

Table 1: Revision History

Date	Developer(s)	Change
Apr 5th 2023	Tianzheng Mai	Project Overview
Date2	Name(s)	Description of changes
Apr 5th 2023	All	Completed rest of document

Reflection Report On An AI-based Approach to Designing Board Games

Team #6, Board Gamers

Ila Michael, ilaom

Bedi Hargun, bedih

Dang Jeffrey, dangj12

Ada Jonah, karaatan

Mai Tianzheng, mait6

April 6, 2023

The final reflection for capstone project 4G06, describing the project, key accomplishments, difficulties and how we could improve next time.

1 Project Overview

Designing and balancing board games can be a challenging and time-consuming job. By leveraging artificial intelligence, this project aims to automate and optimize the board game design process, enabling both the possibility of new and creative game designs, as well as more effective new game creation. The objective of this project is to use a board game named *An Age Contrived* to facilitate the development of an open-source engine that can support running thousands of simulations of the game, helping identify pitfalls in the game mechanics in an interactive data visualization tool, as well as balancing the scoring system.

2 Key Accomplishments

Framework: Through this project, we were successfully able to come up with a framework of tools that can be used by game designers to help balance their board games. We were able to provide the tools that are designed in a way that can facilitate different board games and with small configurations, the game designers can visualize the data that is produced from the game engine/AI modules in the data visualizer module. Some of the graphs in particular can help a designer tremendously like the tree graph that shows the common paths that the

AI took in between simulations. This graph allows the designer to clearly visualize common move patterns and with the help of other graphs present in the visualizer, they can dive deeper into the data to further investigate it. Through the use of our tools, game designers can save thousands of dollars and much of the human effort that goes into testing and revising a board game manually. We believe this is a great outcome and an amazing accomplishment.

Project Management: As a group, we were successful in having regular meetings, dividing the work according to team member’s preferences/strengths and meeting all the deadlines. Our project management helped us stay focused on our goals and internal timelines. We put emphasis on letting all team members know if anyone has any issues that they are facing so that other team members can help them resolve it quickly. Our weekly meetings with our supervisors also helped us tremendously as they provided as valuable feedback and helped us stay on track in meeting our project goals. If it wasn’t for our great project management, we would have not been able to complete what we had planned to do which is why we view this a key accomplishment.

3 Key Problem Areas

Utilizing All Developers in Parallel Earlier: Working in parallel in terms of the different aspects of the project in the beginning. In the early stages of the project, we were not fully sure what the scope of the project was and was mainly focused on dealing with the AI libraries and finding what was the best for our project. At the time, we had developers on standby, we should have directed them to explore data-visualizing libraries earlier. Not utilizing human resources possibly could’ve had the state of our project a little better than what we finished at the end.

Dividing the Structure of the Game Engine More Effectively: An Age Contrived is already a big game engine itself and we divided the game engine in a way where one of the developers took the brunt of the developing work on the game engine itself. This led this only one developer mainly understanding the structure of the game engine, while the rest of the developers had to work around explanations from the main developer. If any assistance came from any other developers, the main developer of the game engine would have to integrate themselves afterward anyways.

AI Training and Feature Development: The AI aspect of the project was the most challenging as the group had some basic knowledge in terms of machine learning but was limited in first-hand experience. Once we had laid the basic structure and game engine set down, we lacked knowledge in tuning the system in ways that would speed up the efficiency of the training of our AI. A large time resource was sunk into tweaking the AI parameters to see what would be the most effective and garner the best results for the project. Furthermore, we had decided to implement too many features before incremental testing of the system. AI training troubleshooting is very difficult because there are many

reasons for AI training not progressing properly after the implementation of a new feature. We would have to troubleshoot whether our implementation was incorrect, the AI itself did not have access to certain actions, or the training parameters were not suitable enough for the AI.

4 What Would you Do Differently Next Time

There were many minor things we could have done better but we feel like the following three items are the changes we would have done differently to have the biggest impact on our project:

1) Getting feedback on our visualizer much earlier: We have spent a couple weeks working on a Python based visualizer which didn't result in any meaningful and scalable tool. If we were to get feedback from our supervisors earlier, we could have started developing the Visualizer V1 (a web app developed with React) in the beginning of the 2nd semester. We ended up with pretty good visualizer but if we had this tool earlier, it would have speed up the development process for An Age Contrived.

2) Moving the AI training to supercomputer earlier: We have waited until the last week and a half to start training the AI on Canada Compute's supercomputer. The reason for this is that we thought the training would only take a couple hours or max a day since Tic-Tac-Toe or other ML models we have used in our machine learning and other software courses took under a day. However, An Age Contrived was orders of magnitude more complex than what we have trained so far and it took supercomputer 5 days to generate a smart but not optimal AI model. So, if we were to start training earlier, we would have a better AI model and better data to analyze and actually have more time to use the data visualizer and potentially uncover some strategies and unbalanced rules that we haven't seen yet.

3) Freezing the code base and stopping new feature development a week earlier than we did: Till the last 7 days of the project, we were still implementing new features to the An Age Contrived game and each new feature resulted in starting the training from scratch which is a big reason why we had to use supercomputer to get some data in the last few days. If we were to stop the new feature development 2 weeks prior to our final demo (or even earlier), we could have get somewhat decent AI models in 2 weeks using our laptops as well. This mistake was mostly recovered using a supercomputer but if we didn't have that resource, we may end up with meaningless data for our target game in our final demo and this was a big project management lesson for all of us.

Overall, we learned a lot and we all have many minor regrets that we would have changed if we could have started over. But we feel proud of what we have achieved as a group and we believe all the minor and major mistakes we made will be great lessons for us in our future career.

5 Major Revisions to Documents

5.1 SRS

Major changes to the SRS was with the formal notation in section 3.1.2 The game engine subsystem. A more detailed Finite State Machine example was added to describe a potential implementation of a game engine in the system. The finite state machine is a simpler version of tic-tac-toe and describes how the state transitions happen.

5.2 Design Documents

The major feedback from Rev0 for the design document was the formatting and the formalization of the document. Our team split the design document back into the initial template given by Dr. Smith and formalized all the modules in the MIS.

5.3 VnV Plan

A major addition we added was more specified functional test cases on a board game, since our project surrounded on a generalized system the initial document suggested a generalized test case that could apply to any board game. This document has a section dedicated to Tic-Tac-Toe functional requirement tests to demonstrate how this would apply to a specific board game.