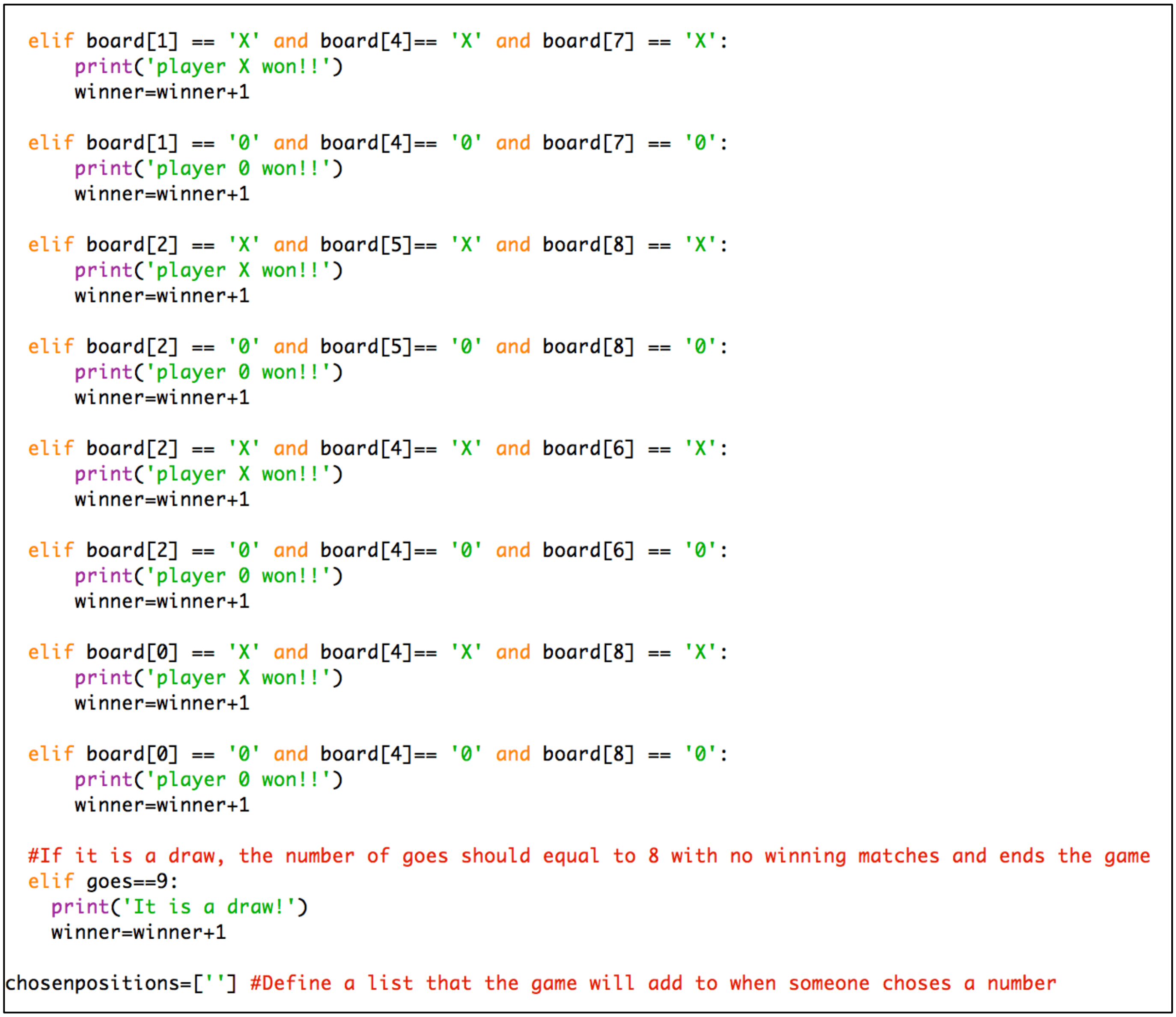
```
print('The grid layout')
print('0|1|2\n3|4|5\n6|7|8')
print(' ')

#defines a valid input list to be checked against when the user enters their choice
valid=['X','0']


#Creating a board from a list which allows a numbered grid to be made
board = ['_'] * 9
def print_board():
  print(board[0] + '|' + board[1] + '|' + board[2])
  print(board[3] + '|' + board[4] + '|' + board[5])
  print(board[6] + '|' + board[7] + '|' + board[8])

print_board() #This runs the routine to print the board

#asking for player 1 input

print('Welcome to tic tac toe!')


