CPSC 2150 Project 4 Report

Skyler Wolf (sWolf-7)
Samuel Jordan (srjorda)
Robby Infinger (RobbyInfinger)
Nicholas Jordan (BallyhooAndBigTop)

Requirements Analysis

Functional Requirements:

1. As a player, I want to define how many tokens needed in a row to win, so I can change the difficulty level.
2. As a player, I want to be able to choose between a fast game implementation, and a memory efficient game, so my computer will be able to handle running the game.
3. As a player, I want to be able to choose the number of rows, so I can change the difficulty level.
4. As a player, I want to be able to choose the number of columns, so I can change the difficulty level.
5. As a player, I want to be able to choose the number of players, so I can play with more friends simultaneously.
6. As a player, I want to be able to specify my own marker so that I can play with whichever character I want.
7. As a player, I want to be prompted again after choosing a taken marker so that all of the markers will be unique and differentiable.
8. As a player, I need to be able to win anytime I connect enough tokens to win in a horizontal direction so I can have a goal to work towards.
9. As a player, I need to be able to win anytime I connect enough tokens to win in a vertical direction so I can have a goal to work towards.
10. As a player, I need to be able to win anytime I connect enough tokens to win in a diagonal direction so I can have a goal to work towards.
11. As a player, I need the game to switch between all players' turns so that I can play with all of my friends.
12. As a player, I need the game to switch between all players' turns so that the game can progress.
13. As a player, I want to be told whose turn it is so that I don't forget.
14. As a player, I need to see the board so that I can understand where I can play.
15. As a player, I want to see an updated board every turn so I can know the current game state.
16. As a player, I need to be able to tell the game where to place my marker so I can try to win.
17. As a player, I want to see the column numbers, so I can place the markers in the correct spot.
18. As a player, I want to be prompted again after making a marker placement on a full column, so I can change my placement.
19. As a player, I want to be prompted again after making a marker placement on a non-existent column, so I can change my placement.
20. As a player, I need to know which player won, so I decide if I want to restart the game or end the program.
21. As a player, I want to see the board after someone wins so I can see the winning move.
22. As a player, I want to see the board if we tie so I can see there are no moves available.
23. As a player, I want to be able to restart the game after the game ends so I can play multiple games in one session.
24. As a player, I can change the game settings after each reset so that I can have each round be different.
25. As a player, I want to be able to end the program after the game ends so I can move on to other tasks.

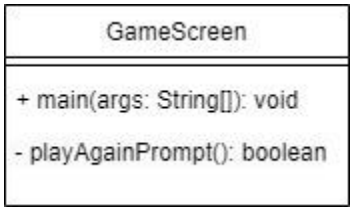
Non-Functional Requirements

1. The system must be programmed in Java.
2. The program must properly compile and run on Unix.
3. The system must run in the terminal with a command-line interface.
4. The system must be able to execute and run in a timely manner.
5. The system must be encapsulated and not stop running unless told to.
6. The system must differentiate between players via their tokens.
7. The system must start with the first token provided.
8. The system must generate a gameBoard with user defined rows and columns.
9. The system must recognize the origin (0,0) to be the bottom left of the gameBoard.
10. The system must allow the user to choose between two different memory implementations of the game, fast or memory efficient.
11. The system must be deployable by the instructed use of a Makefile

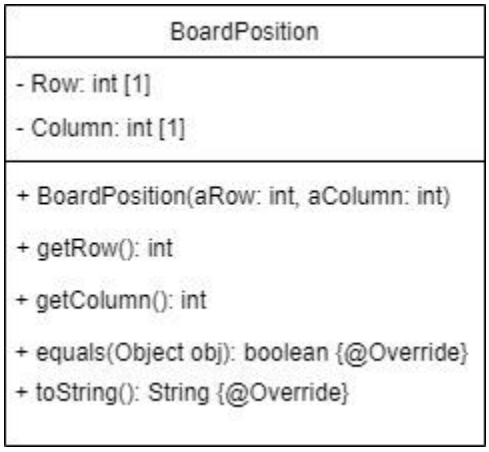
System Design

files documented: GameScreen.java , BoardPosition.java , GameBoard.java , AbsGameBoard.java , and IGameBoard.java

GameScreen.java:



BoardPosition.java:



GameBoard.java {extends} AbsGameBoard.java
GameBoardMem.java {extends} AbsGameBoard.java
AbsGameBoard.java {implements} IGameBoard.java