

Contents

[**JOB ROLES** 3](#_Toc132961538)

[**MEETINGS** 3](#_Toc132961539)

[**MAIN PLANNED GOAL** 3](#_Toc132961540)

[**OFFENSIVE FSM** 3](#_Toc132961541)

[**DEFENSIVE FSM** 3](#_Toc132961542)

[**RBS** 3](#_Toc132961543)

[**IMPROVED GOAL** 3](#_Toc132961544)

[**PROJECT TIMELINE** 4](#_Toc132961545)

[**FSM PLAN** 5](#_Toc132961546)

[**RULED BASE SYSTEM PLAN** 6](#_Toc132961547)

[**Planned Rules** 6](#_Toc132961548)

**BEHAVIOURAL TREE**…………………………………………………………………………………………..……………………………7

[**CHALLENGES** 8](#_Toc132961549)

[**IMPROVEMENTS** 8](#_Toc132961550)

# **JOB ROLES**

**Ian Corkill – Finite State Machine (Developer)**

**Dan Lubek – Behavior Tree (Product Owner)**

**Natalia Wozniakowska – Ruled Base System (Developer)**

**Zefan Stone – General help, Presentation, Documentation (Scrum Master)**

# **MEETINGS**

The team has agreed to engage in weekly meetings on Discord to discuss ideas and work together as a team.

|  |  |
| --- | --- |
| **06/03/2023** | Discussion on how to develop FSM |
| **13/03/2023** | Further discussion on FSM |
| **24/03/2023** | Catch up meeting |
| **28/03/2023** | Bringing new ideas and working together as a group |
| **03/04/2023** | Working together |
| **06/04/2023** | Working together |
|  | Easter Break |
| **10/04/2023** | Working together main focus on FSM |
| **12/04/2023** | Working together main focus on RBS |
| **13/04/2023** | Working together FSM And RBS |
| **14/04/2023** | Working together FSM,RBS, BT |
| **16/04/2023** | Working together FSM,RBS, BT |
| **17/04/2023** | Working together FSM,RBS, BT |
| **18/04/2023** | Working together FSM,RBS, BT |
| **20/04/2023** | Working together BT |

# **MAIN PLANNED GOAL**

The team goal has been to develop both offensive and defensive FSM, RBS, and Behavior tree for the project.

**OFFENSIVE FSM** – Design logic to make the tank attack the enemy opponents without protecting the bases.

**DEFENSIVE FSM** – Design logic to make the tank camp near its base to protect it before making an attack on the enemy.

**RBS** – A structure that will use similar states as FSM but using RBS.

# **IMPROVED GOAL**

Due to complexity and concerns over the set deadline, the team has come up with a better idea to:

* Set FSM (Finite State Machine) - to operate in a general manner. With means collecting consumables when seen, attacking the enemies when spotted, and retreats when the stats are neither too low nor too high.
* Set RBS (Ruled Base-System) - to operate in an aggressive manner. Prioritizing consumables first before entering the attack state, to be able to steal any resources and use them against even if health, ammo, or fuel are low.
* Set BT (Behaviour Tree) - to operate in a defensive manner. Prioritizing consumables first, also lower the speed to conserve fuel usage.

# **PROJECT TIMELINE**

The team has kept a track of the overall development of the project. The % was noted depending on how much work has been contributed on that day.

# **FSM PLAN**

The chart represents how much has been completed on a specific day for FSM.

Before coding the finite state machine, we discussed which states we should focus on. These were the Search, Dodge, Retreat, Attack, Chase, and Ambush. The search state is the most common state, where the tank is roaming around, looking for any consumables or enemies. The dodge state is called in between the attack state, and the tank moves sporadically between shots so it becomes harder to hit. The retreat state is called when the tank reaches low health, and it runs away and looks for health, fuel, and ammo. The attack state is called when the enemy tank is in range and causes the tank to stop and shoot. The chase state is called when an enemy tank is seen but is not in range, and causes the tank to move closer to the enemy. The ambush state has the tank hide near a consumable, and when the enemy goes to collect it, the tank steals the collectible and shoots once.

