西安电子科技大学

安全前沿讨论班（I） 课程实验报告

实验名称 实现 Tic_Tac_Toe 游戏

学　院	网络与信息安全学院	班 级	1618019
姓　名	曹寅峰	学 号	16200610025

实验日期 2019 年 5 月 7 日

指导教师评语：
指导教师：
______年____月____日

实验报告内容基本要求及参考格式

一、实验目的

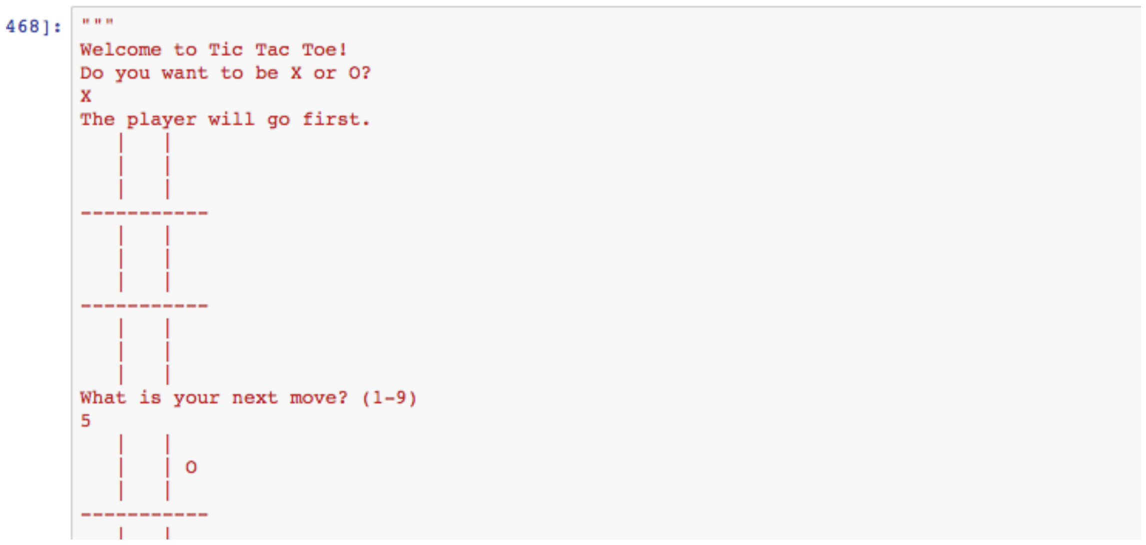
井字棋（Tic Tac Toe, or noughts and crosses）游戏，又称连三子棋，OX 棋，是一种供两人玩的纸笔游戏。两个玩家轮流在九个空格中画上代表自己的 0 或 X，谁先将自己的符号连成一线（横连、竖连、斜连皆可），即获得胜利。倘若在游戏过程中，在所有空格都填满的情况下，双方都没有取得胜利，那么游戏以平局告终。根据游戏规则，将游戏步骤分解，编程实现

