Now that we have defined our players and also defined the game, the next thing is to define a function play that brings it all together. Our function would take in the game, as well as an x player and then a print\_game parameter. The idea is that you can decide if you want to see the game run or not for instance maybe if you are playing computer vs computer.

def play(game, x\_player, o\_player, print\_game=True):

    #returns the winner of the game or None for a tie

    if print\_game:

        game.print\_board\_nums()

If <code>print\_game</code> is true then the <code>print\_board\_nums</code> game method gets called which would print out the game board with numbers. To begin the game, we assign letter to “x”. This is our starting letter then we check while the game doesn’t have empty squares, we play.

def play(game, x\_player, o\_player, print\_game=True):

    #returns the winner of the game or None for a tie

    if print\_game:

        game.print\_board\_nums()

    letter = "X" #starting letter

    #iterate while the game still has empty squares

    #(we dont have to worry about the winner because we'll just return that

    #which breaks the loop)

    while game.empty\_squares():

        if letter == "O":

            square = o\_player.get\_move(game)

        else:

            square = x\_player.get\_move(game)

While the game has empty squares, the appropriate plater gets their move. The next step is to make the move. In the games <code>make\_move()</code> method, it only makes the move if the square is valid and then returns true otherwise it returns false. So if the move gotten by the player is ok, it makes the move (in our case, the move would always be valid because in <code>player.get\_move(game)</code> we have ensured that the move is an available move so our <code>make\_move()</code> is always going to be true).

def play(game, x\_player, o\_player, print\_game=True):

    #returns the winner of the game or None for a tie

    if print\_game:

        game.print\_board\_nums()

    letter = "X" #starting letter

    #iterate while the game still has empty squares

    #(we dont have to worry about the winner because we'll just return that

    #which breaks the loop)

    while game.empty\_squares():

        if letter == "O":

            square = o\_player.get\_move(game)

        else:

            square = x\_player.get\_move(game)

        #define a function to make a move

        if game.make\_move(square, letter):

            if print\_game:

                print(letter + f" makes a move to square {square}")

                game.print\_board()

                print(" ") #just empty line

            if game.current\_winner:

                if print\_game:

                    print(letter + " wins!")

                return letter

            #after making a move, we need to alternate letters

            letter = "O" if letter == "X" else "X"

            # same code below

            # if letter == "X":

            #     letter = "O"

            # else:

            #     letter = "X"

        #tiny pause when player switches

        time.sleep(0.8)

After the player makes the move, we print the game board so the players can see and then if there is a winner, we print out who won. After making the move, we need to switch the letter to the other player. This entire loop is happening while there are empty spaces so if the empty spaces finish and there is no winner, we need to tell the players that it is a tie.

def play(game, x\_player, o\_player, print\_game=True):

    #returns the winner of the game or None for a tie

    if print\_game:

        game.print\_board\_nums()

    letter = "X" #starting letter

    #iterate while the game still has empty squares

    #(we dont have to worry about the winner because we'll just return that

    #which breaks the loop)

    while game.empty\_squares():

        if letter == "O":

            square = o\_player.get\_move(game)

        else:

            square = x\_player.get\_move(game)

        #define a function to make a move

        if game.make\_move(square, letter):

            if print\_game:

                print(letter + f" makes a move to square {square}")

                game.print\_board()

                print(" ") #just empty line

            if game.current\_winner:

                if print\_game:

                    print(letter + " wins!")

                return letter

            #after making a move, we need to alternate letters

            letter = "O" if letter == "X" else "X"

            # same code below

            # if letter == "X":

            #     letter = "O"

            # else:

            #     letter = "X"

        #tiny pause when player switches

        time.sleep(0.8)

    if print\_game:

        print("It's a tie!")

<code>time.sleep(0.8)</code> adds 800ms pause to the game so we can see what’s going on a bit cleare. To run the game, we just do the following at the bottom of the document. We assign the x and o players and also the game and then we call the play function passing in the required parameters.

if \_\_name\_\_ =="\_\_main\_\_":

    x\_player = HumanPlayer("X")

    o\_player = RandomComputerPlayer("O")

    t = TicTacToe()

    play(t, x\_player, o\_player, print\_game=True)

Of course at the beginning of the file, we would have to import the players into the game file.

import time

from player import HumanPlayer, RandomComputerPlayer

Below is an example of how it looks when the script runs.

<img src=”/images/t1c-tac-toe.JPG” /> <br />

<img src=”/images/t2c-tac-toe.JPG” />”><br />

<br />

Cheers 🥂