Problem Statement and Goals

Table 1: Revision History

Date	Developer(s)	Change
Sept 23	Jonah Ada and Tianzheng Mai	Version 1
Sept 24	Michael Ilao	Revised Goals

1 Problem Statement

1.1 Problem

Designing a board game is a complicated process. Balancing a board game is even more complex. And, contrarily to software games that can support hot patches, "patching" a board game after it was printed to fix a mistake is at best expensive and at worst impossible. Software patches to the game mechanics and deploying a new release removing the problems can solve the issues for a virtual game but this would have been totally impossible in a non-virtual environment.

In this project, we have partnered with Bellows Intent, a studio developing a board game named "An Age Contrived" (`https://boardgamegeek.com/boardgame/357890/age-contrived`). The game is part of the "engine building" category, like Scythe (`https://boardgamegeek.com/boardgame/169786/scythe`). The objective of the studio is to use its game to support the development of an open-source engine that can support running thousands of simulations for the game, helping identify pitfalls in the game mechanics (e.g., a strategy that always win), as well as balancing the scoring system.

We will first focus on Bellows Intent's game to validate that it is possible to support the design of a game with AI-based techniques. Then we will assess the reusability of the design gaming framework by applying it to another game (e.g., Scythe).

1.2 Inputs and Outputs

Inputs will consist of different variables which the player can manipulate during the game.

Output will consist of the effect of the user input on the overall game state. The main purpose is to determine the user input's effect on the game state and find loopholes in the game rule to fix them before publishing the game to public.

1.3 Stakeholders

Table 2: Revision History

Bellows Intent	Publisher of The Game ”An Age Contrived”	Primary
Chris Matthew	Principal Author of ”An Age Contrived” and The Game Designer	Primary
Dr. Sebastien Mosser	Supervisor	Secondary
Dr Vladimir Reinharz	Supervisor	Secondary
Group # 6	Developers	Tertiary
Players		Tertiary

1.4 Environment

Hardware: Simulation and AI will run on Alliance Canada's (`https://alliancecan.ca/en`) computer cluster Canada Compute to test millions of different possibilities in a reasonable time frame.

Software: will consist object-oriented modelling to support the representation of the board game (game engine), command module and player agents modules which will integrate state-of-the-art libraries (such as `https://www.gymlibrary.ml/`, `https://pettingzoo.farama.org/`, `https://github.com/davidADSP/SIMPLE`) to code artificial intelligent agents able to play thousands of games.

2 Goals

2.1 Simulation and Learning

- Explanation: Implement artificial intelligent players into the system to play thousands of games efficiently to detect patterns and winning strategies.
- Importance: This is the main focus of the project to test and simulate a board game for quality assurance and balancing before release.

2.2 Efficiency

- Explanation: Keep the run time of the simulation and AI game players below a threshold where the system can be easily run on the developer's computers.

- Importance: This goal will allow all developers to run the simulation on their machines, to get the best data from the system and not have to rent out computing space which would increase costs.

2.3 Complex Problem Solving

- Explanation: The system must use AI game players to analyze complex decision paths. Each different AI should have a different decision-making process.

- Importance: This goal will make sure the output of the system is of quality to the game designers since games will be played intelligently and with different strategies.

Quality Data

- Explanation: The system must track and save decisions, game states and patterns throughout the simulations, as well as compare simulations against each other to find winning strategies. The output should be easily readable for non-technical users to understand.

- Importance: This goal will make sure the game designers can use the system to balance their game.

3 Stretch Goals

3.1 Reusability

- Explanation: The system should be highly modular, where different AI Game agents can be swapped in and out to provide different outcomes. The game engineer should also be able to be swapped in and out. The architecture should allow for a simple game like tic-tac-toe and a complex game like An Age Contrived to fit seamlessly into the system.

- Importance: This goal improves the quality of the code and should be a side-effect of the architecture. This would make the system more useful as it can test more board games than it was initially designed for.

3.2 User Interface

- Explanation: The system should have a live user interface where a user can view a game in real-time, to see what actions the AI game players are making.
- Importance: This goal will make the system easier to use and allow non-technical users to verify it is functional.

3.3 Availability

- Explanation: The system should be deployed on a server, where the it can run for a longer period of time to gather data.
- Importance: This goal will allow us to collect more data which can help further analysis of game balancing.