Software Development Life Cycle.

There are six stages to SDLC.

1. **Requirements Gathering**

**Aim:** Automate a board game using reinforcement learning (RL) capable of playing on its own.

**Key Functionalities:**

* Game Logic Implementation (Unity).
* Integration with RL PPO agent with Unity (Python).
* Visual deployment on platforms like Windows/MacOS.

**Non-Functional Requirements:**

* Single-player experience, interactive gameplay, basic UI/UX, offline functionality.

1. **System Design**

**Game Mechanics Design:** Board layout, movement rules, elemental rules, and winning conditions.

**Architecture:**

* Unity for frontend and backend gameplay controller.
* Python for PPO model development.

**Design**

* Diagrams showing how Unity, Python, and RL components interact.
* Logic flow diagrams to control actions like moves, feedback, and AI turns.

1. **Implementation**

**Unity Game Engine:**

* Creation of game board, pieces visual assets, movement rules, elemental rules, game logic implementation, and checking Win/Loss/draw.

**Python for PPO Agent:**

* Used Stable Baselines3 for training.
* Developed, trained, tested and debugged the RL PPO model.

**Conversion of trained RL model into Unity onnx format.**

Once the model is trained in Python it will be converted into onnx format that can be used in the Unity.

**Integration of RL model into the Unity Environment**

The trained model will be loaded in the Unity game environment and merged in the Unity Environment.

1. **Testing**

**Unit Testing:** Once the model is tested into the Unity Environment. The game’s individual logics will be tested to check the overall game performance.

**Playtesting:** The game will be tested against the real player to check the game performance.

**Efficiency:** efficiency of the game will also be tested meaning whether the AI can produce a challenging experience for the user.

**Bug Testing:** The bugs will also be removed and identified in the game as they are triggered with certain actions so it can take time for them to arise.

1. **Deployment.**

Once the game passes the Testing phase it will be ready for deployment.

**Platforms:** The game will initially be exported to Windows environment that would include compiling the Unity build.

**Documentations:**

Readme files, user manual, and deployment guide will be produced.

1. **Maintenance.**

This phase begins after deployment and ensures that the system remains functional and can be improved based on feedback or issues.

* **Bug Fixes:** Any technical errors in game mechanics or model behaviour are logged and fixed.
* **User Feedback:** Testers provide feedback on gameplay difficulty, bugs, and effectiveness of gameplay.