

Tic-Tac-Toe

Team Name: HAR

Team Members: Helen Li, Allison Park, Riya Gupta

Period: 3

Revision Date: 5/24/2020

Program Purpose:

The purpose of this program is to entertain the user. It allows the user to play tic-tac-toe against a computer or against another person.

Target User Profile:

This program is designed for users of all ages who want to play a game of tic-tac-toe.

Feature List:

We have an option for the user to play against another user or against the computer, and the user can play multiple rounds with the program keeping track of the score between rounds. The user can play as many times as they want. This program also detects if the user puts their piece on an invalid (already taken) space and allows them to choose a new space.

Instructions:

A user opens the program and can choose to play against the computer or another player. If they choose to play with the computer, the user goes first. They can then place their symbol in certain squares by choosing the number that corresponds with the square. If they choose to play with another user, one player is Player 1 and another player is Player 2. After the round of tic-tac-toe is over, the overall score between rounds is given and the user can choose to play again or to exit.

Class List:

TicTacToe.java - An interface, has the array that represents the tic-tac-toe board used in GameBoard.java and Player.java.

GameBoard.java - Implements TicTac.java, has the score for each player, the turn, and the players. It creates two HumanPlayer objects and one ComputerPlayer object. It finds the winner, keeps track of score, and runs the game through the main method.

Player.java - Implements TicTacToe.java, has a symbol and a cell position. It has an abstract method that allows the user to move a piece, and uses its upper hierarchy status to form a basic structure for HumanPlay.java and ComputerPlayer.java's movePiece() methods.

HumanPlayer.java - Extends Player.java, and allows the user to input the position they want to move their piece to. The code is used when the user wants to play against the computer

ComputerPlayer.java - Extends Player.java, and can choose random positions for the computer's turn. The code is used when the user wants to play against the computer.

Team Responsibility:

Allison works on GameBoard.java, Riya works on Player.java and TicTacToe.java, Helen works on ComputerPlayer.java and HumanPlayer.java. Everyone works together on coding, debugging, and brainstorming ideas.

Known Bugs/Workarounds:

In one of our earlier versions, the program wasn't able to correctly handle when a user chose a position that was already taken. This became an issue when players accidentally chose an occupied position, and the game skipped over their turn. We realized that this was because of flaws in our architecture and the way that we chose to approach the overall code. (fixed)

Because of our architecture, we couldn't figure out how to properly access methods in the ComputerPlayer class. Thus we were unable to make any progress on the player vs. computer game option until we fixed our architecture with guidance from Mr. Taylor. (fixed)

Key Learnings:

At first, we put all of our code into one class before separating it out, which led us to many bugs. Our approach to class hierarchies slowed us down, and we had to rethink the entire structure. Next time, we now know to fully think about our class hierarchies and what they have/can do before we code. We also experimented with different ways of approaching a multiple-class hierarchy and learned how to successfully use arrays and polymorphism to create a fun, functioning game. We learned how to code, communicate, and work with others online.

Credit List:

Cousin - Advice on how to start and various advanced java topics.

Mr. Ed Taylor - Help on architecture of program

lab_DanceStudioSeries - Reminder on how interfaces work