Project Proposal: Pinochle AI

Gaming AI with Evolving Cellular Automata with Genetic Algorithms

2 Project Summary:

I will be creating a card playing AI that relies on evolving cellular automata with a genetic algorithm to make card playing decisions. This AI will develop its algorithm off a database of previous games and allow future games played against different opponents to update its algorithm. Testing and benchmarking will ensure the learning environment is effective. Finally, the AI will be ported to a mobile app so that players can play against the AI or let the AI follow along in the game being played to learn from it and provide useful suggestions.

2 Project Description:

2.1 Goal

The goal of this project is not only to develop a working, learning AI for a pinochle game, but to implement it in such a way that it can make a meaningful contribution to a game or competitively play against human players in a real-time environment. This learning AI will be a full implementation of an evolving cellular automata with genetic algorithm. The backend of this AI will be a server that provides access to a database of past games and different AIs that users can choose amongst. As the database grows the AI will continue to adjust its algorithm based on further input.

2.2 Background

Pinochle is a 3 – 6 player card game where players make calculated gambles against their opponents, usually as a team. The game relies on reading and predicting your opponents, working together with your partner, making careful predictions about future events in the game and getting lucky. I will create a few rule oriented AIs to generate a dataset but ultimately will be creating a more powerful AI with an algorithm based off learning from past games.

2.3 Design Requirements

* The project must be completed by the end of the term.
* The developer (me) must learn lots about AI and genetic algorithms because the AI must be developed effectively within the timeframe.
* Dataset: the learning algorithm will require a large dataset of different games played to base its learning off of.
* Proficient mobile development will be required to make a responsive, usable app.
* An efficient and stable server will be required to foster access to game archives and facilitate the multi-AI player games required to synthesize the game database.
  1. Implementation Plan
* The project will be completed in Java, with the exception of the server side, which will be implemented in c++
* The frameworks utilized in the project
* Android operating system and devices will be used to interact with the AI
* Android development will take place in Android Studio from Google.
* Java development will take place in Eclipse.
* Windows server development will take place in Sublime, compiling in Visual Studio.
* No borrowing of code has been planned at this point, but that will certainly change as the requirements of the project become clearer.

2.5 Plan for Testing

* The AI will originally be tested extensively against other iterations of itself and a simple, hard-coded AI I will write to run tests against the learning AI.
* At this point, data-sets will come from simulated games played out by simpler AIs on the computer, eventually enough datasets will be generated from a few AI players to provide a diverse dataset for the genetic learning algorithm to learn from.
* If I can find a source online for previous games I will add that in, but at the moment I haven’t found any.
* Experiments will be conducted with the learning AI against other iterations of itself and against a human player. A set of tests for mistakes will be written up to test for improvements and insure that the learning AI passes certain competency benchmarks.

3 Anticipated Obstacles:

* Learning Curve for Mobile Development: I don’t have any experience with mobile development, but I am eager to learn and have prepared my laptop with the necessary packages.
* Learning Curve for AI Material: I’m only a week into class and it is obvious how enormous a field AI development is. One main challenge of this project will be to do work that won’t need to be redone later and being smart and efficient about how to refactor work that has already been done.
* Database generation: I don’t currently have a database accessible to me, to I will have to generate one, sooner than later.