二、实验环境

Python3.5　jupyter

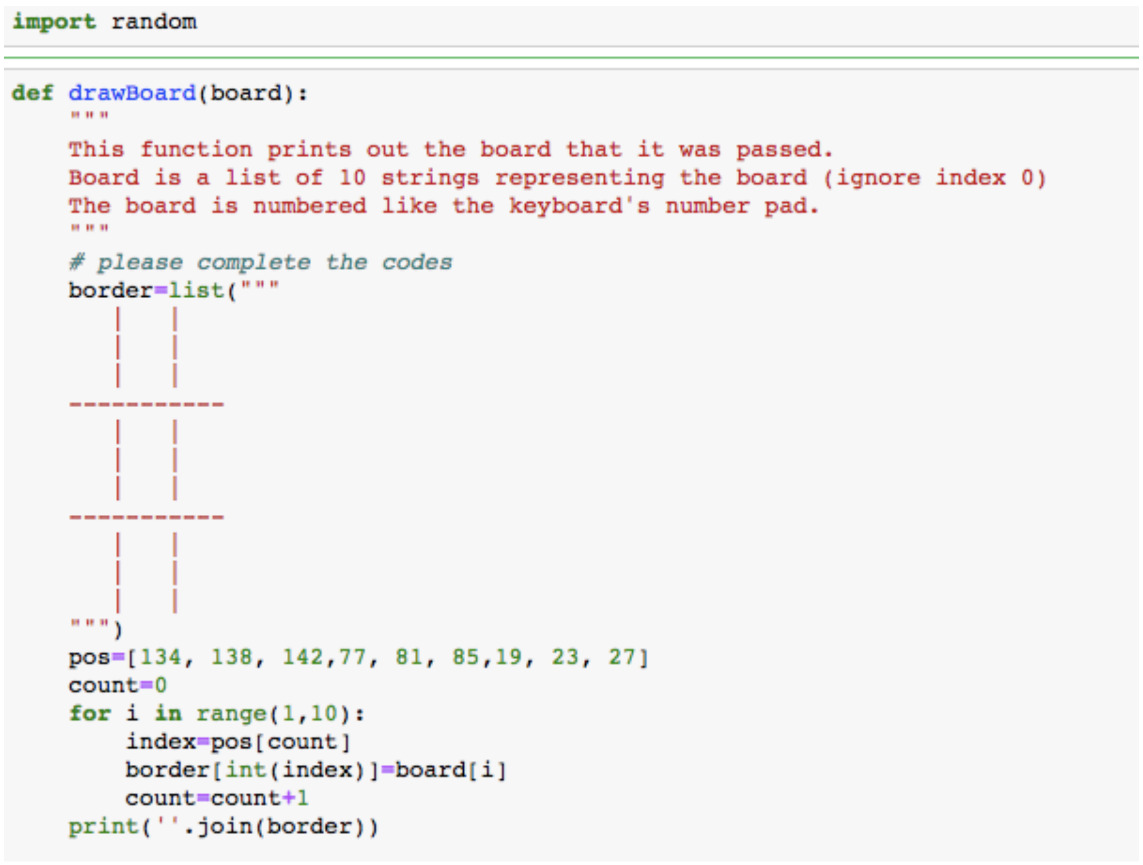
三、实验基本原理及步骤

首先根据说明理解游戏规则，玩家和电脑随机选择先手，各自下棋，要实现棋盘绘画，胜负判定，电脑对战逻辑等。



```
468]: """
      Welcome to Tic Tac Toe!
      Do you want to be X or O?
      X
      The player will go first.
         |   |
         |   |
         |   |
      -----------
         |   |
         |   |
         |   |
      -----------
         |   |
         |   |
         |   |
      What is your next move? (1-9)
      5
         |   |
         |   | O
         |   |
      -----------
         |   |
```

1. drawBoard 函数



```
import random
```

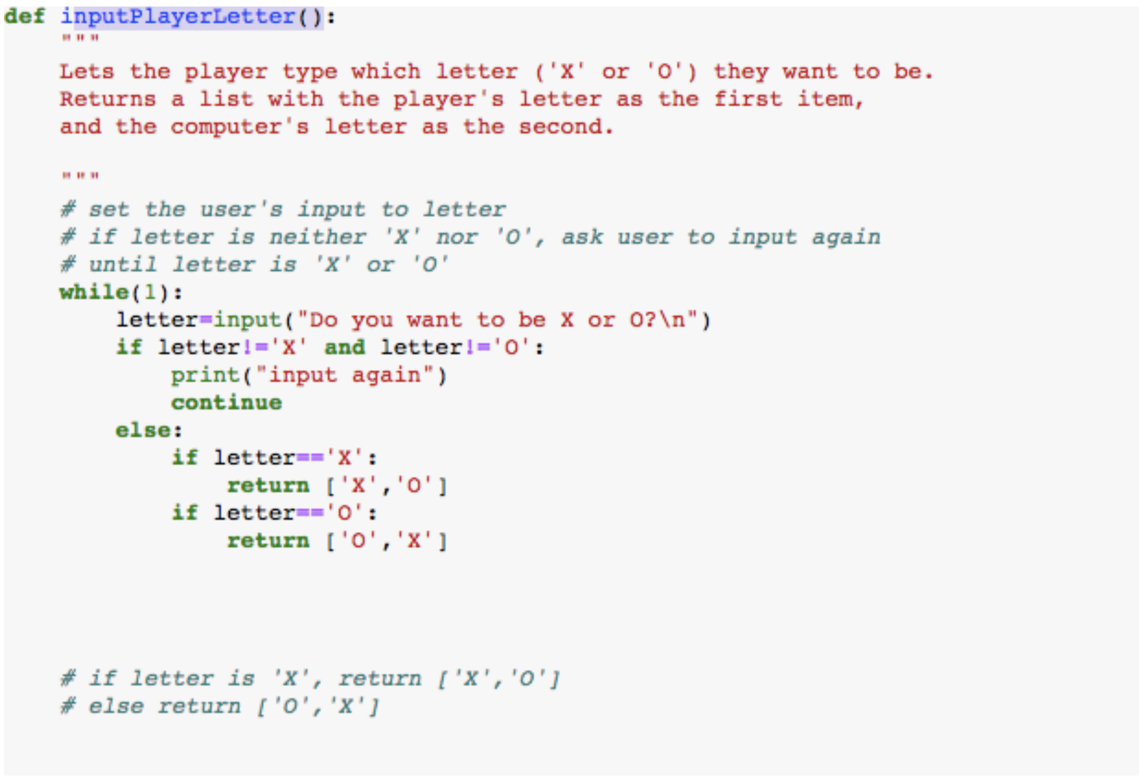
```
def drawBoard(board):
    """
    This function prints out the board that it was passed.
    Board is a list of 10 strings representing the board (ignore index 0)
    The board is numbered like the keyboard's number pad.
    """
    # please complete the codes
    border=list("""
       |   |
       |   |
       |   |
    -----------
       |   |
       |   |
       |   |
    -----------
       |   |
       |   |
       |   |
    """)
    pos=[134, 138, 142,77, 81, 85,19, 23, 27]
    count=0
    for i in range(1,10):
        index=pos[count]
        border[int(index)]=board[i]
        count=count+1
    print(''.join(border))
```

