Project 1 Report
Nicholas Berns

Functional Requirements (User Story Format):

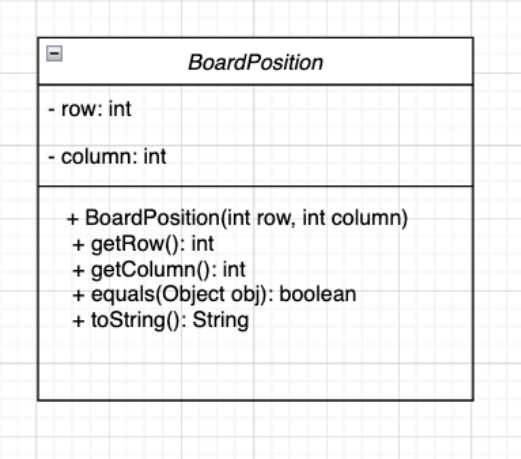
1. As a developer, I want the BoardPosition class to have private attributes row and column to encapsulate the row and column positions, ensuring data integrity and access control.
2. As a developer, I want the BoardPosition class to have a constructor that takes row and column positions as parameters, so users can initialize a BoardPosition object with specific coordinates.
3. As a developer, I want the BoardPosition class to provide getRow() and getColumn() methods, so users can retrieve the row and column positions of a BoardPosition object.
4. As a developer, I want the BoardPosition class to override the equals() method to compare two BoardPosition objects based on their row and column values, allowing easy comparison for equality.
5. As a developer, I want the BoardPosition class to override the toString() method to generate a formatted string representation in the "<row>,<column>" format, facilitating debugging and displaying positions.
6. As a user, I want the GameBoard class to have a 7x6 grid representing the Connect 4 game board, so I can play the game according to the standard Connect 4 rules.
7. As a user, I want the GameBoard class to provide a method to check if a specific column is available for placing a token, so I can make valid moves.
8. As a user, I want the GameBoard class to provide a method to place a token in a specified column, so I can make my moves during the game.
9. As a user, I want the GameBoard class to check if the last move resulted in a win for the player who made the move, so the game can accurately determine the winner.
10. As a user, I want the GameBoard class to check if the last move resulted in a tie, so the game can notify the players if the game ends in a draw.
11. As a user, I want the GameBoard class to provide a method to check if the last move resulted in five in a row horizontally, so the game can determine if a player wins.
12. As a user, I want the GameBoard class to provide a method to check if the last move resulted in five in a row vertically, so the game can determine if a player wins.
13. As a user, I want the GameBoard class to provide a method to check if the last move resulted in five in a row diagonally, so the game can determine if a player wins.
14. As a user, I want the GameBoard class to have a method to display the entire game board, so I can see the current state of the game at any time.
15. As a user, I want the GameBoard class to have a method to retrieve the token at a specific BoardPosition, so I can see what token is placed at a particular position.
16. As a user, I want the GameBoard class to have a method to check if a specific player's token is present at a given BoardPosition, so I can determine if a position belongs to a specific player.
17. As a user, I want the GameBoard class to provide a method to reset the game board, so I can start a new game with a clean slate.

18. As a user, I want the GameBoard class to have a method to display the game board history or replay the moves, so I can review and analyze the game's progress.
19. As a user, I want the GameBoard class to provide an option to save and load the game state, so I can continue a game later without losing progress.
20. As a user, I want the GameBoard class to provide an option to customize the appearance, such as changing the token colors and the board's visual theme, to personalize my gaming experience.
21. As a user, I want the GameBoard class to handle invalid moves gracefully, preventing the game from crashing or behaving unexpectedly.
22. As a developer, I want the GameBoard class to be designed with encapsulation and proper data hiding, reducing potential bugs and unintended modifications.
23. As a developer, I want the GameBoard class to have comprehensive unit tests to ensure its methods behave correctly under various scenarios.
24. As a developer, I want the GameBoard class to use efficient data structures and algorithms to handle game logic, ensuring smooth and responsive gameplay.
25. As a developer, I want the GameBoard class to provide clear and meaningful error messages when invalid actions occur, making it easier for users to understand and address any issues.

