CPSC 2150 Project 1 Report

Jacob Colson, Warren Wasden, Steven Cabezas

Requirements Analysis

Functional Requirements:

1. As a player I need to be able to drop my token on a particular column so that I can progress the game.
2. As a player I want to drop my token in a specific place so that I can block the other player from getting five tokens in a row.
3. As a player I want to be able to drop five tokens in a diagonal line in order to win the game.
4. As a player I want to be able to drop five tokens in a horizontal line in order to win the game.
5. As a player I want to be able to drop five tokens in a vertical line so that I can win the game.
6. As a player I need to be able to select another column if a previously chosen column does not exist so that I can continue my turn.
7. As a player I need to be able to select another column if the column I chose first is full so that I can continue my turn.
8. As a player I want to be able to input a column number to place a token in so that I can use my turn.
9. As a player I want to be able to choose whether or not I would like to play again so that I can continue playing.
10. As a player I want to be able to know whose turn it is so that we don't lose track of whose turn it is.
11. As a player I want the game to allow for a maximum of 2 players so we can play against each other.
12. As a player I want to be able to interact and be informed by a command line interface generated by the game.
13. As a player I want the game to alternate between player X and player O so each player gets a fair turn.
14. As a player I want the game to display the board after each turn so that I can determine where to place my token.
15. As a player I want the game board to update each time I place a token so I don't lose track of where my tokens are.

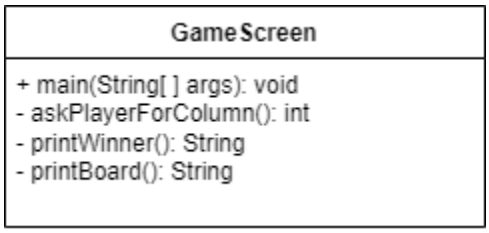
16. As a player I want the game to display the winner at the end so that we know who won.
17. As a player I want the columns to be numbered so I can easily tell which column is which.

Non-Functional Requirements

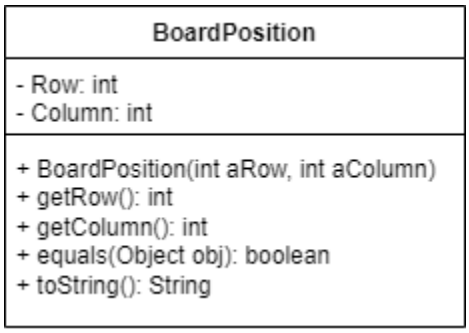
1. The game must be written in Java.
2. The game will allow for two players and alternate between player 1 and player 2.
3. Players will be able to play on a 9x7 board.
4. The game must determine if a specific space is available or already occupied.
5. Game must check if the player has won after each token is placed.
6. The game must check for a tie if neither player has won the game, then ask if they want to play again.
7. If a specific space is unavailable, the user must be asked if they would like to select another position.
8. Players must be informed of which player (X's or O's) won the game, then ask if they want to play again.
9. The game must run with a command line interface for user interaction.
10. Players need to be informed of when a selected row is full (contains 9 tokens).
11. The game must inform the player if a selected column does not exist.
12. The game must inform the user which player 1 or player 2 is X's or O's at the beginning of the game.

System Design

GameScreen:



BoardPosition:



GameBoard: