

Table 1: Revision History

Date	Developer(s)	Change
Sept 18 th	Michael Ilao	Tech Stack, POC, Coding Standard
Date2	Name(s)	Description of changes
...

Development Plan

SE 4G06

Team #6, Board Gamers
Ilaom Michael, ilaom
Bedi Hargun, bedih
Dang Jeffery, dangj12
Ada Jonah, karaatan
Mai Tianzheng, mait6

[Put your blurb here. —SS]

1 Team Meeting Plan

2 Team Communication Plan

3 Team Member Roles

4 Workflow Plan

- How will you be using git, including branches, pull request, etc.?
- How will you be managing issues, including template issues, issue classification, etc.?

5 Proof of Concept Demonstration Plan

For a proof of concept demonstration, two parts of the system must be functional to a basic level. The first being the Simulation of the board game, it will not require all actions and rules to be implemented, only the core mechanics of the game. This core system should be implemented so all actions a player can make during their turn are available to be called, as if a real player is controlling them. This will allow a second system to make choices during a player's turn. This leads into the second part of a basic AI/ML Player that can control the first system. This basic player should be able to make intelligent choices based on the game state and past game states.

If these two systems can be implemented, the project will have a great chance of success as the architecture of the system will be able to support improvements to the AI/ML Player as well as implementing the rest of the game mechanics.

6 Technology

- Python will be used to develop the simulation engine and be used for simulation the AI players. The choice of this language is due to Python's Machine Learning/Artificial Intelligence libraries and the support for Object Oriented Programming.
- Object Oriented Programming will be used as the design methodology as to allow multiple developer to work seamlessly on the same project and for extensibility to other possible board games.
- To ensure common programming standards, developers will use pylint to maintain the same coding style across files. VSCode and prettier will be used for automatic formatting and linting.
- Pytest will be used for integration and unit testing.
- Coverage.py will be used for code coverage as it integrates easily with pytest.
- There are no immediate plans for Continuous Integration/Continuous Deployment as the project will be used for testing by the Stakeholders, which does not need to be hosted on any cloud environment.
- timeit Python library will be used for measuring performance and time of individual modules.
- ML/AI Libraries to be used will be PyTorch, Tensor Flow, NumPy and Pandas.
- The main tools used will be VSCode and any available Python extensions.

7 Coding Standard

The programming paradigm that will be used in this project is Object Oriented Programming or OOP. This will allow to developers to structure code for re-use and extensibility. This abstract way of programming will allow for the system to be integrate into different board games. Another standard that will be used is PascalCase for Class naming and camelCase for method and variable naming. PEP8 will also be used to enforce readable code and good python practices.

8 Project Scheduling

[\[How will the project be scheduled? —SS\]](#)