这个函数功能是根据输入的 board 列表绘制出图形化的棋盘。

主要思路是：

1. 找出棋盘中心的坐标（一维形式）
2. 替换对应坐标的字符串

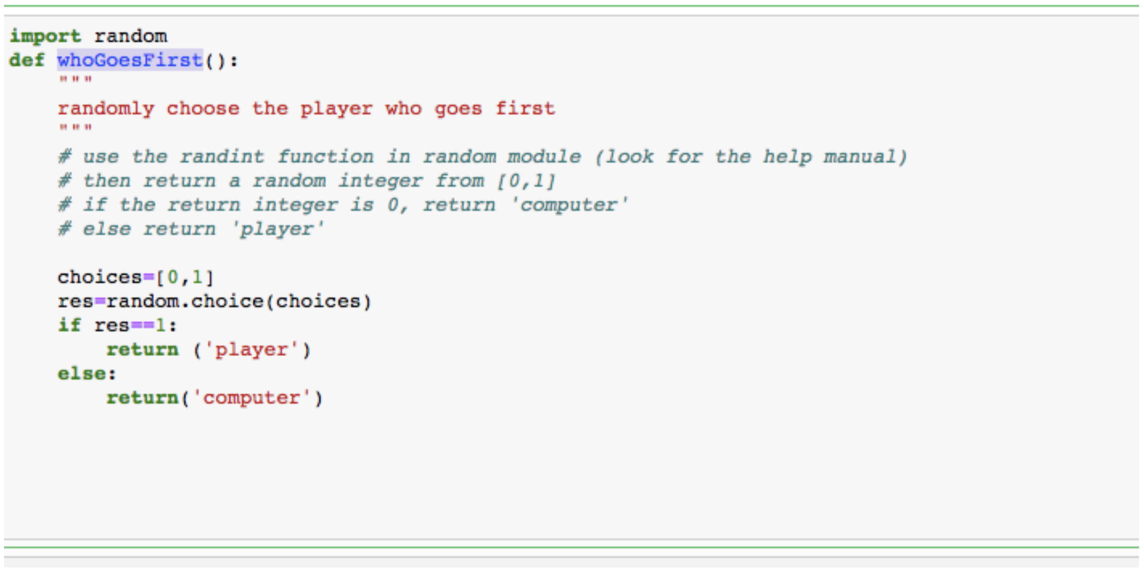
2. inputPlayerLetter 函数



```python
def inputPlayerLetter():
    """
    Lets the player type which letter ('X' or 'O') they want to be.
    Returns a list with the player's letter as the first item,
    and the computer's letter as the second.

    """
    # set the user's input to letter
    # if letter is neither 'X' nor 'O', ask user to input again
    # until letter is 'X' or 'O'
    while(1):
        letter=input("Do you want to be X or O?\n")
        if letter!='X' and letter!='O':
            print("input again")
            continue
        else:
            if letter=='X':
                return ['X','O']
            if letter=='O':
                return ['O','X']



    # if letter is 'X', return ['X','O']
    # else return ['O','X']
```

这个函数比较简单，用于分配玩家和电脑的棋子，简单判断直接返回即可

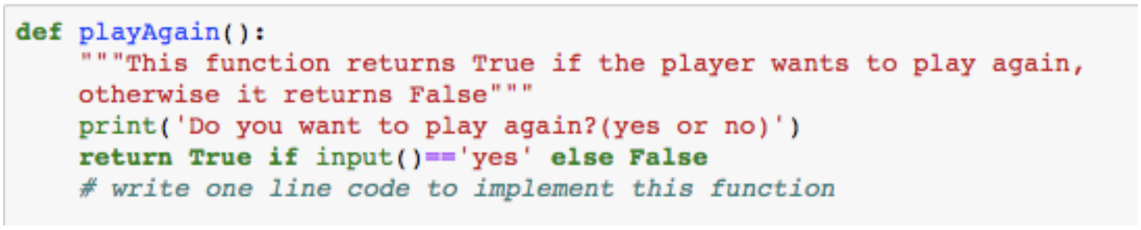
3. whoGoesFirst 函数



```python
import random
def whoGoesFirst():
    """
    randomly choose the player who goes first
    """
    # use the randint function in random module (look for the help manual)
    # then return a random integer from [0,1]
    # if the return integer is 0, return 'computer'
    # else return 'player'

    choices=[0,1]
    res=random.choice(choices)
    if res==1:
        return ('player')
    else:
        return('computer')
```

这个函数的功能是随机分配先手权，利用到了 random 库的 choice 函数，功能是随机选择列表中的一个元素并返回。

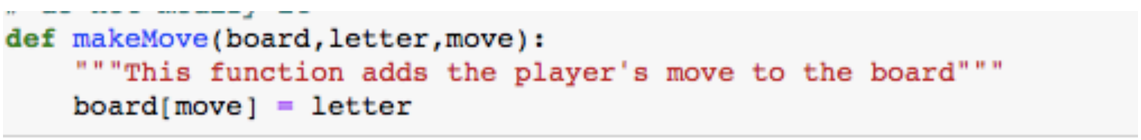
4. playAgain 函数



```python
def playAgain():
    """This function returns True if the player wants to play again,
    otherwise it returns False"""
    print('Do you want to play again?(yes or no)')
    return True if input()=='yes' else False
    # write one line code to implement this function
```

