Report

Constantinos Iasonides

366408

29/03/2019

Abstract

This is a report on Game design Elective. The course is to create a game with an implementation of an AI. An AI is Artificial Intelligence for a program or game without a player or a user. It’s like a brain of a robot. For instance, we use an NPC- Non-Player Character which in this case is my tank, moving around and destroying other 3 different NPCs in one game. In this case I used Unity’s animator, C# scripts, and assets from Tank tutorial in Unity’s website.

Table of Contents

Report Page…3

My Tank Page…4

My AI Page…4

Game Feel Page…5

Conclusion Page…6

Tables Page…7

Bibliography Page…8

**Report**

I used the tutorial of a Unity’s project to acquire the assets for our game. Using those assets, I took a tank and I worked on it. To make my tank work, I created a script called TankAI which controls the tank’s actions. Then using the animator, I added 3 behaviors, ‘Patrol’, ‘Chase’ and ‘Attack’ to make the tank search, find and shoot.

Furthermore, I used these states in Animator. The new animator controller called ‘Patrol’ has the implementation of the ‘Patrol’ state, which makes it move around and search if there is ‘something’ in its proximity. If the tank finds ‘something’ then the state will change to ‘Chase’.

The ‘Chase’ state is implemented in another animator controller called ‘Chase’. A script for this state was made to make the tank chase the ‘something’ for an infinite amount of time. If the tank doesn not find anything, the ‘Patrol’ state will be still activated. When my state finds the target and needs to destroy it then it will go to the ‘Attack’ state.

The ‘Attack’ state will destroy the target. So, again I create another animator controller called ‘Attack’. In the ‘Attack’ state, I am instantiating bullets when the tank finds the target. Then if that target is not yet destroyed and it is avoiding my tank, for example, then the state goes back to ‘Chase’ and will keep chasing the target until it is destroyed.

To make these changes I made transitions between my animator controllers (See table).

# My Tank

My tank has a script called TankAI. For instance, prefab is the bullet or where are my ‘eyes’ shoot a ray, cast and find the other Game Objects. This script has Health, Bullets, Turret (eyes), Shell and Power Launch. Here I call some functions that will be needed later in the game to help my tank execute the things that I want it to do. The tank will move around through Navmesh, a tool from Unity to help avoid any collisions with other Game Objects and follows various waypoints. My tank will spot and chase, but also destroy other tanks in the game with the help of the animator controllers and states.

### AI.

My AI follows three simple steps. Patrol, Chase and Attack. In these three different steps I will explain what is going to happen.

#### Patrol

In Patrol state, the AI will follow a path using Navmesh. During the movements, it will search with a function called checkSomething(). In this function using ‘ray cast’ in my script, it will find the target using the Layer Mask that the tanks have. If they find something, a message will appear and notify me that they have found something. Consequently, when the function has found something, the trigger needs to be changed and go into the chase state; done by the function anim.SetTrigger(“goToChase”). After I set the parameters. There the animation knows that it must change to the next state. When the function has not found anything, it resets back to Pathfinding and keeps searching for targets.

##### Chase

In Chase state, I am calling the function checkSomething() so when it finds something my animator will change the parameter to goToChase. From there, I need my tank to follow the target. Therefore, calling the tank to follow the function checkSomething(), which has already locked onto the target and it is following it. If the target is far away or there is no target at all, the state will reset back to Patrol. When the tank chases the target and needs to destroy it, then, the script calls the state Attack through the anim.SetTrigger(goToAttack). If the target is destroyed, it resets with animator.ResetTrigger(goToAttack). Next, anim.SetTrigger(goToChase) is called to chase the target or goes to anim.SetTrigger(goToPatrol) to go to pathfinding again and search for tanks.

##### Attack

In the Attack state, using my function Fire(), I am instantiating the bullet to shoot at the target. With the help of ‘ray cast’, the tank can acknowledge that there is a target present and it starts shooting. If the tank is destroyed, then it will go back to chase by using animator.ResetTrigger(goToAttack) to animator.SetTrigger(goToChase).

**Game Feel**

With the team we decided to work on various parts of the Game Feel. In the Game Feel we implemented a Main Menu, Pause Menu, Health Bars, Music, Winners panel and a better camera. All these components were implemented by my group. I created the Main Menu and the Pause menu of the game. I used Canvas and a panel with buttons, and I created a script with the Scene Manager so when the buttons are pressed the game will be played. For instance, in calling the ‘Escape’ button in the final scene a Pause menu pops up. There the buttons appear and you can choose either to resume, go to main menu or quit the game. This an extra so I can show the viewer that we implemented extra work. Health bars are to see how much life the tank has and on top of it has a canvas with the names of the tanks. The music in the background is to there to add adrenaline to the gameplay and at the end the winner appears on screen. Lastly, a better camera which allows the viewer to see all the tanks, was included.

**Conclusion**

In conclusion, me and my team have created something that is playable without players playing it. You can see four different tanks with four different AIs battling for the win. In the end, I learned how an AI works. First, it needs states to provide the brain, then it needs to know how a game object can move and lastly how to avoid and attack another game object. All these things can be implemented in many ways. One way is by making scriptable objects and then giving them states. Another way is by using the animator to show the states so it becomes more comprehensive for the coder.

**Tables/Figures**

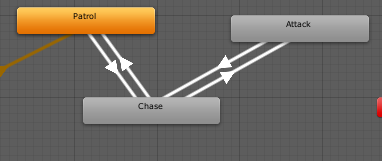


Figure 1. Here we see the transitions between the states. The three animator controllers are having transitions between them, so they know where to go next.

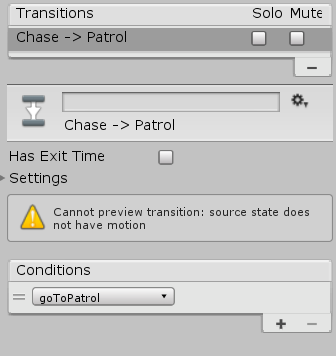


Figure 2. Here we see the transition from ‘Chase’ to ‘Patrol’. Also, it has a condition so that it knows exactly where to go.