Artificial Intelligence for Simulation

# Introduction

The objectives of this coursework were to be able to create and implement a simple AI software system and to understand how the AI works. To do this, the task given was to create an artificially intelligent tank to fight other artificially intelligent tanks in a battle royal type of game. We were told to organise ourselves in groups of 5 or 6, which for us, was easy to do due to our previous collaboration work in first year.

We decided very early on that we would like to be efficient with this project. We created a Discord group to be able to converse with one another and to be able to host meetings. We then created our repository on GitHub to be able to easily collaborate with each other and to be able to push changes to the projects with many problems. Next, we set up a Trello page where we could add tasks that needed to be done, assign people to those tasks to keep track of who has done what, check any changes that needed to be done as well as check the progression of the project. All these tools allowed the group to be as efficient and effective as possible whilst keeping disruptions and problems down considerably.

[ADD PICTURES OF COMPLETED REPO AND TRELLO]

# Team Name and Roles

Once we formed the group, our first week was dedicated into creating the tools mentioned above. We had to decide on a group name and roles. Whilst on our first meeting call on Discord, we were throwing out names to see what everyone liked until Jai mentioned Russia’s greatest love machine. Everyone enjoyed the idea of the tank being Russian, and so the name of the team was set. Next, we decided on roles for the group. As the two strongest C# coders in the group, Oskar and Romans were given the roles of lead programmers. Artur was given the role of tester and programmer helper. Jai was made the project manager to look over all the roles and was given the duty of creating the documentation. Vinushan was originally given the documentation role but was mainly focusing on strategy for the tank. He also was taking down minutes for our meeting. With the roles sorted, we started by creating the repository and allowing all of us to publish to it. Next, we set up the first couple of to dos in the Trello page and assigned everyone to watch all the lecture. We also decided then that a group meeting should be held roughly biweekly with more meetings closer to the deadline. We also gave ourselves an early deadline so that if any problems were to arise, we would have time to solve them.

# State Machine

Once the lecture was release and everyone had had their lab session on finite state machines, we had a meeting to start to implement this in our AI. We first planned multiple strategies.

[PICTURES OF REJECTED FLOW CHARTS]

[PICTURE OF FINAL FLOW CHART]

This final flow diagram is one we as a group decided was the best for our strategy for our tank. This allowed us to see the number of states we would need and how to implement them as well as think about how to implement the rest of the diagram as code.

Oskar took lead on this creating most of the code for the state machine. The first was the virtual base class BaseState. This class has 3 functions that all abstract and would be overridden in classes that inherited from the class. Next, the StateMachine.cs script was written. The state machine can take all the states that are created and adds them to an internal private dictionary to be looked up when called. It can set the state of the tank, get the state the tank is in, and change the state of the tank.