Assignment 2 Software Design Document

CS2300 Section 4 Fall 2021

Brett Ford

Project Description

The program is meant to play a game with lines. The lines are displayed on a playerboard with X and O used to represent the different player moves. The size of the board, and the number of turns recorded are in a text document, as well as the moves being taken.

The turns are considered valid if they meet three criteria: they must represent a line, they must not begin or end at the same location as another line played in the previous turns, and they must not be perpendicular to another line already played in the past k turns.
If the line is valid, all the squares it traverses are flipped to that players symbol.

At each turn in the game, the squares played, the slope of the line, and the game board itself are displayed.

The game endy when there are no empty squares, or when two illegal turns are taken consecutively.

Approach

I used many modules to solve this assignment. This allowed me to solve one problem at a time. I used the equation 3.9 found on page 42 of the practical linear algebra textbook to determine what cells a line passed through. I used nested loops to traverse the board several times.

Detailed Design

I wrote this language in java.

It has 7 modules that do most of the work.

printBoard
This method uses nested for loops to print the gameboard to the standard output

takeMove
This is invoked each time a move is taken, it calls other methods as appropriate

checkValid
This method checks the validity of each move, if it is invalid, it returns false, and the takeMove function does not place it on the board

hasWon
This method uses nested for loops to check if there are any empty squares on the board. It returns true if the board is completely full and the game is over.

calculatePoints
This method calculates how many points a player has. This is returned at the end of the game. It receives what player it is checking for as an argument.

placeOnBoard
This method places moves onto the board. It places either an X or an O depending on which players turn it is.

MakeLine
This method checks every square on the board to see if their distance from the line is greater than $\sqrt{2} \div 2$. If it is smaller, then the line goes through that square, and it is passed to the placeOnBoard method to be marked.

Flowchart showing how the modules interact (not needed if you only have one module)

This program uses two data structures to understand the gamestate. The first is a matrix called gameboard that holds the moves that have been made. It is an array whose size is dictated by the text document. It begins filled with underscores to represent empty spaces, but as the game is plates they are replaced with X or O to represent the played lines.

The second is an arrayList of objects named record. The object is defined in the TurnHistory class. Each object in this class is effectively an array with six pieces of information:

- The starting column
- The ending column
- The starting row
- The ending row
- The slope of the line
- The slope of the perpendicular line

This is used to check the validity of a move. Because the validity of moves is only considered if the number of moves is fewer than K, the record has its oldest elements removed if it is larger than K.