Team TMGJ: Summit Report

By Theo, Gus, Matthew and James

Summit is a game based on chance, prediction and decision making. Achieving the best result in this game requires both a degree of luck, as well as making the best decision based off estimates and often inadequate information. This report will explain how our program analyses all the available data given, creates estimates of the opponent's hands, then finally make decisions that will maximise profit.

Our program first identifies all the players in the game as well as its own position. Because all the information of the programs own hand is already known, it only needs to check the logs on the other remaining players in the game. The logs are then read, and the program estimates the dice of all the other players.

This can be done by first detecting any rolls that where rejected, if a roll was rejected it means that the player must have dice equal or greater than the value rolled. If a high roll was rejected by a player, then they must have a good hand and it becomes more likely that the optimal play is to drop out. As well as keeping track of all dice rolls rejected by opponents, dice rolls are logged into the opponents estimated hands. As the opponents pick up more dice, the accuracy of the estimated opponents hand increases. Therefore, the longer the game goes on, the better the program becomes at detecting if it will win a showdown or not against the other players.

In many of the games played, there is often a lot of unknown information about the opponent's hands. It is more profitable in the long run to assume a worst-case scenario, if the program believes it will win the showdown and doesn't, there is usually a heavy cost involved. Therefore, this program will usually only go through with a showdown there is a high chance of success.

When estimating the opponent's hand, any unknown dice that the opponent has will be estimated to be a 4, which is slightly above average, this will make it more likely that our program only participates in showdown's when it has a large lead over other players and a good chance of winning. The program will also count the quality rolls the opponent has. If the opponent has a high quality hand (full of 5's and 6's), the chance of them calling for a showdown is high, so if our player has a low hand its better to fold at that moment rather than going into the showdown and having to quit there.

The program also detects the quality of its own dice. As the hand quality increases, it becomes less likely that the hand will improve on the next roll, so at a point where all dice are 5's and 6's, its optimal to call for a showdown as continuing to roll for minimal improvement will give other players a better chance of winning. When a poor hand is detected, it is always best to fold as soon as possible to minimise any losses.

Finally, in the showdown the program will calculate the score of all estimated hands of the opponents. It will then compare those scores to its own hand and finally make the decision to stay or leave.

In conclusion, our program is effective due to its ability to analyse the data available and making cautious decisions based on incomplete data. The ‘better safe than sorry’ approach helps maximise profit in this game as folding early with a bad hand is far more profitable than trying to improve the hand only to lose in the final showdown.