

# Tic-Tac-Toe! Game

By Group11

September 16, 2024

## 1 Approach:

- We used YOLO model to our task as required, By which it depends on a dataset of labels that the model can train on it to detect the hand gesture required in the task
- As for the game implementation, it was decided to use Python's most popular games library called Pygame, the library will be used as a form of GUI while the game itself is being played through a data structure called list, basically Pygame will use the list as reference
- And finally, we will use opencv's functions to open the webcam and take a video of the user

## 2 Methodology:

### 2.1 Game implementation

#### 2.1.1 Game initialization

The program starts off with a couple of global variables, each of which can be configured to change the GUI's design, for example the SQUARESIZE is used to configure the size of each square in the Tic-Tac-Toe grid, the program will also take the from the YOLO trained model that we created.

Afterwards we initialize the Pygame screen with the configured parameters, create a variable that let the program know which player's turn is it, and create a board list that will hold all the information of the game's board, which will be later on used to draw the game.

Opencv will be used for the first time in setting up a video capture with the resolution being the same as the game screen, it will read each frame and pop it up as an image for the user to see themselves

#### 2.1.2 Drawing board

At the start of each move, the program will take the board list and use it as a reference to draw the game, each drawing will start off with turning the screen white then draw a 3x3 grid using horizontal and vertical lines.

The program then loops through the board seeking for the letters x and o, once found the program will take the index of the letter, multiply the index by the already established square size and add in half the size of the square, through that we get the center of the square grid.

Through the center we can easily draw the o using the pygame circle function, while the x needs to be a bit more precise by finding the coordinates of the starting and finishing point of 2 lines, by

getting the coordinates of the left or right upper or bottom corner of the square grid then add or subtract in a quarter of the square size.

After drawing the board the program then adds in text to guide the player on what should they do

### 2.1.3 Game's loop

At every loop, opencv will take a frame and put it into the YOLO model, the YOLO model then attempts to create boxes around the gestures the user made attempting to either label it as an x or o.

Once a box has been made the program will try to translate the coordinates of the vertices of the box and translate the return value into either x or o. once an x or o has been found the program attempts to find the center coordinates of the rectangle then translate this coordinates into which grid could it belong to.

Once found the grid and the gesture the program will see in the board list if the chosen grid is empty or not, if not empty then the game reloops again, if empty then the game let the user know which gesture was inputted, place the gesture in the list and redraw the board so that the player see the aftermath of the move.

Once the player made a move successfully the game will check if there is a game over condition, the game over conditions occurs when a row of the list are equal to each other, excluding the none, or likewise if a column of the list is equal to each other, excluding the none again, or if there is a diagonal of values equal to each other or if there is no none value in the list.

the source code of the gameover function is coded in a extremely readable and donkeywork way because the 3x3 grid is small and wasn't worth the trouble to overcomplicate it, once the game over condition has occurred the program will let the user know who won or if it's a tie and redraw the board to indicate the winning line, to quit press on the user's camera and type q

## 3 Results:

- A game that requires only one mouse click: the click that starts the program and only needs your hands to play it properly.
- The game draws every move made and lets the player know who made the move, who and how the player won
- it is not 100% accurate but it is accurate enough for the game to be playable

## 4 Challenges:

- The first challenge we faced was to collect the largest amount of photos to our dataset, which the model needed much data to be trained well
- the model needed to be more generalized, as we didn't want it for a personal use, so we varied in photos angles, brightness, and background
- designing the visuals of the game was tedious as it required some pixel calculations to make it look symmetrical

- implementing the webcam into the game was difficult, while yolo managed to detect the gestures pretty accurately, figuring out each gesture belongs to which grid was very tedious