Team TMGJ: Summit Diary

By Theo, Gus, Matthew and James

Monday 16th July:

- First team meeting, with an open discussion to gain a comprehensive understanding of the dice game Summit.
- An emphasis on strategies which exploit probability density functions.
- Summit has an uncanny similarity to poker.
- Working through the game with the minimum number of players (2) to keep it as simple to follow as possible.
- Adding a for loop and if statement to the takeTurn method to determine the best possible decision to be made during a roll.

Wednesday 18th July:

- Making the BestPlayer class, which we are starting with a very basic design based off a simple roll.
- Segregating the roles into categories of fold, showdown and roll. If the dice roll is low then give up, high showdown
- Created a very basic player that counts rounds and makes it more likely to pick showdown over time.
- We decided to make our own basic player to compete to find out who's is the best performing.

Wednesday 25th July:

- Modified the starter code so that it would run millions of games at once and keep the total money won to see players performance, this will make it easier in the future to compare performance between players and makes it easier to tweak settings to improve overall performance.
- Occasionally throws errors instead of giving results, probably due infinite rounds where both players are constantly rolling creating an infinite loop.
- TMGJ v1 counts how many "good dice" it has (5 or 6 dice) so if it has 5 good dice, it will go to a showdown. Basic functionality defeats roll and random player effectively but can be improved further.

Thursday 9th August:

- Performance seems to be affected by which player goes first. To fix this, in every game played, the order of each player is shuffled in our test code, this means that if two equivalent players play a million matches against each other, then they will get equal pay out, instead of the first player having an advantage.
- Have experimented by using a roll average for each of the five die but have yet to beat the existing code which uses totals. Looking to find some way of incorporating the two together to maximise the likelihood of winning.

Monday 13th August:

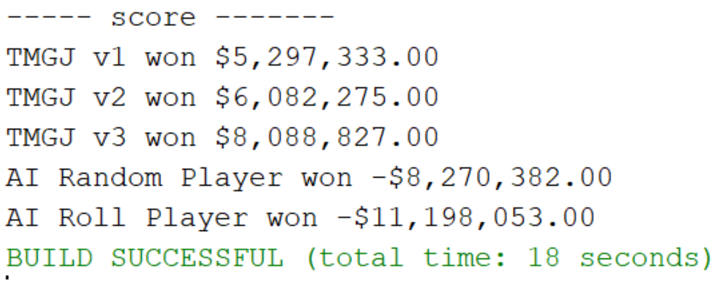
- Now includes a round counter, giving slight improvements over v1 as it makes decisions based on how far into the round it is, the longer the round the more likely the opponents have good hands. Has been named the new player as TMGJ v2.
- Improvements made by trial and error, tweaking values to improve overall score and using different opponents such as Random player, roll player and tmgjv1.

Friday 7th September:

- Added the various states into the program, so it detects and takes into consideration what other players are rolling. It will also attempt to guess the other players hands using information given by the log
- Had problems with detecting the position of the player, as the log does not tell who has dropped out.

Monday 10th September:

- Negative bets decide the player position. Starting method created to keep track of player position each game.
- Created a new team player TMGJ v3, which now estimates the opponent's hands based on the information in the logs.
- After some tweaking with v3, a 35% performance increase can be seen between v2 and v3 running 3 million games.



```
----- score -------
TMGJ v1 won $5,297,333.00
TMGJ v2 won $6,082,275.00
TMGJ v3 won $8,088,827.00
AI Random Player won -$8,270,382.00
AI Roll Player won -$11,198,053.00
BUILD SUCCESSFUL (total time: 18 seconds)
```