Hi there, my name is Will and my project is about Using Reinforcement Learning and other Algorithmic Techniques and Ideas to develop strategies for Tournament-Style Backgammon Matches.

While numerous AI models and agents exist for backgammon, they all focus on winning only individual games. However, backgammon is always played in a first-to format, such as first to 5 points.

The differences in strategy lie within the doubling cube, as well as victory by gammon and backgammon. These circumstances can grant the victor additional points, rather than just the 1 point they would receive from winning normally.

20 years since the release of TD-Gammon, a reinforcement learning model, it remains the strongest AI backgammon players, and yet is missing crucial information for tournament-style games, and also just relies on a tacked-on heuristic algorithm for the doubling cube.

This led to me to wonder if it was possible to develop a model that, in tournament-style matches, could outperform TD-Gammon.

I have hence organised my deliverables accordingly, starting with ensuring the game environment is completely stable and thoroughly testing it with a random agent before I added a GUI to make it easier to follow the game as it progressed. These have all been completed.

I then drew inspiration from Level 2 AI Search, using the logic and thought process behind the algorithms to transform them from searching algorithms to broader decision algorithms. The greedy algorithm has been completed and refined, while the genetic algorithm has been started.

Finally, my Advanced deliverables hope to include agents that can rival current and historical backgammon Ais.

For the GUI I am using Pygame as I am familiar with it and, in my opinion, it is the simplest graphical library in python that still has the capabilities I require. For developing, training and testing my models I am using PyTorch as it is the standard, and we have also been taught how to use it in Level 3 Deep Learning.

As mentioned earlier, I am revisiting some of Level 2 AI Search to inform and support the development of my intermediate algorithms, and to play my models against TD-Gammon I am using Python’s PubEval module which lets you utilise Tesauro’s TD-Gammon for benchmark testing.

So now that I’ve covered my motivation and approach, let’s understand how I’m analysing and evaluating my research.

Naturally, to determine performance of models I will be collecting win percentage and the scores of each game. In tournament-style matches average points per game a more valuable statistic than just win percentage, but I can use them both as a tiebreaker if two models are performing very similarly.

Additionally, I’m also collecting information about each turn, such as the board at the start of the turn, the dice roll, the move made, and the resulting board. At the end of each game I note how many time steps, or turns, there were.

This is so I can ensure the models and the game environment is working as expected. For example, if I saw that the random agent was actually relatively even to the greedy agent I would know something was wrong and could make necessary corrections.

Furthermore, it is reassuring to see that the board is being updated correctly and that the moves are legal within the dice roll and rules of the game, plus it allows me to track whether different agents are making the correct moves. The greedy agent is effectively predictable in every scenario, so it is very helpful to check its making the right moves.

The number of time steps are collected for two main reasons. The first being that better-performing models are likely to finish games in fewer and more consistent time-steps. This is hence another statistic that can be used in deciding between stronger performing models. The second reason being that processing time is an important factor with large-scale neural networks. Tesauro wasted hours of supercomputer processing time when he started training TD-Gammon, for example.

I collect the data I mentioned on the previous slide for hundreds of agent vs agent first-to-25 matches. I will also collect data from human participants playing against my models, making sure each player plays the same models the same number of times to avoid any skewing of data.

The aims for my strongest model are that it performs at least as well as the best performing commercial program, with further milestones being when it outperforms GBK (the best AI before TD-Gammon) as well, and then when it performs as well as TD-Gammon did upon release.

I expect the rest of my intermediate deliverables to be completed by the end of the Christmas holidays and hope to finish my advanced deliverables before Easter.