

**This is the work of a student with specific learning difficulties.**

**Please mark in accordance with the guidelines.**

**Name/Student No. (whichever to be used to identify this assignment):**

**Stuart Andrew Leslie Hayes - 20363714 / CE**

**………………………………………………………………………………………**

**Course:**

**BSc (Hons) Comp Games Dev FDE**

**……………………………………………………………………………………….**

**Subject (if Combined Honours)**

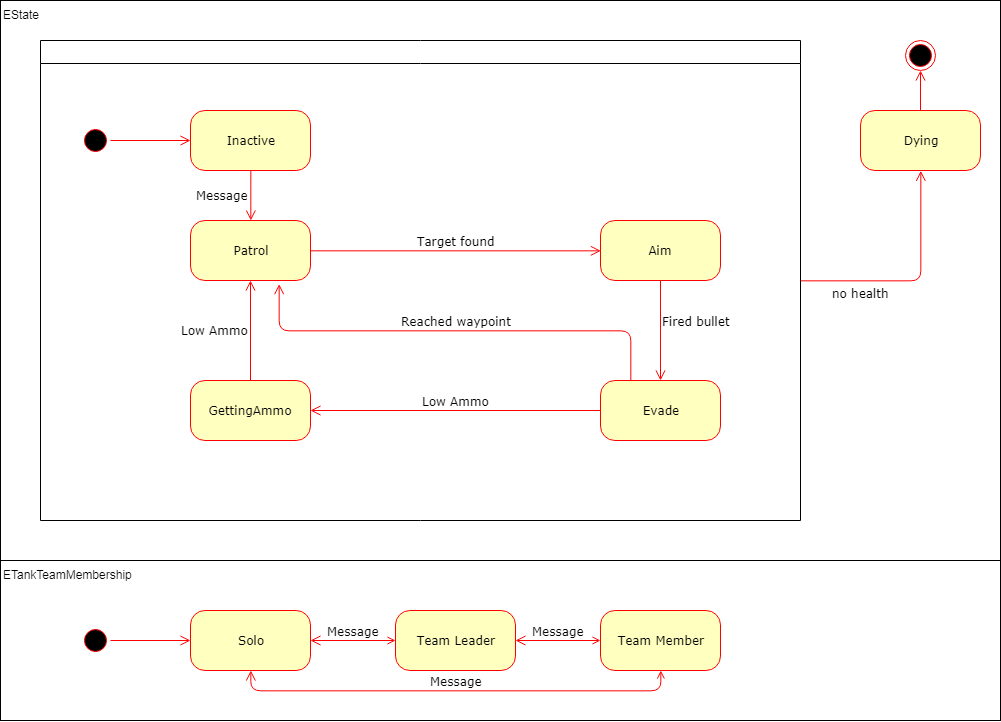
**…………………………………………………………………………………………**

# Discussion & Evaluation

## Patrol and targeting

The method that is used states, so the tank starts idle. A message is sent to the tank to go to patrol state, in this state the tank goes to its waypoint and rotates its turret. To target the turret first checks the distance is within range of the bullet, if that passes then the target is checked to see if it is within the cone of vision and the third and final check, which is the most expensive which is why its last, a ray is cast to see if a building is in the way. If all pervious check has passed the tank goes into the aim state. In the aim state the tank stops moving, starts a timer and the turret rotates towards the target at a faster rate. Once the timer runs out the tank, the tank will check the target is within distance and do one final ray to see if a building is in the way and then fire and return to patrol state.

State transition



The tank uses two state. Estate dictates what actions the tank will take while ETankTeamMembership dictates if the tank is in a team, if it’s a leader or if it’s on its own.

Estate:

Inactive makes the tank idle.

Patrol is where the tank patrols waypoints while searching for a target to shoot.

Aim causes the tank to stop and speed up aiming and focus on its target.

Evade, the tank resets its turret and runs to its waypoint.

Dying is the tanks final state, once the tank runs out of health it performs a death animation and then ends the tank.

ETankTeamMembership:

Solo, during patrol the tank finds its own waypoints.

TeamLeader, acts like solo, during patrol the tank finds its own waypoints.

TeamMember, during patrol the tank asks the TeamManager for its waypoint which is relative to the team leader

* Discuss how you would implement the same behaviour with a single ‘SceneUpdate’ function rather than separate ‘EntityUpdate’ functions. Contrast this with your implementation, considering the respective benefits and drawbacks

## Implementation

A single scene update could be done, the order in which objects would be easier to visualise the order of objects but would become very long and it would be easy to miss an object’s update, but this would become long and unorganised very quickly leading to maintenance problems. This approach is better for data-based problems while are tanks are more like objects.

Each entity having its own ‘EntityUpdate’ makes the entity more object orientated. This means code is more organised and more readable, there would be less mistakes and maintenance become much easier. This causes a bigger overhead as setting up each object separately requires more space and more calls to do the same thing. Having the code separate for each object gives better encapsulation which give better security to the objects data. Objects trying to communicate, and share data is much more difficult, a messaging system can be used to help reduce this problem. This approach is closer to how we think of objects making it clear, easier to understand and possibly easier to program.