**Tic-Tac-Toe Game**

**INT 404 ARTIFICIAL INTELLIGENCE**

**Assignment 2019-2020**

**Lovely Professional University**

Group no. 1

Name : Dipanshu Raj Name : Shubham Tiwari Name : Yashika Sen

Reg no. : 11801965 Reg no. :11802108 Reg no. : 11802580

Roll no. 01 Roll no. : 02 Roll no. : 03

**Description:**

Tic Tac Toe is one of the most played games and is the best time killer game that you can play anywhere with just a pen and paper. The game is played by two individuals. First, we draw a board with a 3×3 square grid. The first player chooses ‘X’ and draws it on any of the square grid, then it’s the chance of the second player to draw ‘O’ on the available spaces. Like this, the players draw ‘X’ and ‘O’ alternatively on the empty spaces until a player succeeds in drawing 3 consecutive marks either in the horizontal, vertical or diagonal way. Then the player wins the game otherwise the game draws when all spots are filled.

To develop this game, we need to install pygame. Pygame is a great library that will allow us to create the window and draw images and shapes on the window. This way we will capture mouse coordinates and identify the block where we need to mark ‘X’ or ‘O’. Then we will check if the user wins the game or not.

**Game Components :**

Pygame library : It contains the modules needed for computer graphics and sound libraries.

Time library : because we will use the time.sleep() method to pause game at certain positions.

TTT : the main 3×3 Tic Tac Toe board and at first, it will have 9 None values. The height and width of the canvas where we will play the game is 400×400.

Pg.init()method : we initialize the pygame with pg.init() method and the window display is set with a width of 400 and a height of 500. We have reserved 100-pixel space for displaying the status of the game.

Pg.display.set\_mode() : initializes the display and we reference it with the screen variable. This screen variable will be used whenever we want to draw something on the display.

This project uses many images like the opening image that will display when the game starts or resets. The X and O images that we will draw when the user clicks on the grid. We load all the images and resize them so that they will fit easily in our window.

FUNCTIONS :

**blit()** function is used on the surface to draw an image on top of another image.

The **draw\_status()** function draws a black rectangle where we update the status of the game showing which player’s turn is it and whether the game ends or draws.

The **check\_win()** function checks the Tic Tac Toe board to see all the marks of ‘X’ and ‘O’. It calculates whether a player has won the game or not. They can either win when the player has marked 3 consecutive marks in a row, column or diagonally. This function is called every time when we draw a mark ‘X’ or ‘O’ on the board.

The **drawXO(row, col)** function takes the row and column where the mouse is clicked and then it draws the ‘X’ or ‘O’ mark. We calculate the x and y coordinates of the starting point from where we’ll draw the png image of the mark.

