**Tools Used**

|  |  |
| --- | --- |
| Tools | Purpose |
| Unity | Used to develop the main game with visual Graphics User Interface for the deployment of the OrOrgins game on multi-platform. |
| Pygame | Used to build the prototype to train the RL agent in the 2D simulated environment. |
| Stable Baselines3 (SB3) | Provided the PPO algorithm implementation and pre-built RL training utilities. |
| GitHub | Version control and team collaboration. |
| Microsoft Project | Used for the project management, scheduling, progress tracking , resource, cost allocation and managing the workloads. |
| Adobe Firefly | Facilitated with the creation of Board design and pieces with the help of prompts. |
| Adobe Express | Helped with the editing of the board design and the pieces. |
| Vs code | Is the IDE used for the coding the Python |
| Microsoft Visual Studio | Is the Ide used for the coding in the C# |

**Algorithm: Proximal Policy Optimization (PPO)**

* PPO is a Reinforcement Learning Algorithm introduced in 2017 by OpenAI and it has become a popular policy optimization technique due to its ability to balance simplicity, efficiency, stability and performance (Prodrampally, 2018). We used PPO due to its ability the limit by how much a policy can change, keeping the policy from changing too much. It is also more suitable for a game like Origins with complex rules because it does not require manual reward shaping.

**Model**

Our model is based on proximal-policy-optimization, but it has several components.

* **Custom game environment**: We created a custom game environment using gym.env with custom state, action and observation spaces.

**State Spaces:** we created state space, meaning everything that can be done in the game.

**Action Spaces:** Every action that can be taken in the game.  
Observation spaces: It means the information that the agent can observe currently.

**Reward Function:** we created a reward function but currently not integrated into our RL Agent.

**PPO Algorithm:** PPO algorithm to train our model.

**Method**

The approach taken for the creation of the RL Agent is CRISP DM and is based on different understandings.

**Business Understanding**: We aim to build the RL Agent in a way that it provides competitive gameplay that can be marketed and could serve as revenue generating model.

**Data Understanding:** It involves understanding of the rules of the game, how the game should work, what the space and observation spaces be.

**Data Preparation:** Application of the rules of the game, ensuring that the pieces only move according to the rules of the game. Defining the custom game environment, state, action and observation spaces, so that the RL Agent only performs as required.

**Modelling:** Creation and implementation of PPO algorithm to teach agent how to play the game.

**Evaluation:** Testing, validating and evaluation weather the trained model performs according to the game mechanics.

**Deployment:** Once the model is evaluated and works as expected, it can be converted and implemented unity version of the game.

**Theoretical Foundations for the Origins RL Agent**

Theorical foundation refers to the methods, techniques, concepts and theories that guided the creation and development of the RL Agent for Origins.

**Reinforcement Learning and its key concepts**

Reinforcement learning is machine learning technique in which the agent learns by receiving a reward for a positive action or a penalty for a negative action. The key concepts are listed below.

**Agent:** It interacts with environment and provides rewards or penalties.

**Environment:** The Environment that the agent interacts with.

**State:** The current state of the environment that the agent sees.

**Action:** The move or action the Agent can see.

**Reward:** The feedback that the Agent receives after taking an action.

**Proximal Policy Optimization**

PPO is a Reinforcement Learning Algorithm introduced in 2017 by OpenAI and it has become a popular policy optimization technique due to its ability to balance simplicity, efficiency, stability and performance (Prodrampally, 2018). We used PPO due to its ability the limit by how much a policy can change, keeping the policy from changing too much. It is also more suitable for a game like Origins with complex rules because it does not require manual reward shaping.  
**Custom Game Environment**

The Custom Game Environment is build using Open AI Gym Framework, where the Agent learns by interacting with stimulated version of the game**.**

**Key Components:**

**State Spaces:** All the possible configuration that the game can possibly exist.

**Action Spaces:** All the possible actions that the agent can take.

**Observation Spaces:** The information that the Agent sees.

**Reward Function**

**Reward Function**

Reward Function defines how, when and what rewards are given to the Agent based on the actions it takes. Positive reward for taking the desired actions and negative reward for taking the invalid or undesired ones.

**Exploration and Exploitation**

Exploration refers to the agent exploring the new strategies while exploitation refers to it refining its previous strategies. We aimed to balance exploration and exploitation for optimal performance.