Creating a Board Game

Edward Tuckman. Adviser: James Glenn

1 Introduction

There were two goals for this project: create a board game that would be interesting for people to play, and then use AIs to test whether the game has a deep strategic component. Artificial agents can be used to test whether a board game is interesting to play over a large number of games. We can do this by varying the AI strategy and testing whether this significantly impacts the outcome of the games. Comparing the outcomes and game trees of these games can reveal whether the game has a deep strategic level. If the strategies have different results, players should also be able to recognize the different AI strategies and be able to play against them in different ways. After multiple iterations, the final game created was simple: The board is an arbitrary number of rings, set at 4 to allow people to play the game easily. Each ring is broken up into that many spaces. On a player’s turn, they play a piece in a spot, and rotate one of the rings. This rotates all of the pieces in that ring the same amount. A player wins when they fill in an entire ring, or a vertical column from the inner ring to the outer one. A number of rule sets and different size boards were implemented, and showed to people to see which was the most enjoyable to play. After settling on one variation, multiple AIs were trained to play the version that people enjoyed most and then played against each other and human players.

2 Designing the Game

The first part of the project was designing a game that would be fun to play and easily representable for AIs. The original game that I was attempting to build was very different than the game I wound up with. The original game was played on an eight by eight grid, and players would have to move a piece across the board while using neutral obstacles to block their opponent’s piece from doing the same. As the idea evolved, the obstacles included neutral enemies that would move either randomly or towards the closest player. Eventually, it included manipulating the board itself: players would have the ability to add walls or pits to the board, forcing both pieces to change their planned route. This is the main element that would
be incorporated into the connect game that I ended up settling on.

The problem with this original game is that it was very unintuitive. All of the versions that people tried they had a lot of trouble figuring out what the strategy should be, or even how to take their first couple turns. It was not the kind of game someone could just pick up and understand, so it had to be simplified.

The only part of the original game that made it into the final version was the idea that the board could be manipulated. In the game’s final version, the board was made into concentric rings. The players have to connect either all of the spots inside the same ring or one spot in a line across all rings. After they play a piece, they can rotate one of the rings to a more favorable position. This preserved the idea of changing the board strategically, but was also a lot easier for people to pick up immediately. After trying a few variations of board size, people seem to find a four by four grid the easiest to deal with. In addition, players have to rotate a ring on each of their turns, as opposed to only having the option or having to choose between playing or rotating in one turn. The different variations considered are included in the app. People continued to enjoy playing the game, and find it interesting even after playing against other people for five or ten games. This meant it was time to move forward.

3 App Description

The android app made for this game is fairly simple. It opens to a menu screen, that lets the player decide on a number of options. The player first sets the game board size, as the game works for any number of rings and spaces. I limited the number for now, as the larger boards are effectively unplayable: they take too long to play and don’t add much to the experience. The player can then decide which game mode they want to play and the AIs. They can pick whether they want to play one of the AIs, or play two AIs against each other. Once they pick their options and launch the game, they are presented with a board and can play the game. At any point they can undo their moves or an AI move to try something different, if they think they can get around whatever it is the AI tried to do.
4 Writing the AIs

There are three kinds of AIs that were trained on the game, plus a player that plays randomly. In this section I will briefly walk through each one.

4.1 Minimax

The minimax algorithm is fairly simple. When a move is requested from the agent it looks ahead in the game tree some number of moves. At each move it decides: if it is the agent’s turn, it picks the move that maximizes the value of the state. If it is the opponent’s turn, it picks the value that minimizes it. When it gets to the deepest depth it can explore, it uses a heuristic to determine the value of a state, instead of picking moves. The heuristic for this game checks for possible connections of 2, 3, or 4, with streaks of higher numbers being weighted much more heavily than those of lower ones. The heuristic also looks at whether rotations can create streaks, and weights those slightly lower than streaks that are currently on the board.

4.2 QLearning

QLearning attempts to learn the relative values of the various states of the game board. It does this by running large numbers of game, and keeping a knowledge base of the states it has seen. It starts by knowing which states are win states, and which ones are loss states. When it makes a transition to a state in which it knows the value, it updates the pre-transition state, and assigns it a value based on the future value: the most optimistic future value times some discount factor. This represents working towards a desired state, but also maintains that a goal may be in the distant future, and the current state may not mean a win or loss is guaranteed. States are updated with this equation: $Q(s_t, a) = (1 - \alpha) * Q(s_t, a) + \alpha (r_t + d * max(Q(s_{t+1}, a)))$ where $s$ is a state and $a$ is the action taken at that state. $r$ is the reward (win/loss) at this state, and $\alpha$ is the learning rate. QLearning has had success with other board games, so it made sense to try to implement it on a new board game. It has also recently been used in neural networks as a reinforcement learning technique to play Atari 2600 games at a level as good if not better than the best human players.

4.3 Monte Carlo

Monte Carlo is similar to QLearning, in that it simulates a large number of games. However, instead of making decisions based on previous simulations it essentially runs a large number of random games. It keeps track of which states result in the most wins, and tries to move towards states with the highest win percentages while playing normal games. Monte Carlo performed the worst of the three AIs implemented. It has been shown to converge to minimax, so it makes sense that it does not perform as well after only a
short training period. In other situations it has the advantage over minimax that it can be interrupted at any time and still yield the best move, but since we are using it to train an AI, this does not help here.

5 Conclusions

After the AIs were trained, they played against AIs of the same type and the other types in addition to real people. The training generates the most information about the beginning of the game, which turns out to be essentially useless. Even though the training spends a lot of the time in the first six turns, these turns have very little impact on how the game turns out. Therefore minimax is the best at playing the game: minimax’s weakness is at the beginning of the game when the game tree cannot be fully explored. When the decisions begin to matter, minimax can explore most if not all of the game tree and decide based on total information. If the board was bigger or games lasted longer, it could be the trained AIs would be able to do better than minimax. In the final version, the trained AIs are only used for opening playbooks, and then fall back to minimax. This does better than letting the trained AIs play for the entire game.

This leads to the conclusion that this game is actually not particularly strategically interesting. If only half (or less) of the game matters for the end state of the game, that creates a lot of wasted time. However, people enjoy playing it even from the beginning, and feel that the rotating mechanic is somewhat unique. Even though the game wound up not having many strategic layers, it is still a successful as a board game.