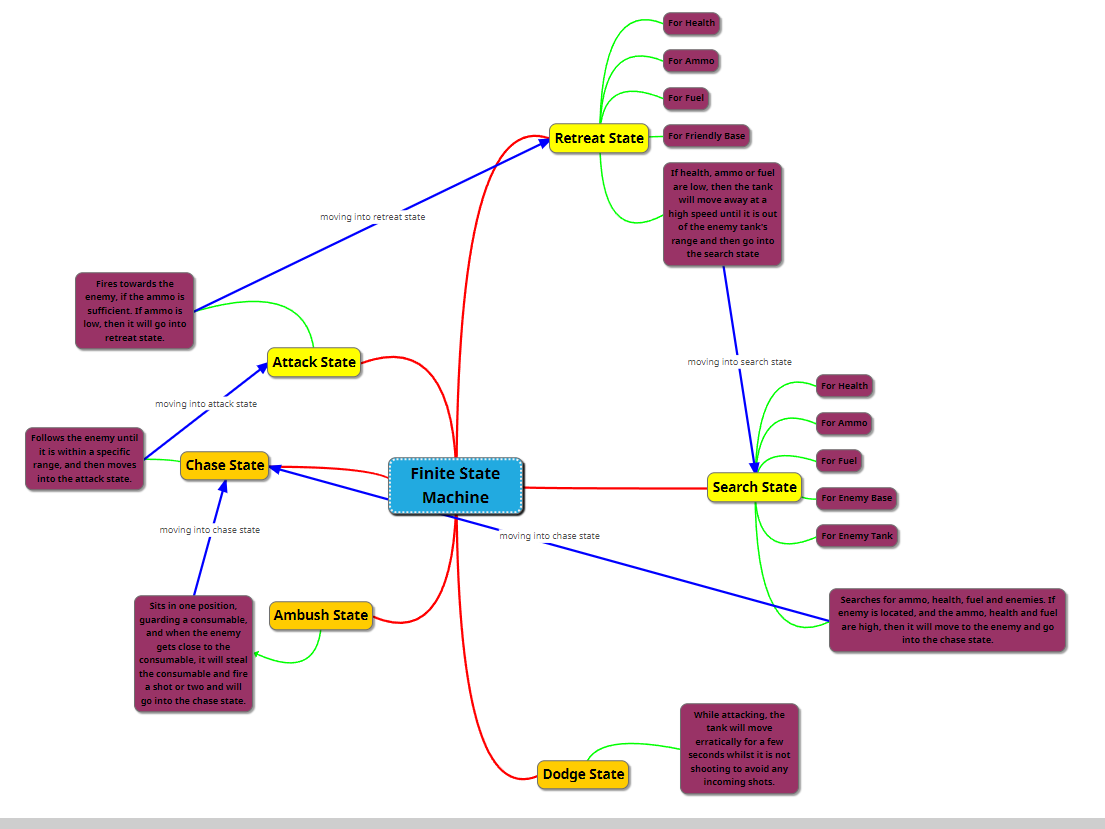
Further, we have investigated the depth of each state and how to switch from one state to another.

This is shown in the left mind map below. The mind map on the right shows the first plan for the FSM

Timeline

Description automatically generatedDiagram, timeline

Description automatically generated

The mindmap represents the final plan for FSM.

# **RULED BASE SYSTEM PLAN**

## **Planned Rules**

If the tank is in Search State **AND** the Enemy is nearby change to Chase state.

If the tank is in Search State **AND** the Base found change to Chase state.

If the tank is in Chase State **AND** the Enemy Approach change to Attack state.

If the tank is in Chase State **AND** the Base Approach change to Attack state.

If the tank is in Attack State **AND** Low Health change to Retreat State

If the tank is in Attack State **AND** Low Ammo change to Retreat State

If the tank is in Search State **AND** Low Ammo change to Retreat state

If the tank is in Search State **AND** Low Health change to Retreat state

If the tank is in Search State **AND** Low fuel change to Retreat state

If the tank is in Chase State **AND** Low fuel change to Retreat state.

The chart represents how much has been completed on a specific day for RBS.

# **BEHAVIORAL TREE**

As the Behavioural Tree implementation makes use of the Finite State Machine’s states, work could not commence until considerable progress what made on that front. However once work started, it was soon decided that since the Rule-Based System implementation created a more offensive-focused tank then the Behavioural Tree implementation should focus on a more defensive strategy. As such, the requirements for actions to take place (such as firing) are higher than in the other implementations. Sequences are used in multiple states to check that Health, Ammo and Fuel are at acceptable levels before firing or changing states. Selectors are also used when targeting either enemy bases or the enemy tank.

Since the focus of this implementation was a more defensive nature, other methods were used to achieve this. The tank will reduce its speed in certain states to reduce its fuel consumption as well as prioritising searching for consumable resources over enemies and bases. The combination of this as well as the Behavioural Tree implementation creates a tank that focuses on resource management and simply staying alive longer than the opponent, often winning matches by forcing the enemy to run out of fuel.

# **CHALLENGES**

The team has run into several types of issues when working on the project.

Firstly, getting the right values to do certain things was difficult to combine. Some functions would override the existing value and the tank would stop collecting consumables when needed. Further, the team didn’t correctly understand how to access and work with functions from the AI tank. Constant errors, trials, and research were needed which was time-consuming.

Finally, the understanding of RBS was not clear enough and the team was concerned if the project was made correctly or not. However, the team has found a way to work around it knowing the project is working as intended.

# **IMPROVEMENTS**

Fixing the challenges of combining function values proved to be time-consuming however, the team has made a progressive improvement to fix this issue. Using several different functions and calling them at different times while storing the previous value and adding them back when needed. Additionally, the team has spent considerable time researching the understanding of FSM, RBS, and TB to understand extra details about them.