```
# File: Project2.py
# Student: Ramsay Ward
# UT EID: rjw2777
# Course Name: CS303E
#
# Date: Nov 11th, 2022
# Description of Program: This program will have the user play a smiple game of
Tic-Tac-Toe against an AI.


import random



HUMAN   = 0
MACHINE = 1
WELCOME = "Welcome to our Tic-Tac-Toe game!\nPlease begin playing.\n"
YOU_WON  = "Congratulations! You won!\n"
YOU_TIED = "\nLooks like a tie. Better luck next time!"
YOU_LOST = "\nSorry! You lost!"




def initialBoard():
    return  [ [" ", " ", " "], \
              [" ", " ", " "], \
              [" ", " ", " "] ]


class TicTacToe:

    def __init__(self):
        self.__getPlayer = HUMAN
        self.__board = initialBoard()


    def __str__(self):
        return self.__board[0][0] + "|" + self.__board[0][1] + "|" +
self.__board[0][2] + " "  +"\n" + "-----" +"\n" + self.__board[1][0] + "|" +
self.__board[1][1] + "|" + self.__board[1][2] + " " + "\n" + "-----" +"\n" +
self.__board[2][0] + "|" + self.__board[2][1] + "|" + self.__board[2][2]+ " "


    def getPlayer( self ):
        return self.__getPlayer


    def isWin( self ):
        isWin = False


        #colums
        if (self.__board[0][0] + self.__board[1][0] + self.__board[2][0]) == "XXX":
            isWin = True

        if (self.__board[0][1] + self.__board[1][1] + self.__board[2][1]) == "XXX":
            isWin = True
        if (self.__board[0][2] + self.__board[1][2] + self.__board[2][2]) == "XXX":
            isWin = True


        #rows
        if (self.__board[0][0] + self.__board[0][1] + self.__board[0][2]) == "XXX":
            isWin = True

        if (self.__board[1][0] + self.__board[1][1] + self.__board[1][2]) == "XXX":
            isWin = True

        if (self.__board[2][0] + self.__board[2][1] + self.__board[2][2]) == "XXX":
            isWin = True

        #Diagonal
        if (self.__board[0][0] + self.__board[1][1] + self.__board[2][2]) == "XXX":
            isWin = True
        if (self.__board[0][2] + self.__board[1][1] + self.__board[2][0]) == "XXX":
            isWin = True




        #colums
        if (self.__board[0][0] + self.__board[1][0] + self.__board[2][0]) == "OOO":
            isWin = True
        if (self.__board[0][1] + self.__board[1][1] + self.__board[2][1]) == "OOO":
            isWin = True
        if (self.__board[0][2] + self.__board[1][2] + self.__board[2][2]) == "OOO":
            isWin = True

        #rows
        if (self.__board[0][0] + self.__board[0][1] + self.__board[0][2]) == "OOO":
            isWin = True
        if (self.__board[1][0] + self.__board[1][1] + self.__board[1][2]) == "OOO":
            isWin = True
        if (self.__board[2][0] + self.__board[2][1] + self.__board[2][2]) == "OOO":
            isWin = True
        #Diagonal
        if (self.__board[0][0] + self.__board[1][1] + self.__board[2][2]) == "OOO":
            isWin = True
        if (self.__board[0][2] + self.__board[1][1] + self.__board[2][0]) == "OOO":
            isWin = True

        return (isWin)



    def swapPlayers( self ):

        if self.__getPlayer == HUMAN:

            self.__getPlayer = MACHINE
        else:
            self.__getPlayer = HUMAN

    def humanMove( self ):
        # Ask the HUMAN to specify a move.  Check that it's
        # valid (syntactically, in range, and the space is
        # not occupied).  Update the board appropriately.
        print()
        print("Your turn:")




        while True:

            userInput = input("  Specify a move r, c: ")
            print()
            location = userInput.split(",")
            if len(location) != 2:
                print("Response should be r, c. Try again!")
            else:
                r = int (location[0])
                c = int (location[1])




                if int(location[0]) > 2 or int(location[0]) < 0:
                    print("Illegal move specified. Try again!")


                elif int(location[1]) > 2 or int(location[1]) < 0:
                    print("Illegal move specified. Try again!")

                elif (self.__board[r][c] != " "):
                    print("Illegal move specified. Try again!")


                else:
                    break



        self.__board[r][c] = "X"

        return





    def machineMove( self ):
        # This is the MACHINE's move.  It picks squares randomly
        # until it finds an empty one. Update the board appropriately.
        # Note: this is a really dumb way to play tic-tac-toe.
        print()
        print("Machine's turn:")
        while True:
            r = random.randint(0, 2)
            c = random.randint(0, 2)
            if self.__board[r][c] == " ":
                print("  Machine chooses: ", r, ", ", c, sep="")
                print()
                self.__board[r][c] = "O"
                return





def driver( ):
    """ This plays tic-tac-toe in a pretty simple-minded
    fashion.  The human player goes first with token "X" and
    alternates with the machine using token "O".  We print
    the board before the first move and after each move. """

    # Print the welcome message
    print( WELCOME )

    # Initialize the board and current player
    ttt = TicTacToe()
    print( ttt )

    # There are up to 9 moves in tic-tac-toe.
    for move in range(9):
        # The current player may be HUMAN
        # or MACHINE
        if ttt.getPlayer() == HUMAN:
            # If HUMAN, take a move, print the board,
            # and see if it's a win.
            ttt.humanMove()
            print( ttt )
            if ttt.isWin():
                print()
                print( YOU_WON )
                return

        else:
            # Else MACHINE takes a move.  Print the
            # board and see if the machine won.
            ttt.machineMove()
            print( ttt )
            if ttt.isWin():
                print( YOU_LOST )
                return

        # Swap players.
        ttt.swapPlayers()

    # After nine moves with no winner, it's a tie.
    print( YOU_TIED )

driver()
```