Candidate Proposal #1

Rummy 500 with Symbolic AI Opponent

By Matt Grzenda

For my CSC 466 research project, I would like to create a game of Rummy 500 with a symbolic AI computer opponent. I have been playing Rummy 500 since I was a child, and I am already very familiar with the rules. This project would fall under the realm of puzzle and game AI, which already has extensive sources for me to use on this project.

I am choosing a symbolic AI for this project because a subsymbolic architecture is not a great fit for this type of situation. A game created by humans is full of heuristics which are used to then make a play which benefits a player. The subsymbolic architecture is better suited to mimic processes which humans do sub-consciously, such as voice recognition or word jumbles. There may be some places in the Rummy 500 game where a subsymbolic architecture may be better, such as determining how many cards to draw from the discard pile, based on how many cards are in your opponents hand. However, as a whole, a symbolic architecture will be best.

The textbook *Artificial Intelligence* by Melanie Mitchell has a section on AI for games. Although the section does not cover Rummy 500 specifically, the techniques for other games could definitely be adapted to work for my case. With this reference, among others, and the fact that I have been doing the research for this project for the last year, makes me confident that I am capable of completing it.

To track my progress, I will split this project into 8 stages and update my website accordingly. The are listed below:

1. A program that can generate a move, perform a discard, and draw a card based on no heuristics or rules whatsoever (discard, moves, and drawing cards each get their own set of heuristics)
2. A program that can discard based on a heuristic, but still have the same functionality as #1 for generating moves, and draws
3. A program that can discard and draw based on a heuristic, but still have the same functionality as #1 for generating moves
4. A program that can play the game based on pre-loaded heuristics
5. A program that can select from a set of heuristics, which is the best for this state in the game.
6. A program that can change a heuristic based on what it has learned
7. A program which can create its own heuristics based on what it learns from the games.
8. A program which will store its heuristics to file, for later use (making the AI persistent)