Kaitlyn Sandor
CPSC 290 Project Proposal
Advisor: Jay Lim
April 23rd, 2023

Evaluating Heuristics and Strategies on "The Game"

Problem:

The Game is a 4 player game of number placement, communication, and collaboration towards one common goal. The basic goal of the game is to place all cards 2-99 in one of 4 piles. There are two piles starting with 1 and moving up, and two piles starting with 100 and moving down. Once a card is placed in a pile, the only cards that can be added to the pile must come after the preceding card. For example, if 88 is placed upon 100, only cards less than 88 can be played on that pile. The same is true in reverse. If a 30 is played upon a 1, only cards greater than 30 may be played on that pile. The only exception to this rule is a difference of 10. For example, if a pile starting at 100 and moving down is currently at 32 and someone has the card 42, the player may play the card 42 and the current value of that deck would move up to 42. The same would be true in reverse, a deck starting at 1 and moving up with a current value of 22 enables a player to play the card 11 and move the value of that deck down to 11.

All players must keep 6 cards in their hand at all times. On each turn, a player must play at least two cards out of the 6 in their hand. They may play more than two. Players are allowed to communicate a desire to play on a specific pile with a specific level of desire but not reveal which cards they have specifically. Players are allowed to see every card that has been played already in order to help make their decisions. Once all of the cards in the deck are in the hands of the players, players only need to play one card per turn. The game ends when the current player cannot play any cards in their hand. The game is scored as a group with the score of the group

being the number of cards remaining to be played. The closer the score is to 0 the better the players played.

Project:

I discovered this card game over winter break and instantly thought of CPSC 474, Computational Intelligence for Games. This game is very interesting for heuristic and communication analysis. I am interested in seeing what level of communication is needed for the computers to play optimally as well as what strategies perform best. I am interested in comparing minimax, monte carlo tree search, different communication strategies, and different heuristics to analyze different strategies for The Game.

Specific heuristics include always defaulting to the best player who has a play on a pile, always defaulting to your best cards to minimize your own risk, and a combination of the two. I am also interested in using different scales of communication for the players. One idea is to have the players rate how good their play on a pile is from 1-11 (1-10 being how close to the card they are and 11 being a 10 difference card (very very good play)).

Implementation:

Creating this game was harder than I thought it would be. Creating the classes relied on a lot of my previous experiences with object oriented programming to decide on best practices. For the game, I decided on a few different types of objects. There are 4 piles, one deck (where players draw from), 4 players, one baseline strategy with common code, and 2 more specialized strategies. There are also the concept of “plays”, a desired card to place on a pile, and

Kaitlyn Sandor
CPSC 290 Project Proposal
Advisor: Jay Lim
April 23rd, 2023

"preferences", or how badly a player wants to place a card on a specific pile. These objects took me a long time to make and really stretched my understanding of object oriented topics. I am still not convinced I implemented each of them perfectly, but in the end they got the job done.

The first strategy I implemented was a purely selfish strategy. This strategy works by minimizing personal risk when choosing which cards to play. Each player calculates how much damage playing each of their cards will make on each of the piles and then chooses to play the cards that have the least damage, regardless of the cards of the other players. When encountered with a special "10" case, this player will always opt to play it, and will continue to play cards as long as they have a "streak" (the difference between cards being no more than 1).

The next strategy I implemented used rudimentary communication between players. Each player was able to communicate a "preference" for a specific pile, from 100-0 with 0 being a perfect 10 difference to help the pile. The basic operation of this strategy looks like this:

1. The current player communicates their preferences (one per pile)
2. Each player evaluates their own preferences
    a. If a player has a preference that is "better than" the current preference for a pile, that player communicates the preference. For example, "Player 1 has a preference on pile 5". This new preference is then placed as the "ideal preference" to beat by the next players
3. If the current player has top preference on a pile, play it for all available piles
    a. Otherwise, pick the pile with the least personal risk (selfish strategy from strategy 1)

4. Repeat the above procedure until the current player has played the minimum number of cards for their turn

This strategy greatly out performed my selfish player strategy. The selfish player strategy averaged a final score of 24 while the communication strategy averaged a final score of 13. However, the communication strategy could be much improved. In completing this project, I vastly underestimated how hard communication and strategy implementation would be. Some other ideas I had for strategies include taking into account how far away a player's preference was (for example, if they were not going for 3 turns it might make more sense to play your mediocre card immediately), trying to optimize for 10 differences to bring the pile down or up, communicating with less degrees of specificity (1-5 rating scale instead of exact), and factoring in changing preferences as cards are played during a singular turn. Overall, this project enabled me to draw on my knowledge from CPSC 474. In the future with more time, I would like to implement a Monte Carlo Tree Search to see how that compares to the communication and selfish play strategies.

Website:

The website for my project was completed in Flask. The site enables users to compete against 4 bots (all playing the selfish strategy) in "The Game". A future goal is to allow users to select between which version of the bot they would prefer to play against. Learning how to use Flask redirects was new for me and a very fun challenge! It is really satisfying to be able to interact with my final product on a website, something that has never been made before for this particular game.

Kaitlyn Sandor
CPSC 290 Project Proposal
Advisor: Jay Lim
April 23rd, 2023