Functional and Non-Functional Requirements for Game.

|  |  |
| --- | --- |
| Functional Requirements | Non-Functional Requirements |
| Installation of Unity and Microsoft Visual Studio for setting up game environment. | Cross-platform Compatibility: The game should be playable on multiple platforms, like Windows and potentially mobile devices. |
| Game mechanics: The game should allows players to control pieces on a board to ensure they move according to the rules. | User Experience (UX): The game should deliver an interactive experience for the player with balanced complexity, receptive AI and visually attractive graphics. |
| Playability: The game should be playable between human to human, and human to AI. | Resilience: The game should be able to recover from unexpected events, like crashes or power failures, retaining the game data, managing the game in consistent state. |
| User Interface (UI); The game should have an interactive UI to allow seamless gameplay between human player and RL bot. | Difficulty Level: The game should have different levels of difficulty for RL bot like Easy, Medium, and Hard. |
| Game State: The game should be able to track the game state (i.e., player turn, piece positions, win-loss-draw conditions). | Game Analytics: The game should log player Win/Loss ratio, playtime, and AI performance, for player tracking and analysis. |
| Move validation: The system should validate allowed moves based on the rules of the OrOgins game. |  |
| Automated Gameplay: The RL agent should be able to play against a human, with the capability to make moves and counter human moves efficiently |  |
| Save and Resume: The game should allow user to save and resume the game. |  |
| Game Restart: Player should be able to restart the game at any point while playing game. |  |
| Move suggestions: The game should provide hint to the user for available move for each piece to eliminate incorrect moves. |  |