Proposal: Implement a Web-Based Strategy Game with Complex AIs

Josh S. Polkinghorn (B.A. in CS)

Alana M. Weber (B.S. in CS)

Faculty Mentor: Laurie Murphy

CSCE 499 -- Fall 2012

Abstract

The goal of our project will be to design and implement a single-player strategy game with a basic local AI and a clean, user-friendly interface. Our focus will be developing an AI of reasonable difficulty. These are the essential features of the game, but additional functionality will be added as time permits, including multiplayer capability, network, saving and loading game state, and an enhanced interface. Through this project we hope to learn about AI development techniques and strategies, about developing web user interfaces, and about the design and architecture of medium-scale software projects.

1. Introduction

Based on our desire to work with artificial intelligence and an interest in game development, we decided to implement a turn-based strategy game with simple rules, but with an emphasis on developing a complex AI component for it. We see this project as important primarily because of its educational potential.

The end goal of our project will be to create a strategy game with one or more AIs that present unique and variable challenges to human players. The target users will be anyone interested in playing the game, whatever their skill level. The target audience may have no previous knowledge of the game, so an additional goal will be to produce a clean, understandable GUI. Educationally, over the course of this project we hope to learn about AI development techniques and strategies, about developing web user interfaces, and about the design and architecture of medium-scale software projects.

Our research so far includes familiarizing ourselves with various existing strategy games in order to choose one appropriate for the project. We have also looked into different possibilities for user interfaces, such as having a web-based game, porting the game to a mobile device, or using a standard Java interface. So far, we have found the web-based interface option to be the most effective and simple solution. We have also researched which programming language should be used for the game logic, and our experience with Java makes it the natural choice.

2. Goals

Because the project has such flexibility, its scope depends heavily on what we decide to implement. Our basic goals are as follows:

- To design and implement single-player strategy world-domination game, similar to Risk.

- To implement a local, basic AI for the game
- To develop a clean and easy-to-use web-based user interface for the game

The following are additional features that will be added as time permits:

- An enhanced user interface
- Multiplayer
- Expanded difficulty settings of the AI to include multiple challenge levels
- A networked game architecture
- A system for saving/loading game state in order to better test and develop the AIs

We will consider the project exceptionally successful if the user interface is pleasing and easy to manipulate, and if the AI can win against a reasonably knowledgeable human player. We will develop the game iteratively, testing features at each step in the process.

The following are educational goals we hope to accomplish during this project:

- To explore artificial intelligence in depth
- To develop marketable skills in the production of user interfaces
- To gain experience in designing software applications
- To gain experience with popular systems and techniques in software development

3. Predicted Requirements

We do not anticipate any costly materials will be required for this project. We will use open-source software for code writing and editing, as well as books and research materials that we already have on hand. We do have an immediate need for a code repository, preferably either a PLU project account or github. If we decide to move the game to an online, networked framework, we may need hosting for a web server.

4. Predicted Challenges

There are three main immediate challenges: designing the front-end user interface, coordinating communication between the different parts of the program, and designing the AIs themselves.

Our first challenge will be selecting an appropriate framework for developing the user interface. We prefer not to use Java Swing, both because it is cumbersome and because we would like to focus on other parts of the project rather than building the GUI. We are currently in the process of researching other UI options, including web design using HTML/Javascript. The UI itself will need to be simple, aesthetically pleasing, and easy to use. We can accomplish this by gathering user feedback during the development process.

Second, arguably the largest challenge to creating a working framework will be enabling the different parts of our project to communicate with one another. If we use a web-based UI, for example, it will need to communicate with a web server, which will in turn need to communicate with the local game engine. Whatever AI client we develop will need to read and process game data as well. Ensuring that each piece of the system works as intended before attempting communications between them should allow us to overcome whatever arises. An overall challenge will be to focus on the core functionality of the project and to avoid adding non-essential features until that functionality is in place.

Finally, the trickiest challenge (and the focus of our project) will be designing and implementing the AIs themselves. We will need to spend time in the fall (or any time before we start coding the AIs) researching and planning how we will approach this. We also plan to consult faculty with knowledge in the area to take advantage of our available resources. Ultimately, some of these challenges will have to be solved by trial and error.

5. Timeline

- September: Conduct necessary research, including deciding on UI system and general architecture. The research will include UI prototyping to determine the best venue.
- October: Work on requirements document
- November: Prototyping and begin design process
- December: Finalize design document
- January: Begin implementation of core features - game logic
- February: Continue implementation - GUI and user feedback
- March: Finish implementation of core features, including AI, begin on additional objectives
- April: Testing and further AI development, extra features if time.
- May: Finish any outstanding issues and work on final presentation. Graduate.