这个函数判断是否需要再次玩游戏，特点是要求使用一句话判断，这里我们使用 pythpn 的三目运算（if else）达成一句话完成的效果

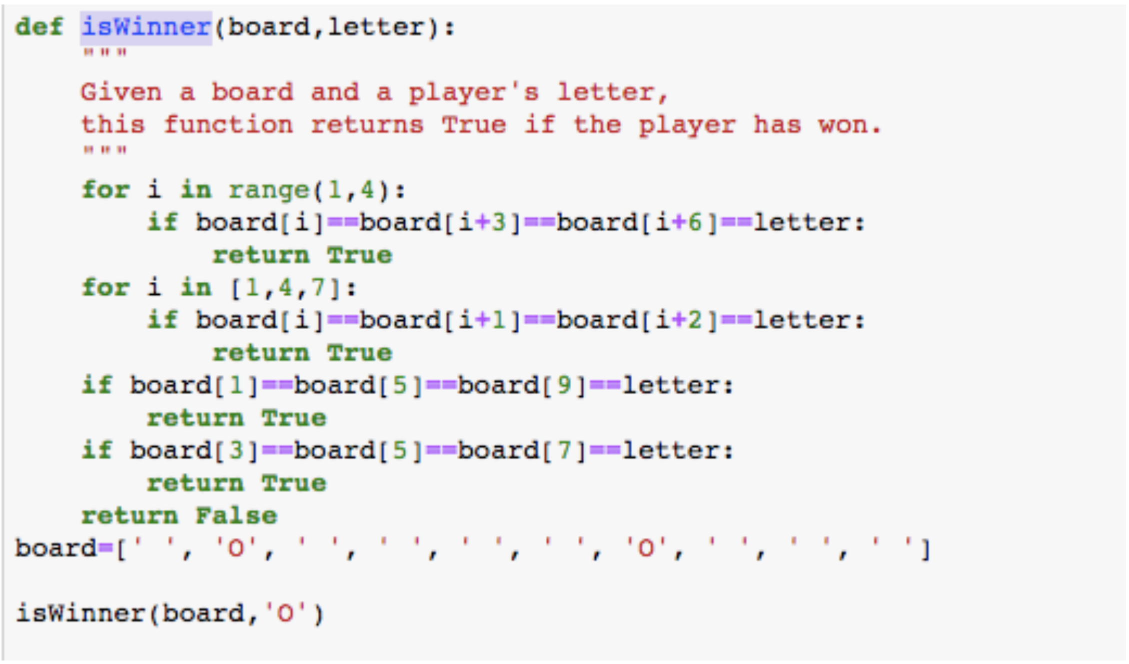
5. Makemove 函数



```python
def makeMove(board,letter,move):
    """This function adds the player's move to the board"""
    board[move] = letter
```

通过位置改变棋盘格局，直接替换即可

6. isWinner 函数

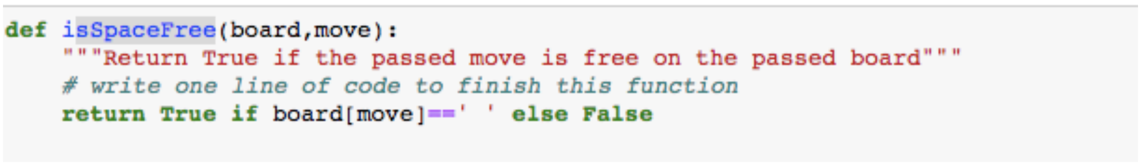


```python
def isWinner(board,letter):
    """
    Given a board and a player's letter,
    this function returns True if the player has won.
    """
    for i in range(1,4):
        if board[i]==board[i+3]==board[i+6]==letter:
            return True
    for i in [1,4,7]:
        if board[i]==board[i+1]==board[i+2]==letter:
            return True
    if board[1]==board[5]==board[9]==letter:
        return True
    if board[3]==board[5]==board[7]==letter:
        return True
    return False
board=[' ', 'O', ' ', ' ', ' ', ' ', 'O', ' ', ' ', ' ']

isWinner(board,'O')
```