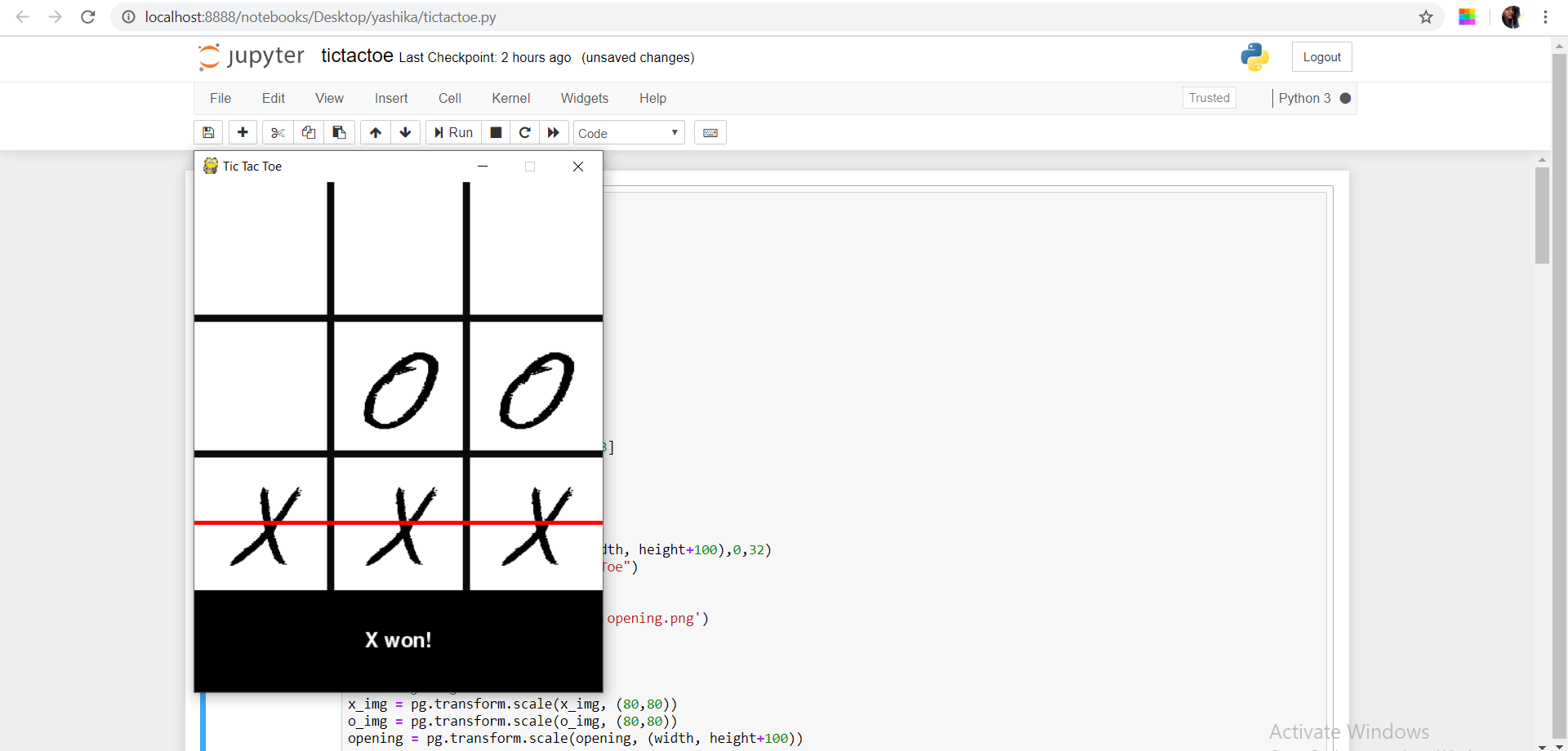
The **userClick()** function is triggered every time the user presses the mouse button.When the user clicks the mouse, we first take the x and y coordinates of where the mouse is clicked on the display window and then if that place is not occupied we draw the ‘XO’ on the canvas. We also check if the player wins or not after drawing ‘XO’ on the board.

The last function is the **reset\_game().** This will restart the game and we also reset all the variables to the beginning of the game.

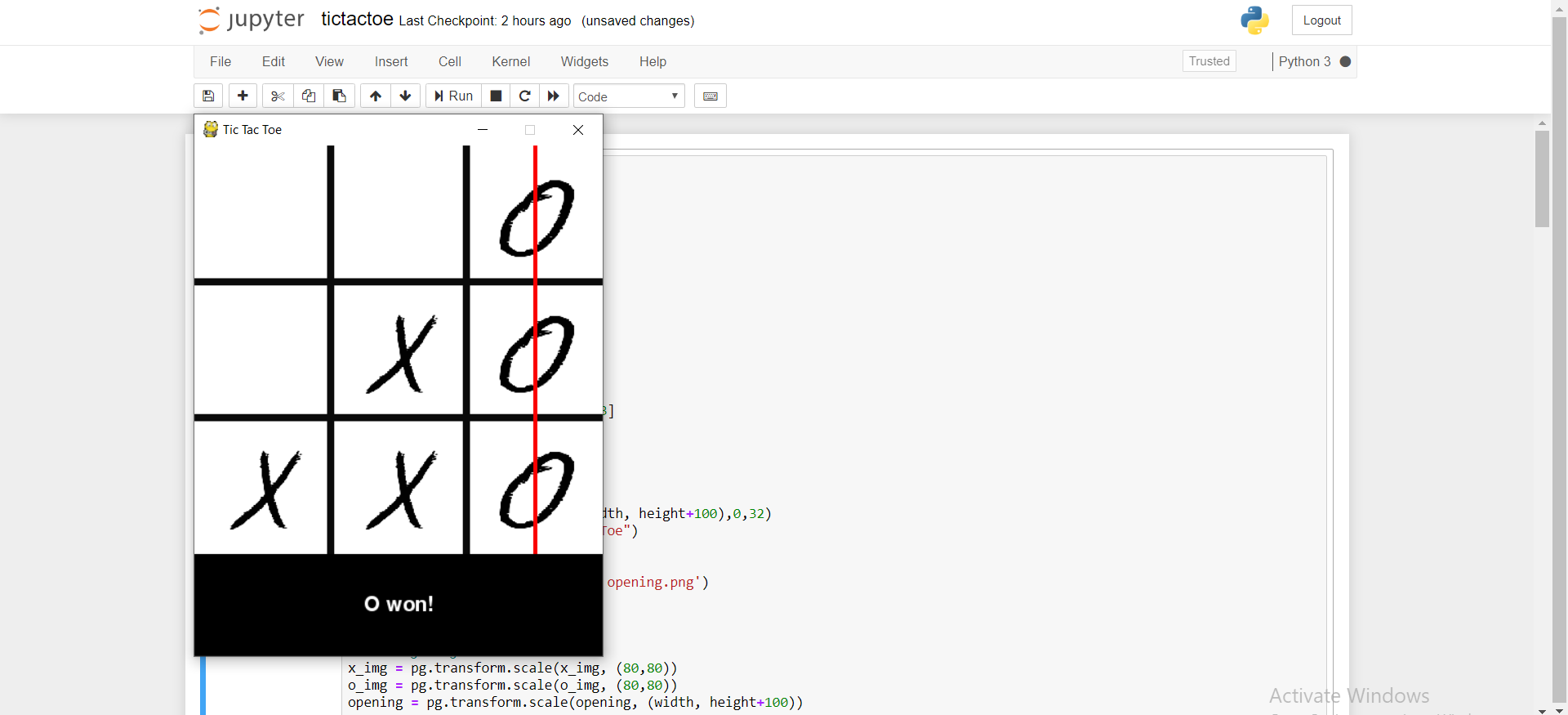
To start the game, we will call the **game\_opening()** function. Then, we run an infinite loop and continuously check for any event made by the user. If the user presses mouse button, the **MOUSEBUTTONDOWN** event will be captured and then we will trigger the **userClick()** function. Then if the user wins or the game draws, we reset the game by calling **reset\_game()** function. We update the display in each iteration and we have set the frames per second to 30.

