Tic-Tac-Toe
Java Project

Taylor Cuffie

Video Link: https://youtu.be/DERhVWR61il

The 8 methods were:

pickShape()

makeGame()

playerMove()

cpMove()

checkBoard()

checkDraw()

printGame()

checkWin()

- Board is 3x3 Two-Dimensional Array
- Player input (Letter, Number) is translated into (Row, Column)
- The playerMove() method calls the cpMove() method and vice versa. This allows the computer and player to take turns.
- Variables to hold wins, losses, draws
- Variable to hold Player Character and Computer Character
- Scanner to accept player input

makeGame()

- I used a Two-Dimensional Array
- Starts New Game by initializing each index in the array “_”
- Calls pickShape()
- Called after win, lose, draw

printGame()

- Prints Array in this format to display game board:

```
System.out.println(x[0][0]+"|"+x[0][1]+"|"+x[0][2]);
System.out.println(x[1][0]+"|"+x[1][1]+"|"+x[1][2]);
System.out.println(x[2][0]+"|"+x[2][1]+"|"+x[2][2]);
```

```
_|_|_
_|_|_
_|_|_
```

initial layout

pickShape()

- Allows User to Choose the X or O character
- If user input is not “X” of “O”, Accepts next input
- If the user Chooses the X character they get to go first
- Uses Scanner
- Calls a move method

playerMove()

- When the player moves, the character that they have chosen is entered into their selected index
- They select an index by choosing a lettered row A-C and a numbered column 1-3
- Their input is then changed to an index of the array ex. A1 = [0,0]
- Uses Scanner
- If index is out of bounds, does not exist on game board, or is not empty, takes next input

cpMove()

- Randomly Chooses a spot
- Causing the computer to call itself multiple times created a stack overflow error; So I then asked the computer to enter it's character into the next open spot
- Should prioritize winning over blocking player(only checks board for player Character if checkBoard(cshape) didn't return a value)

checkBoard()

- Primarily Used by cpMove() method
- Accepts shape parameter
- If shape is in any two connected spaces, returns value of remaining space

checkWin()

- Checks If any columns, rows or diagonals have 3 spaces with the same character
- If Yes, and the character is the player's, player wins
- If Yes, and the character is the computer's, computer wins
- Increments winNum or lossNum depending on whether or not winning character belongs to player
- Calls makeGame() method to start new Game

checkDraw()

- If there are no more empty slots the Game Restarts
- Increments drawNum
- Calls makeGame()

Future Improvements

- Use GUI instead of console
    - Choose unique characters
- Find better checkBoard() method
- Allow player to choose difficulty level
    - Maybe Hint button
    - Player Instructions
- Computer that learns players tendencies
- Play more/ Find Bugs
- Different Colors for Each Player

Other Difficulties

- Finding checkBoard() method
  - Solved by checking each row and column using for methods; attempting to limit double checks
- Ensuring that the input was in the proper format
- Switching between int, String, and Char variables
  - Problematic
- Ensuring that the computer's character was entered into an unused slot
- Finding the best times to check for wins, losses, and draws
  - Solved by playing game; and fixing code for errors
    - Ex. Ensured that checkDraw() is always called after checkWin() otherwise program might overlook wins or losses when the game board is full