这个函数是判断胜利条件，我们知道，井字棋有八种胜利格局（横三种，竖三种，交叉两种）

依次判断即可

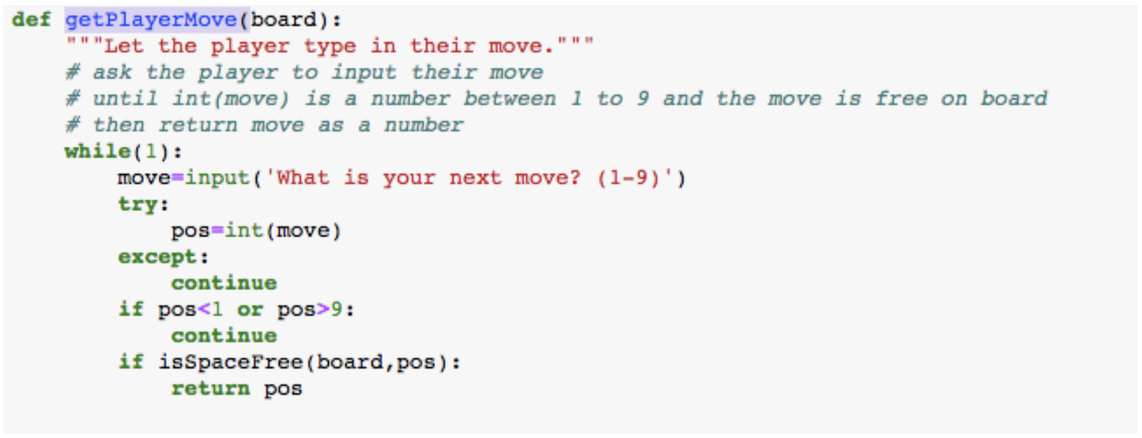
7. isSpaceFree 函数



```python
def isSpaceFree(board,move):
    """Return True if the passed move is free on the passed board"""
    # write one line of code to finish this function
    return True if board[move]==' ' else False
```

判断对应位置是否为空，同样利用三目运算符达成一句话的效果

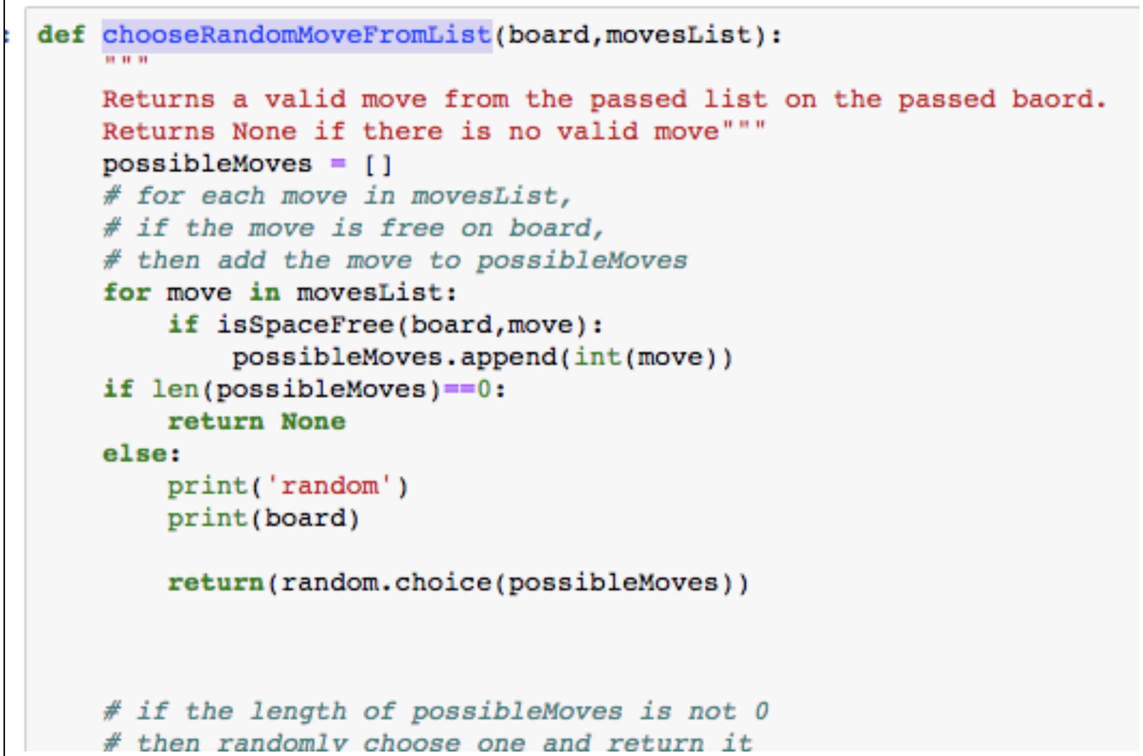
8. getPlayerMove 函数



```python
def getPlayerMove(board):
    """Let the player type in their move."""
    # ask the player to input their move
    # until int(move) is a number between 1 to 9 and the move is free on board
    # then return move as a number
    while(1):
        move=input('What is your next move? (1-9)')
        try:
            pos=int(move)
        except:
            continue
        if pos<1 or pos>9:
            continue
        if isSpaceFree(board,pos):
            return pos
```

