CPSC 2150 Project 1 Report

Rowan Froeschner, Graham Frazier, Terance Harrison, Kalyaan Narnamalpuram

Requirements Analysis

Functional Requirements:

1. As a player, I need the game to let me know if I picked a nonexistent or full column and allow me to try again.
2. As a player, I need the Xs and Os to drop to the most bottom available row after they select their column to imitate how gravity affects the game in the real world.
3. As a player, I need to alternate turns on the board of team X and O because the point of the game is to alternate turns and see who can win.
4. As a programmer, I will need to set the coordinates of each slot where a token can go to keep track of everything.
5. As a programmer, I need 42 different ‘ ‘ characters to represent each empty slot in the 6 x 7 board so the space can be recognized as unoccupied.
6. As a programmer, I need to create a function that runs after every turn to see if a winner can be declared due to a vertical 4 in a row
7. As a programmer, I need to create a function that runs after every turn to see if a winner can be declared due to a horizontal 4 in a row
8. As a programmer, I need to create a function that runs after every turn to see if a winner can be declared due to a diagonal 4 in a row.
9. As a programmer, I need to check if a column is full or not after every turn in order to disable another X or O being placed in that column.
10. As a programmer, I need to conclude that the game is a draw if the board completely fills up and there are no more available spaces.
11. As a player, I need the columns labeled so that I know which column is which while I am playing the game.
12. As a player, I need a way to input or select the column I want to put the marker/token on the board.
13. As a player, I need an option after the game ends to play again, so I do not need to reset the program every time I want to play again.
14. As a player, I need a tracker of how many games are won by each player (X or O) after each round.
15. As a player, I should be able to clearly see the board so i can track the pace/location of game pieces within the game.

16. As a player, it needs to be a simple board, so the game is accessible to all ages.
17. As a player, I need to know when the column is full, in case I accidentally try to place a marker in a full column.
18. As a player, I need the game to say who won before a new round starts so we can confirm the score.

Non-Functional Requirements

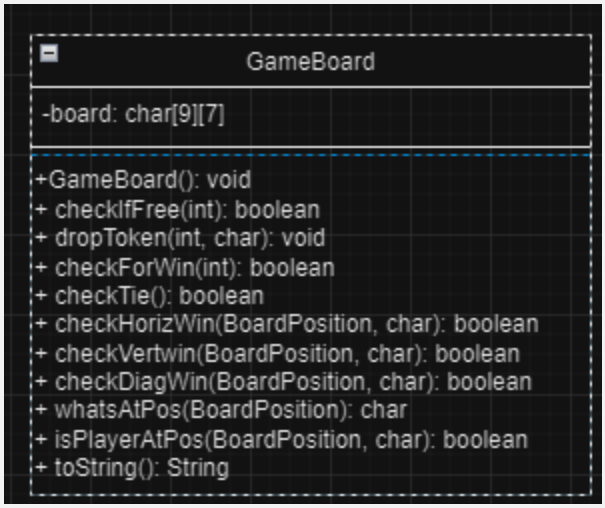
1. There needs to be a board that is 9 x 7 using lines to give the player a visual of the game board.
2. The game must always begin with “X” going first.
3. There should be messages that pop up indicating which player’s turn it is or if they need to redo their turn for playing an illegal move to help guide the game more directly.
4. The code must be in java.
5. The program must run on unix.
6. The boards bottom left corner should always begin at 0,0 to not create problems and keep uniformity.

System Design

GameBoard:

- **+ void Gameboard()**
- **+ boolean checkIfFree(int c)**
- **+ void dropToken(char p, int c)**
- **+ boolean checkForWin(int c)**
- **+ boolean checkTie()**
- **+ boolean checkHorizWin(BoardPosition pos, char p)**
- **+ boolean checkVertWin(BoardPosition pos, char p)**
- **+ boolean checkDiagWin(BoardPosition pos, char p)**
- **+ char whatsAtPos(BoardPosition pos)**
- **+ boolean isPlayerAtPos(BoardPosition pos, char player)**
- **+ String toString()**

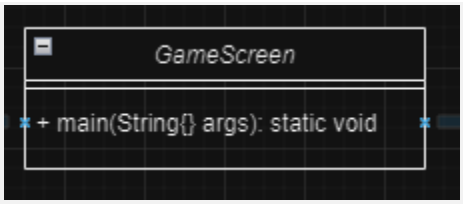
Class diagram



GameScreen:

- **+ static void main(String[] args)**

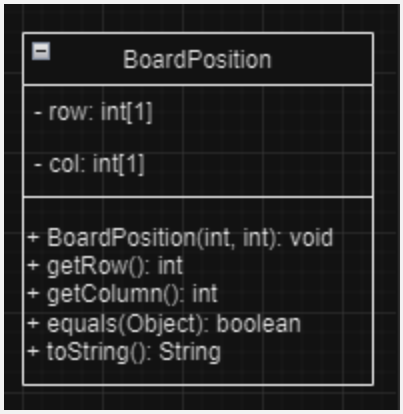
Class diagram



BoardPosition:

- **- Int Row**
- **- Int Column**
- **+ BoardPosition(int aRow, int aColumn)**
- **+ int getRow()**
- **+ int getColumn()**
- **+ boolean equals(Object obj)**
- **+ String toString()**

Class diagram


BoardPosition
- row: int[1]
- col: int[1]
+ BoardPosition(int, int): void
+ getRow(): int
+ getColumn(): int
+ equals(Object): boolean
+ toString(): String