player1=input('player 1. Choose either X or 0')
player2=''

while player1!='X' or '0': #Keep asking for user input until a valid entry
  if player1=='0':
    player2='X'
    print('player 2 you are X')
    break

  if player1=='X':
    player2='0'
    print('player 2 you are 0')
    break

  else:
    print('That was not a valid input! Please try again')
    player1=input('player 1. Choose either X or 0')
```

Python Code with comments - Part 2



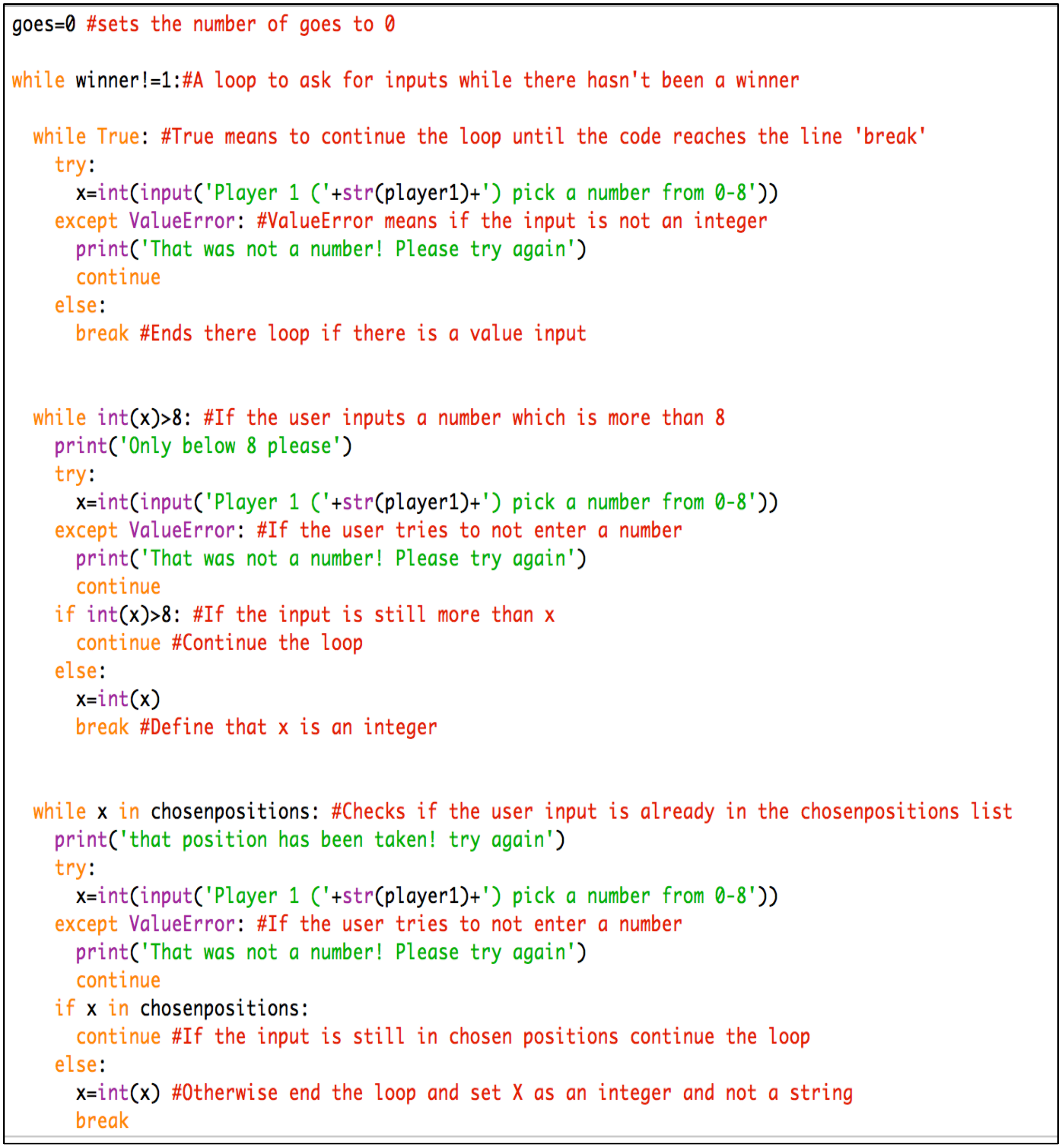
```
winner=0 #variable to see if there is a winner or not

def winner_check():

  global winner

  #These are all of the possible win combinaitons within the game

  if board[0] == 'X' and board[1]=='X' and board[2] == 'X':
     print('player X won!!')