通过函数接收用户的期望下棋位置，要注意判断输入类型，这里检测了是否为数字以及数字是否为整数

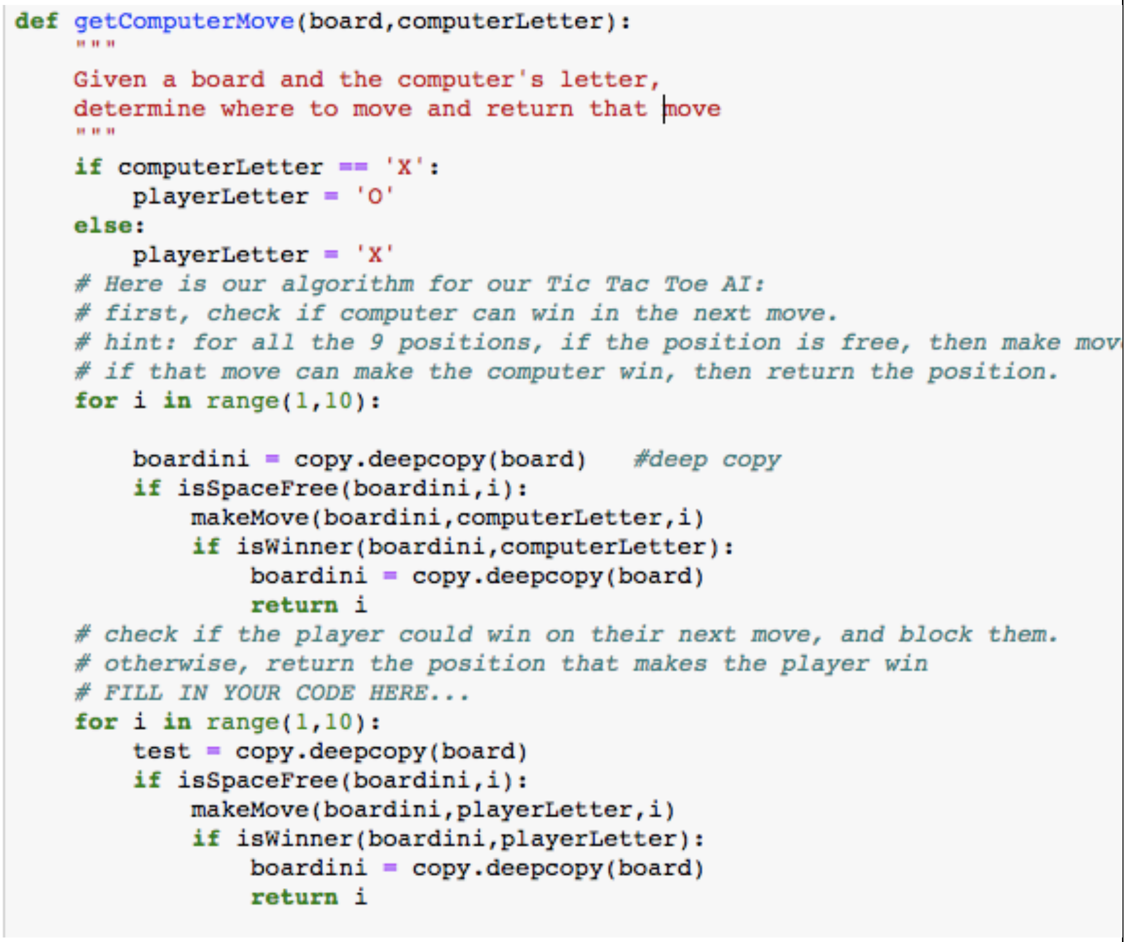
9. chooseRandomMoveFromList 函数



```
def chooseRandomMoveFromList(board,movesList):
    """
    Returns a valid move from the passed list on the passed baord.
    Returns None if there is no valid move"""
    possibleMoves = []
    # for each move in movesList,
    # if the move is free on board,
    # then add the move to possibleMoves
    for move in movesList:
        if isSpaceFree(board,move):
            possibleMoves.append(int(move))
    if len(possibleMoves)==0:
        return None
    else:
        print('random')
        print(board)

        return(random.choice(possibleMoves))



    # if the length of possibleMoves is not 0
    # then randomly choose one and return it
```

如同名字一样，从列表中随机选择一个位置返回

10.getComputerMove 函数



```python
def getComputerMove(board,computerLetter):
    """
    Given a board and the computer's letter,
    determine where to move and return that move
    """
    if computerLetter == 'X':
        playerLetter = 'O'
    else:
        playerLetter = 'X'
    # Here is our algorithm for our Tic Tac Toe AI:
    # first, check if computer can win in the next move.
    # hint: for all the 9 positions, if the position is free, then make mov
    # if that move can make the computer win, then return the position.
    for i in range(1,10):

        boardini = copy.deepcopy(board)    #deep copy
        if isSpaceFree(boardini,i):
            makeMove(boardini,computerLetter,i)
            if isWinner(boardini,computerLetter):
                boardini = copy.deepcopy(board)
                return i
    # check if the player could win on their next move, and block them.
    # otherwise, return the position that makes the player win
    # FILL IN YOUR CODE HERE...
    for i in range(1,10):
        test = copy.deepcopy(board)
        if isSpaceFree(boardini,i):
            makeMove(boardini,playerLetter,i)
            if isWinner(boardini,playerLetter):
                boardini = copy.deepcopy(board)
                return i
```