TEST CASES :

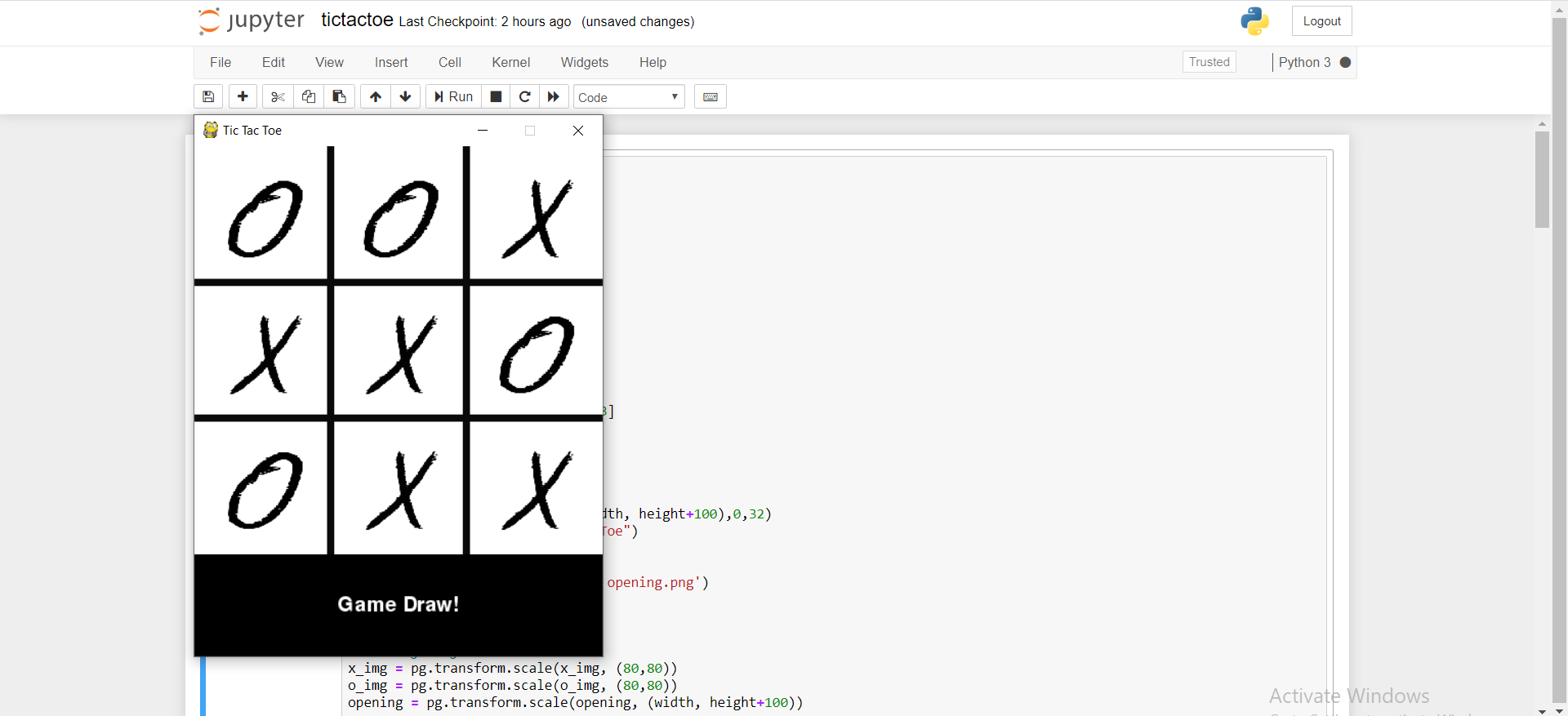
1. When X wins :



