CMSC 234

Project 2 | TicTacToe

Derya O. Kurin

**Lessons Learned:**

For the UI, I made some changes to my initial UI design. I wanted to play with the ImageViews instead of Buttons, and because I found a very nice play board png image on the internet and a circle and cross png images, I wanted to use these images to create more appealing UI for the game after finding them for free.

Since the game was between the player and the computer, creating the player moves was straight forward with an onClickListener that was registered on the ImageViews. But for the computer I needed spend some time reading articles and watching various approaches. Finally, I utilized an arrayBoard with only -1 and 0 values for the 9 positions on the board to keep track of the game internally.

This project gave me some hard time but at the end I experienced a lot of challenges from finding the best design for the requirements to creating a board game.

I wanted to improve the appearance of it and wanted to add a delay after the player move, not to have the computer move write away.

That’s why, I implemented an animation using animate and rotation and fading features while setting the view for the ImageView.

If I had more time, I would spend more time on improving the computer move. I learned that with strong algorithms I can make the game difficulty level more advanced with applying more logic on the computer moves.

I would also improve the UI design since the toast text is overlapping the created by text, but these concerns are more to do with spending more time on this app.

**Algorithm:**

Declare boolean playerTurn and assign it to true

Declare a String array of all the possible combinations to winning the game.

Declare an int array arrayBoard

Declare an int winner and assigned it to -1

(-1: Game not finished, 0: Computer wins, 1: Player wins, 2: Game tie)

Declare Method checkIfEnd() : boolean

Declare a Boolean flag and assign it to false

Declare an int I assign it to 0

While I is smaller than the arrayBoard flag remains false, when there is no place to make a move flag turns to true, return flag

Declare method computerMove: void

Declare a random number between 0 and 8

IF flag is false but there is no more place to move

Print: Game tie

ELSE

DO

Declare a random number between 0-8

WHILE that random position is occupied (-1)

Declare an image circle for the computer

IF the random position is not occupied (not 0)

SWITCH for each ImageView position

Write cases for each ImageView from 0 to 8

Set the circle to the ImageView where the random position is

(Additional feature: Use animation while setting the image)

When computer position is set, set the random position in the arrayboard to 0 (indicating it is occupied)

Negate the player turn value

Call check() to see if there is any winner after the move

IF the winner is -1 (indicating game is over)

Call playAgain and show PlayAgain button

Declare method check() : void

Declare sumOfCircles and assign it to 0

Declare sumOfCrosses and assign it to 0

Create a nested FOR loop

FOR each combination in winCombinations

Declare an int variable sign

IF sign is -1

Assign sumOfCircles and sumOfCrosses to 0

ELSE IF sign is 0

Increment sumOfCircles

ELSE

Increment sumOfCrosses

ENDFOR

Reset sumOfCircles and sumOfCrosses by assigning them to 0 again for the next check call

Declare method playAgain: void

Param: ImageView id

Create an ImageView sign with an animation(\*Extra feature)

Register onClickListener for the click event on each possible cross or circle for each grid in the 9\*9 table view

Set winner to -1 to continue to the game

Change playerTurn value to true

FOR every index element in arrayBoard

Turn its value to empty (-1)

ENDFOR

END method

Declare method step: void

Params: place:int, mean:int

Change playerTurn

Check if there is any winner by calling check()

While no winner call playAgain

Call computerMove()

END method

Declare methods for placing crosses for the player moves

For: firstRow1, firstRow2, …… secondRow1, …….thirdRow1….. thirdRow3

If playerTurn is true

IF there is a winner then return

ENDIF

IF arrayBoard for that position is occupied return

ENDIF

Create cross ImageView and set its position to the clicked place on the grid

Call step with its place from 0 to 8 as the first parameter and 1 for the mean (For cross)

END method

onCreate set arrayBoard with 9 elements and empty all the places (by assigning to -1) for each position

END onCreate

Screen Shots:

A screen shot of a computer

Description automatically generatedA screen shot of a smart phone

Description automatically generated

It was hard for me to capture the Toast message while taking the screenshot since it is only visible for a few seconds.

I could show Player Won toast message. If the game is tie or computer wins the Toast messages appear accordingly.