Scotland Yard: The AI Component

We developed an AI for both Mr X and the Detectives.
Mr. X's AI uses a board scoring heuristic in order to determine which move to make.
A score of the board is dependent on 4 factors:

- Mr X's number of valid moves.
- His position on the map.
- His distance from the nearest detective.
- His cumulative distance from all the detectives.

Mr X's scoring heuristic uses Dijkstra's with two different approaches. The first, is that the further away he is from the nearest detective the higher the score is. We also take the total distance to all of them into consideration. We found that the problem with taking the cumulative distance only is that it would rate highly a board with a single detective very close to Mr X. We found the problem with taking the nearest detective only was that our detectives would slowly encircle Mr X and leave him with no moves.

His score increases the closer he gets to the centre of the map as he has a lower chance of getting cornered. This is modelled by the euclidean distance (coordinate to coordinate) he is from the map centre. Finally, his score is decreased by the amount of moves he will have at a given location. This allows him to avoid the arterial routeways - locations that have many moves, yet also easily accessible by detectives.

Our AI evaluates what the score of the board would be after each of Mr X's possible moves, assuming that after each move, detectives take one step closer to him. Using the Minimax algorithm, this process is repeated for each of the game states that were produced - building a tree of possible game states.

Once the algorithm reaches the lowest depth in the game tree, (terminal states) it evaluates the score and hence finds the best outcome that can result from Mr X's initial list of possible moves. It then generates a list of moves to that location for Mr. X to choose from. We optimised our AI by implementing alpha beta pruning. The AI therefore, stops evaluating a branch of the game tree once it sees a score worse that its current best.

Our detective AI is much simpler than our Mr. X's, however it proved to be very effective. Our detectives start by making random moves until Mr X reveals his location. When Mr. X is revealed, the detectives move closer to that location using Dijkstra's algorithm. Once all the detectives arrive close to Mr. X's location, they randomly move around in the surrounding area, until Mr X reveals himself again.

Testing & The Competition

When played against the random detective AI, Mr X always won. When tested against human (GUI) players, Mr X won the majority of the time. However, when tested against our detective AI, it was unpredictable as to who would win, although the detectives tended to win slightly more often.

In the competition, our AI placed 6th overall. It drew the first match and won the following two. Whilst our first opponent managed to beat our Mr. X using their detective AI, our detectives remained undefeated. Our Mr. X AI won the 4 remaining matches. It did not time out in any of the matches.

In the future, we would consider coalescing branches of the game tree that gave the same outcome for a more efficient tree. As well as this we also could have used the time that the detectives took to make decisions, to further increase the size of our game tree. Finally, we could have used multi-threading in order to non-deterministically calculate each game tree - with alpha beta pruning by having the threads to talk and kill others off. We could have also employed a more optimum scoring function by using a program that learns by running the program repeatedly and choosing score values that won the most often. Overall, we feel we put significant time and effort into this project and it was rewarding for both of us.