1. When O wins :



1. When match draws :

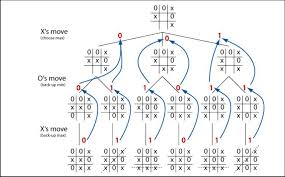


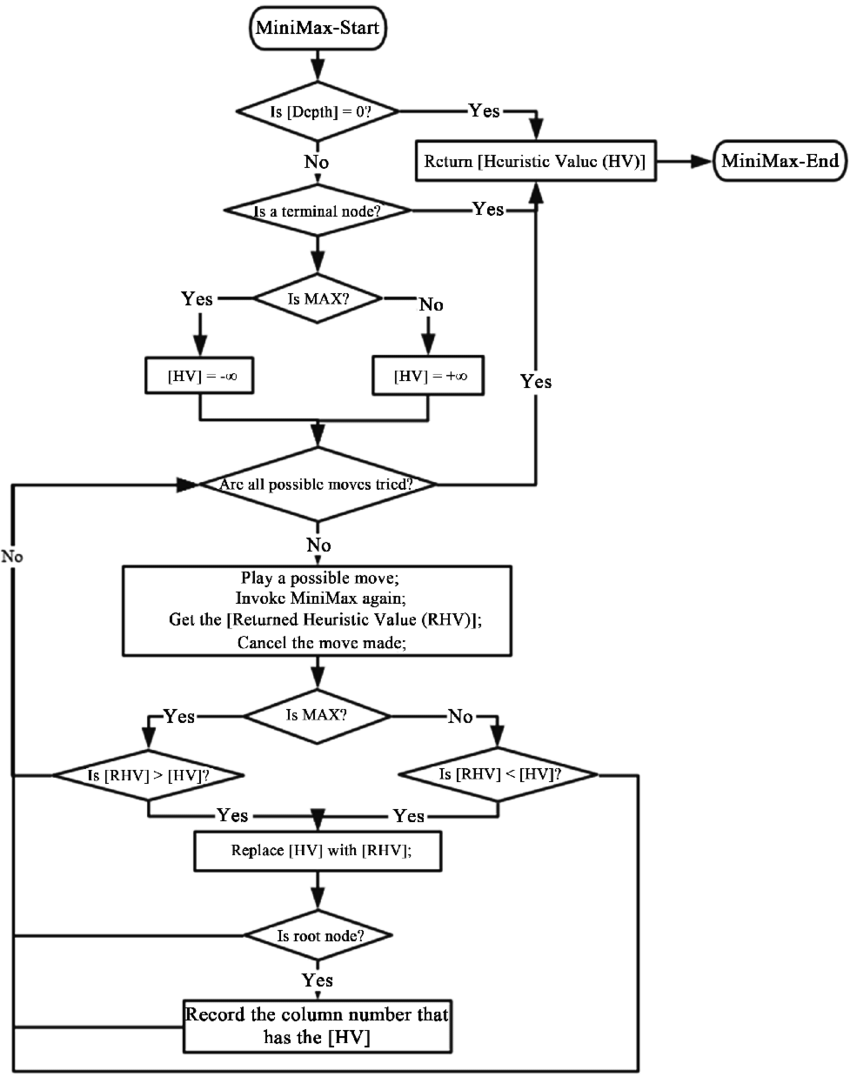
ALGORITHM :

Minimax is a recursive algorithm which is used to choose an optimal move for a player assuming that the opponent is also playing optimally. As its name suggests, its goal is to minimize the maximum loss (minimize the worst case scenario).

A Minimax algorithm can be best defined as a recursive function that does the following things:

1. return a value if a terminal state is found (+10, 0, -10)
2. go through available spots on the board
3. call the minimax function on each available spot (recursion)
4. evaluate returning values from function calls
5. and return the best value





**SOURCES :**

1)<https://www.youtube.com/watch?v=5s_lGC2sxwQ>

2)python-game-based-projects.com