

# Rummy 500 Skeletal Plan

This is the first in what is most likely a series of plans for this extensive research project. This plan will give the project a title, provide an overview of the project, and define preliminary tasks.

## Project Title

Rummy 500 with Symbolic AI Opponent

## Project Overview

This will be a Rummy 500 card game written in Prolog with a symbolic AI opponent. The User will be a human playing against the computer. At first the program will be designed to play in a random fashion. No heuristics used for drawing a card, it will check to see if it has a move and will play the first one it finds. It will discard a random card from the deck. Once this is in place and working, heuristics will be added to a knowledge base for the program to use. Starting with one heuristic and working towards more than one to use.

## Task 0: Start the Paper

Start by planning out the paper, finding good reputable sources, and writing an outline, before starting to actually write it.

## Task 1: Design the Opponent Conceptually

Creating the opponent of the game is the main part of this program, so it should get the most planning. This will be a point to lay out how to make an opponent that plays Rummy 500 randomly.

## Task 2: Design the User Interface Conceptually

The way that the user interacts with the program should also get some thought.

## Task 3: Initializing the Game

Before any code can be written to implement drawing a card, making a move, or discarding, the first thing that needs to be done is to set up the game. Here I will create the representations of a deck of cards, the discard pile and each player's hand.

## Task 4: Random Drawing

The first task done during a Rummy turn is to draw a card. Since it requires a heuristic to draw from the discard pile (must know if you can make a move with a card) it can not be done randomly. So for now the opponent will just draw from the deck.

## **Task 5: Random Rummy Moves**

The program will be created to make the first legal move it can find. Starting with a program that can return a random selection of 3 cards out of the opponents hand. This will move into a final form where a move can be formed based on the rules of Rummy moves. The opponent will not be able to play on the other opponent's hand and moves, for now, will only consist of exactly 3 cards.

## **Task 6: Random Discard**

The last part of any turn is discarding. There are many heuristics to determine which card to discard, but for now a card will be chosen at random from the opponent's hand.

## **Task 7: Write the User Interface**

This may be split up into multiple tasks over time. The user interface will be text based and print out the user's hand, the moves that have been played, the discard pile, and the user's and opponent's scores.

## **Task 8: Playing Off of the User's Hand**

Now that a rummy 500 opponent could successfully play in a game making random moves, other functionality can be added to the opponent. The first of these is to allow the opponent to play off the user's hand.

## **Task 9: Discarding Heuristic**

A simple heuristic to stop the opponent from discarding a card which can be played as a move will be added to the rule base.

## **Task 10: Drawing Heuristic**

A heuristic will be added to allow the opponent to see if it can use any cards in the discard pile, draw from the discard pile if that is the case, and use the card in the move.

## **Task 11: Drawing Heuristic 2**

One of the other heuristics is to check how many cards your opponent has before taking from the discard pile. If your opponent has a low amount of cards, then you should not pick from the discard pile, unless the desired card is near the top of the discard pile. If the opponent has a moderate to high number of cards, pick from the discard pile.

## **Task 12: Move Heuristic**

This heuristic will try to find the highest scoring move in the computer opponent's hand and allow for multiple moves to be made if possible.

### **Task 13: Discard Statistics (If time)**

A lot of heuristics have been used to benefit the computer opponent. Now it is time to use some statistics to benefit it as well. The probability of an opponent being able to use your card in a move after it is discarded will be calculated. There are two types of moves in rummy so the statistics that the User has all of the cards necessary to create an original move with it have to be calculated for each card in the computer opponent's hand.