电脑的“AI”判断程序。这个程序是我调试最久的程序，电脑的下棋逻辑如下：

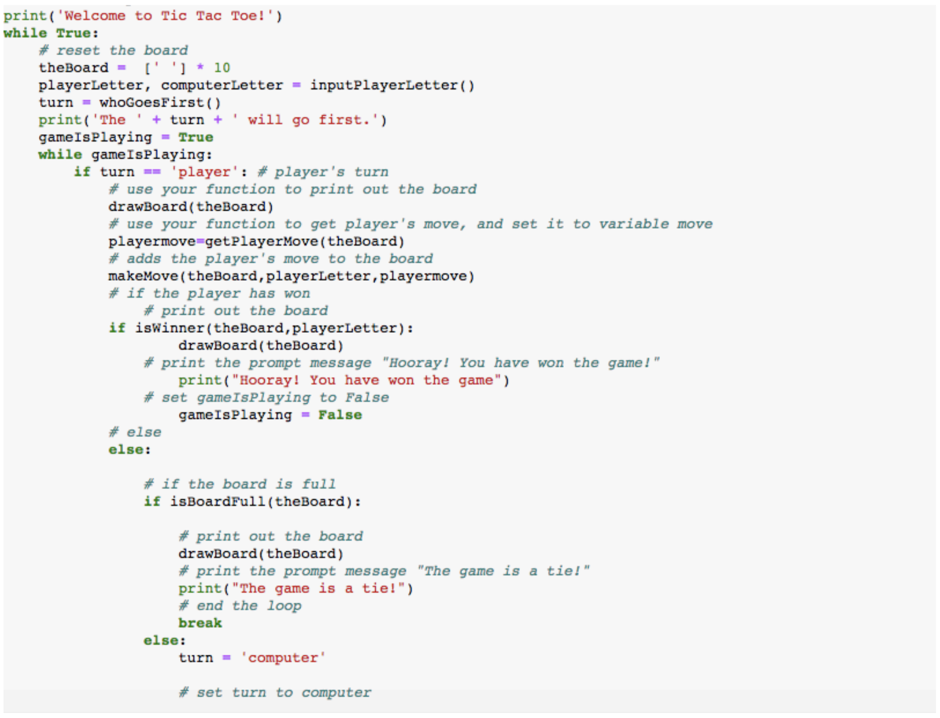
1.如果某一位置能赢，下在该位置

2.如果某一位置阻止玩家获得升级，下在该位置

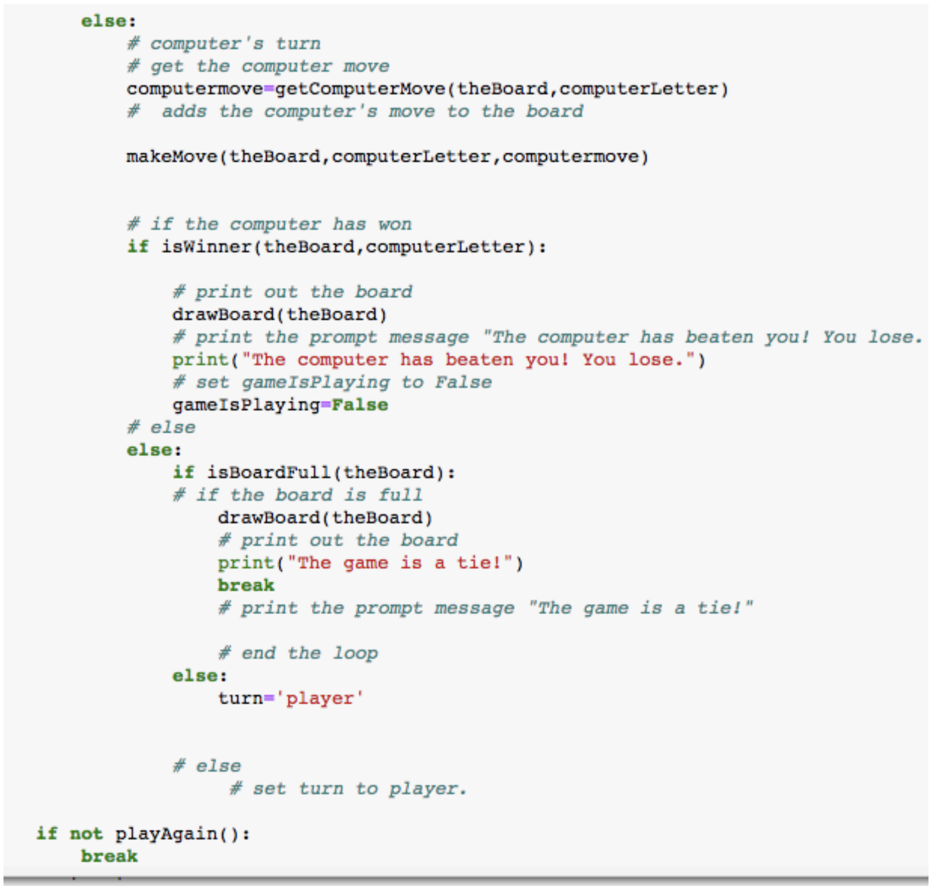
3.如果都没有，按照角落，中央，其他优先级随机选择一个位置

这里注意初始棋盘会被改变，所以利用了 copy 库复制出一个棋盘，同时在返回之前**恢复到初始状态**

11.main 函数



```
print('Welcome to Tic Tac Toe!')
while True:
    # reset the board
    theBoard = [' '] * 10
    playerLetter, computerLetter = inputPlayerLetter()
    turn = whoGoesFirst()
    print('The ' + turn + ' will go first.')
    gameIsPlaying = True
    while gameIsPlaying:
        if turn == 'player': # player's turn
            # use your function to print out the board
            drawBoard(theBoard)
            # use your function to get player's move, and set it to variable move
            playermove=getPlayerMove(theBoard)
            # adds the player's move to the board
            makeMove(theBoard,playerLetter,playermove)
            # if the player has won
                # print out the board
            if isWinner(theBoard,playerLetter):
                    drawBoard(theBoard)
                # print the prompt message "Hooray! You have won the game!"
                    print("Hooray! You have won the game")
                # set gameIsPlaying to False
                    gameIsPlaying = False
            # else
            else:

                # if the board is full
                if isBoardFull(theBoard):

                    # print out the board
                    drawBoard(theBoard)
                    # print the prompt message "The game is a tie!"
                    print("The game is a tie!")
                    # end the loop
                    break
                else:
                    turn = 'computer'

                    # set turn to computer
```



```
        else:
            # computer's turn
            # get the computer move
            computermove=getComputerMove(theBoard,computerLetter)
            #  adds the computer's move to the board

            makeMove(theBoard,computerLetter,computermove)


            # if the computer has won
            if isWinner(theBoard,computerLetter):

                # print out the board
                drawBoard(theBoard)
                # print the prompt message "The computer has beaten you! You lose.
                print("The computer has beaten you! You lose.")
                # set gameIsPlaying to False
                gameIsPlaying=False
            # else
            else:
                if isBoardFull(theBoard):
                # if the board is full
                    drawBoard(theBoard)
                    # print out the board
                    print("The game is a tie!")
                    break
                    # print the prompt message "The game is a tie!"

                    # end the loop
                else:
                    turn='player'


                # else
                     # set turn to player.

    if not playAgain():
        break
```

最终将以上的函数们组合起来，形成总函数。

这里老师非常贴心（点个赞），把主要步骤都详细注释了，填空即可，别写错变量名称即可

四、实验结果分析

以下为三种不同结局

```
Welcome to Tic Tac Toe!
Do you want to be X or O?