     winner=winner+1

  elif board[0] == '0' and board[1]== '0' and board[2] == '0':
    print('player O won!!')
    winner=winner+1

  elif board[3] == 'X' and board[4]== 'X' and board[5] == 'X':
    print('player X won!!')
    winner=winner+1

  elif board[3] == '0' and board[4]== '0' and board[5] == '0':
    print('player 0 won!!')
    winner=winner+1

  elif board[6] == 'X' and board[7]== 'X' and board[8] == 'X':
    print('player X won!!')
    winner=winner+1

  elif board[6] == '0' and board[7]== '0' and board[8] == '0':
    print('player 0 won!!')
    winner=winner+1

  elif board[0] == 'X' and board[3]== 'X' and board[6] == 'X':
    print('player X won!!')
    winner=winner+1

  elif board[0] == '0' and board[3]== '0' and board[6] == '0':
    print('player 0 won!!')
    winner=winner+1
```

Python Code with comments - Part 3



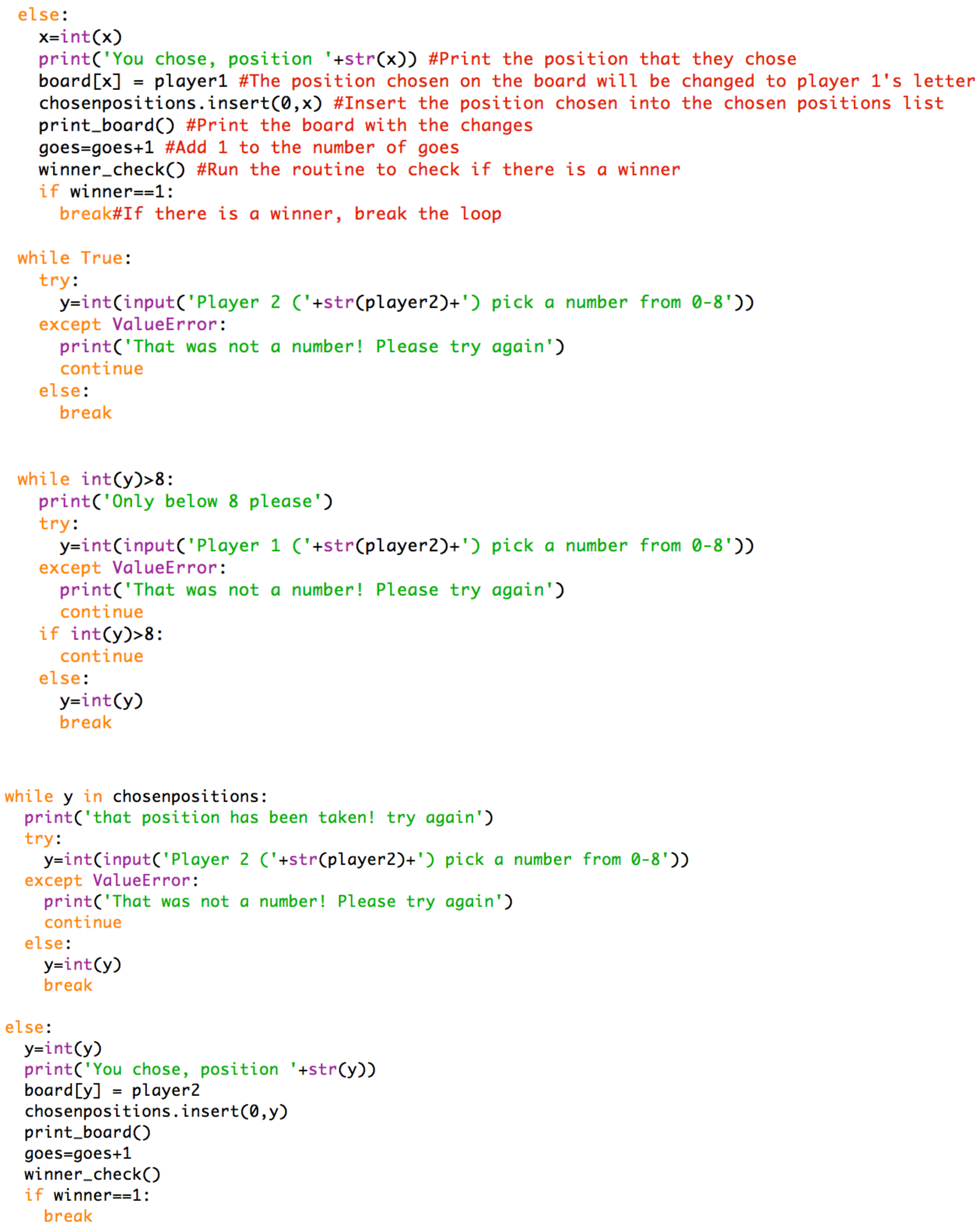
```
  elif board[1] == 'X' and board[4]== 'X' and board[7] == 'X':
      print('player X won!!')