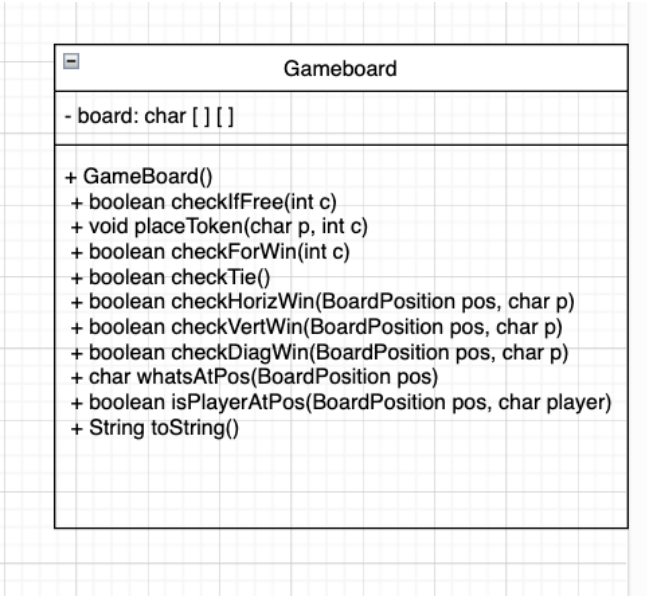
Nonfunctional Requirements:

1. Performance: The game should respond to user actions promptly and render the game board smoothly, maintaining a high level of user engagement.
2. Reliability: The GameBoard class should accurately track and determine the game's outcome, ensuring a fair and reliable gaming experience.
3. Usability: The game's user interface should be intuitive and easy to understand, allowing players to learn and enjoy the game quickly.
4. Compatibility: The game should be compatible with different platforms and devices, enabling players to access and play the game seamlessly.
5. Accessibility: The game should follow accessibility guidelines to ensure that all players, including those with disabilities, can fully participate in the gaming experience.
6. Security: The game should handle user data securely, avoiding any potential data breaches or unauthorized access to sensitive information.
7. Maintainability: The code in both BoardPosition and GameBoard classes should be well-organized, documented, and maintainable for future updates and enhancements.
8. Portability: The game should be easily portable to different environments without extensive modifications, making it versatile for deployment.
9. Loading Time: The game should load quickly and efficiently, allowing players to start playing without delays or long loading times.
10. Error Handling: The game should handle errors gracefully, providing informative error messages to assist users in resolving issues.
11. Documentation: Both BoardPosition and GameBoard classes and other classes should have comprehensive documentation, including comments and usage examples, to aid developers in understanding and using the classes effectively.
12. Performance Optimization: The GameBoard class should be optimized for memory usage and processing speed to provide a smooth gaming experience, even on low-end devices.

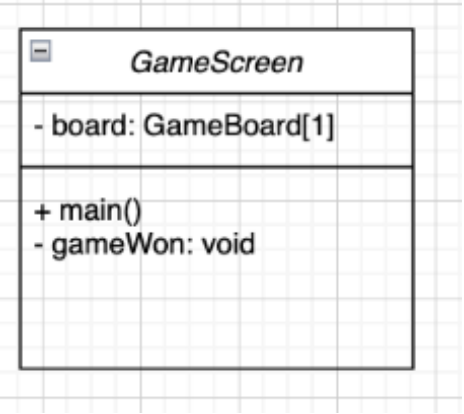
Class Diagram for BoardPosition.java:



Class Diagram for GameBoard.java:



Class Diagram for GameScreen.java



Please see the Activity Diagrams PDF submitted with this project to view the UML Activity Diagrams