**CM331 – Artificial Intelligence**

**Spring 2025**

**Final Project Proposal**

**Team Members (At time of Proposal): Hayden Eddy, Joshua Anderson**

**Title: MaoBot, Mao, MaoMind, Maochine learning (it doesn’t really matter).**

**Abstract:**

To put it simply, we hope to make an AI agent that plays the card game Mao. For a ‘bad’ explanation of Mao, here is the Wikipedia page for the game: <https://en.wikipedia.org/wiki/Mao_(card_game)>. Mao is a game of abstract rules, and so this project is mainly an exploration of the uncertainty dimension of artificial agents. Because Mao is such a varied, abstract game, we will explain what we have defined for the game thus far, but first, we list the project requirements/goals:

1. Define a varied list of computer-friendly rules to be used in the game.
2. Implement Mao, so that it may be played by both humans and agents (It is a game, so it would be more fun if you could interact with it). In early development, this implementation will most likely be done with text on the command line, but the goal is to create a java swing GUI.
3. Develop an agent to play the game. The agent will be developed in stages:
   1. Reactive: The agent will play the first card from their hand that can be played. This type of agent will not work past the first round of the game. At the time of this proposal, this is already ‘finished’. Against a human player, this agent shouldn’t win in most games.
   2. Simple: The agent has limited prediction capabilities. It contains a list of possible rules and refines them as the game moves onward each round. This agent will work past the first round but is expected to win less than a human player.
   3. Advanced (Project Target): The agent performs probabilistic reasoning over possible rules, and has a strategic mind for playing cards, rather than the first applicable card found in their hand. The agent should also be capable of calling out penalties for other players. At the time of proposal, the most promising implementation for this ‘strategy module’ is a monte carlo tree search, but we haven’t covered it yet in class. This agent is expected to work well in long games, and to be able to beat human players frequently.
   4. MaoBot (If we have time): This agent performs all previous stage’s functions but also considers the strategy of other players. Any complications brought out from previous stages are expected to be smoothed out. This agent is expected to beat human players most of the time.
4. The Implementation of Mao should allow for multiple agents. The normal card game usually falls apart when above 4 players, so that will be our goal.

Additional Goals (Not Requirements):

1. Enable the game to be played by multiple human players.
   1. The game could be played over a network, with one player acting as a host.
2. Create alternate starting rulesets, such as the game starting with 0’s, 1’s, or 11’s added into the deck.
   1. The game could also allow players to save/load custom starting rulesets.
   2. Agents should consider these rulesets in their reasoning.
3. Allow the loading of new rules. These rules should be compatible with the AI’s prediction modules.

**Mao**

Now we will describe what Mao is to us, since not every game of Mao is played the same way in different groups. Mao is a card shedding game like Uno, except the rules are unspoken (and in fact, if you discuss the rules during the game, you are penalized). This does not mean there are no rules, and in fact at the start of the game (round 1), the rules are the same in every game:

1. Cards may be played on the same suit
2. Cards may be played on the same rank
3. The Deck is the standard 52 playing cards
4. Players who cannot play may draw 1 card
5. Players win when they run out of cards
6. Players must announce when they have 1 card (may not be implemented for our game)
7. Jacks are wild; Wild cards may be played at any time.
8. Jacks change the current suit to a suit of the active player’s choice.
9. Any player who plays a card incorrectly/out of turn/does not announce 1 card may be given a penalty of 1 card by any other player in the game.
10. The player who wins the game may add a new rule to the next round of their choice. They may choose to do this before or after the cards are dealt for the next round, but must do it before a card is played. Players may also choose to remove a rule in play instead of adding a new one, but this requires them to know that rule is in play. Note: removal may be too complex for AI agents.

The game state’s observability is divided into two parts:

1. The Discard Deck (Fully Observable)
2. The Deck (Not Observable)
3. Player Hands (Partially Observable)
4. The Rules (Variably Observable)

On round 1, the rules are fully observed. With each round, the rules become less concrete to each player. Some rules are more easily observed than others. For example, if you get an 11 card in your hand, it isn’t too hard to guess that the winner of the previous round chose to put them in the deck. For our project, rules must be computer friendly; That is, there can be no rules that cannot be implemented simply in our code. For example, we cannot make a penalty to be given to any player that laughs, smiles, or tells a joke, which is something that you could do in a normal game of Mao.