      winner=winner+1

  elif board[1] == '0' and board[4]== '0' and board[7] == '0':
      print('player 0 won!!')
      winner=winner+1

  elif board[2] == 'X' and board[5]== 'X' and board[8] == 'X':
      print('player X won!!')
      winner=winner+1

  elif board[2] == '0' and board[5]== '0' and board[8] == '0':
      print('player 0 won!!')
      winner=winner+1

  elif board[2] == 'X' and board[4]== 'X' and board[6] == 'X':
      print('player X won!!')
      winner=winner+1

  elif board[2] == '0' and board[4]== '0' and board[6] == '0':
      print('player 0 won!!')
      winner=winner+1

  elif board[0] == 'X' and board[4]== 'X' and board[8] == 'X':
      print('player X won!!')
      winner=winner+1

  elif board[0] == '0' and board[4]== '0' and board[8] == '0':
      print('player 0 won!!')
      winner=winner+1

  #If it is a draw, the number of goes should equal to 8 with no winning matches and ends the game
  elif goes==9:
    print('It is a draw!')
    winner=winner+1

chosenpositions=[''] #Define a list that the game will add to when someone choses a number
```

Python Code with comments - Part 4



```
goes=0 #sets the number of goes to 0

while winner!=1:#A loop to ask for inputs while there hasn't been a winner

  while True: #True means to continue the loop until the code reaches the line 'break'
    try:
      x=int(input('Player 1 ('+str(player1)+') pick a number from 0-8'))
    except ValueError: #ValueError means if the input is not an integer
      print('That was not a number! Please try again')
      continue
    else:
      break #Ends there loop if there is a value input


  while int(x)>8: #If the user inputs a number which is more than 8
    print('Only below 8 please')
    try:
      x=int(input('Player 1 ('+str(player1)+') pick a number from 0-8'))
    except ValueError: #If the user tries to not enter a number
      print('That was not a number! Please try again')
      continue
    if int(x)>8: #If the input is still more than x
      continue #Continue the loop
    else:
      x=int(x)
      break #Define that x is an integer


  while x in chosenpositions: #Checks if the user input is already in the chosenpositions list
    print('that position has been taken! try again')
    try:
      x=int(input('Player 1 ('+str(player1)+') pick a number from 0-8'))
    except ValueError: #If the user tries to not enter a number
      print('That was not a number! Please try again')
      continue
    if x in chosenpositions:
      continue #If the input is still in chosen positions continue the loop
    else:
      x=int(x) #Otherwise end the loop and set X as an integer and not a string
      break
```

Python Code with comments - Part 5



```
else:
  x=int(x)
  print('You chose, position '+str(x)) #Print the position that they chose
  board[x] = player1 #The position chosen on the board will be changed to player 1's letter
  chosenpositions.insert(0,x) #Insert the position chosen into the chosen positions list
  print_board() #Print the board with the changes
  goes=goes+1 #Add 1 to the number of goes
  winner_check() #Run the routine to check if there is a winner
  if winner==1:
    break#If there is a winner, break the loop

while True:
  try:
    y=int(input('Player 2 ('+str(player2)+') pick a number from 0-8'))
  except ValueError:
    print('That was not a number! Please try again')
    continue
  else:
    break


while int(y)>8:
  print('Only below 8 please')
  try:
    y=int(input('Player 1 ('+str(player2)+') pick a number from 0-8'))
  except ValueError:
    print('That was not a number! Please try again')
    continue
  if int(y)>8:
    continue
  else:
    y=int(y)
    break


while y in chosenpositions:
  print('that position has been taken! try again')
  try:
    y=int(input('Player 2 ('+str(player2)+') pick a number from 0-8'))
  except ValueError:
    print('That was not a number! Please try again')
    continue
  else:
    y=int(y)
    break

else:
  y=int(y)
  print('You chose, position '+str(y))
  board[y] = player2
  chosenpositions.insert(0,y)
  print_board()
  goes=goes+1
  winner_check()
  if winner==1:
    break
```