# **Project Diary – Falvey Group**

**17.01.2018 –** Group met upto discuss feasible data structures and ways to represent played cards on an Iota board. Concluded that an adjacency matrix built from a 2d-array would be sufficient and an attempt to start building the matrix began.

**22.01.2018 –** After class there was a more in-depth discussion on what exactly our adjacency matrix should contain in each position. A rough concept for our BoardGraph class and Spot class decided on.

A first draft of these classes was created in the afternoon and the adjacency matrix was starting to take form.

**23.01.2018 –** Pseudocode for the BoardGraph Class, Spot class and FalveyPlayer class was completed in the morning. This pseudocode stepped through each of the things we wanted to achieve at each step; from getting instances of the board to designing our getAvailableMoves() method.

**24.01.2018 –** Implementation began. Once we got started we quickly realised our Spot class was going to need more functionality as our BoardGraph class would be accessing these instances to query cards in the game (location on board, values, etc.). A neighbour structure was designed so each Spot would know who its direct neighbours were.

Our Iota strategy version 0.1 was completed here. The program could determine what available legal positions there were on the board and decide which singular card to put down to earn the highest score.

**25.01.2018 –** Goal today was to get the program to be able to evaluate multiple positions and determine whether or not more than one card could be placed on each turn.

This was achieved recursively by checking in each direction after a card was able to be placed, the method would then iterate through remaining cards in the hard and try see if any more cards could be placed. Each variation of each move was stored in a ‘potential moves’ array.

Our Iota strategy version 0.2 was completed here. The program could place up to 2 cards on each turn.

**26.01.2018 –** A small bug was stopping our program from placing more than 2 cards on each turn. Easily fixed but we noticed our game could only ever place cards in one direction i.e. it wouldn’t place cards on either side of a card even if they were on the same line.

This was fixed by having a method that adds all the permutations of available moves together and seeing if there were any other available legal plays by merging them.