input again
Do you want to be X or O?

input again
Do you want to be X or O?

input again
Do you want to be X or O?

input again
Do you want to be X or O?
X
The player will go first.

      |   |
      |   |
      |   |
   -----------
      |   |
      |   |
      |   |
   -----------
      |   |
      |   |
      |   |

What is your next move? (1-9)1

      |   |
      |   |
      |   |
   -----------
```

```
       |   |
       |   |
       |   |
   -----------
       |   |
     X |   | O
       |   |

What is your next move? (1-9)2

       |   |
       |   |
       |   |
   -----------
       |   |
       |   | O
       |   |
   -----------
       |   |
     X | X | O
       |   |

What is your next move? (1-9)9

       |   |
       |   | X
       |   |
   -----------
       |   |
       | O | O
       |   |
   -----------
       |   |
     X | X | O
       |   |

What is your next move? (1-9)4

       |   |
     O |   | X
       |   |
   -----------
       |   |
     X | O | O
```

```
      |   |
   -----------
      |   |
    X | X | O
      |   |

The computer has beaten you! You lose.
Do you want to play again?(yes or no)
yes
Do you want to be X or O?
X
The player will go first.

      |   |
      |   |
      |   |
   -----------
      |   |
      |   |
      |   |
   -----------
      |   |
      |   |
      |   |

What is your next move? (1-9)1

      |   |
      |   |
      |   |
   -----------
      |   |
      |   |
      |   |
   -----------
      |   |
    X |   | O
      |   |

What is your next move? (1-9)7

      |   |
    X |   |
      |   |
   -----------
      |   |
    O |   |
      |   |
   -----------
      |   |
    X |   | O
      |   |

What is your next move? (1-9)5

      |   |
    X | O |
      |   |
   -----------
      |   |
    O | X |
      |   |
   -----------
      |   |
    X |   | O
      |   |

What is your next move? (1-9)9

      |   |
    X | O | X
      |   |
   -----------
      |   |
    O | X |
      |   |
   -----------
      |   |
    X |   | O
      |   |

Hooray! You have won the game
Do you want to play again?(yes or no)
yes
Do you want to be X or O?
O
The computer will go first.
```

```
      |   |
      |   |
      |   |
   -----------
      |   |
      |   |
      |   |
   -----------
      |   |
      |   | X
      |   |

What is your next move? (1-9)7

      |   |
    O |   |
      |   |
   -----------
      |   |
    X |   |
      |   |
   -----------
      |   |
      |   | X
      |   |

What is your next move? (1-9)1

      |   |
    O | X |
      |   |
   -----------
      |   |
    X |   |
      |   |
   -----------
      |   |
    O |   | X
      |   |

What is your next move? (1-9)9

      |   |
    O | X | O
```

```
       |   |
    -----------
       |   |
     X | X |
       |   |
    -----------
       |   |
     O |   | X
       |   |

What is your next move? (1-9)6


       |   |
     O | X | O
       |   |
    -----------
       |   |
     X | X | O
       |   |
    -----------
       |   |
     O |   | X
       |   |

The game is a tie!
Do you want to play again?(yes or no)
```