

CMP 302- Gameplay Mechanics Report

AI state machine-based mechanic

(Patrolling Guard)

Student ID: 1904516

Student name: Arjun Bhatnagar

Overview

The application made for the completion of this coursework is an initial level of the game where the player is required to dodge the patrolling bots controlled by AI traversing throughout the map. The player is controlled by the user using the WASD keys. Also added in the game is a shooting mechanic. The player can shoot projectiles at the enemy bots using the left mouse click to destroy them. Destroying both the bots restarts the level. AI also possesses a health measure.

Link To the demonstration video: https://youtu.be/QYD_AINW3Ss

Specifications

The coursework brief requirements stated to build a unique functionality not to be modified from an existing object or system from the unreal engine. The examples and video demonstrations from previous years uploaded on YouTube provided an idea on what to build for the coursework. The decision to build an AI based state machine was satisfactory because it helped in achieving the requirements of the coursework almost fully.

The coursework consists of the following:

It is a Third-person game. The player can move around the world and must kill the AI bots patrolling the map to restart the same level. The game also consists of the player having a shooting mechanism where the player shoots projectiles at the bots to reduce their health and kill them.

Controls:

- WASD to move
- Spacebar to jump
- Left-Click of the mouse to shoot
- Arrow keys/Mouse to move the camera

The games for which this idea can work similarly is Hitman, Splinter cell, Last of Us and likewise third person stealth games.

Mechanic System

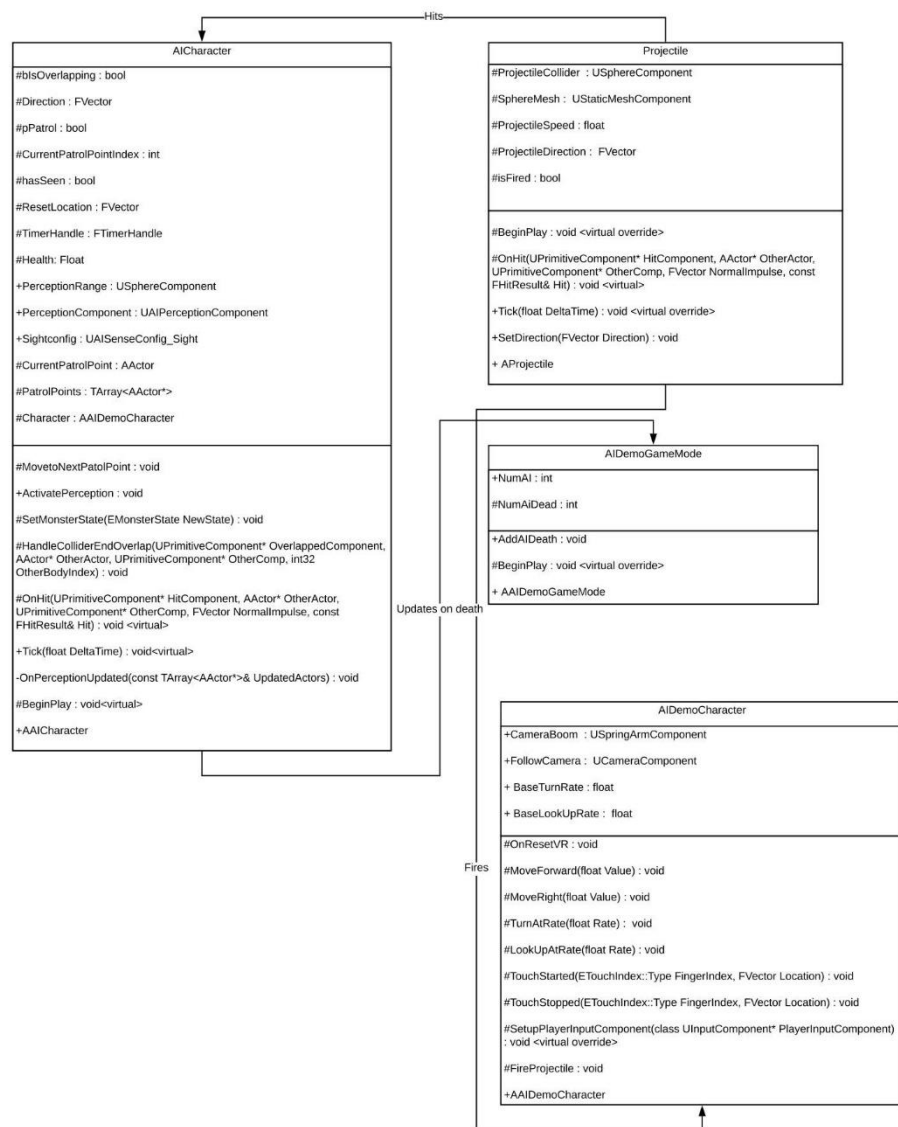
For building the game of such nature the application required for three characters. One had to be the player and the other two had to be the patrolling bots.

There are mainly three types of mechanics system being used in the construction of this application. They are as follows:

- ❖ **Enemy AI:** The Ai in the project application is using enum states for the proper working in the system. They are the Idle state, the Patrolling state, and the chasing state.
 - Idle state: This is the state where the enemies (AI) stay in their default location which is their spawn location.
 - Patrol state: The patrol state for the AI is the one where the AI keeps traversing throughout the map on defined points. These defined points are empty actors which the AI uses as a point of reference to indicate the enemy model where to go next. Since there are two enemies, there are two separate paths for the which they have to follow. One is in the shape of a square and the other is in the shape of a rhombus.

- Chase state: This state uses peripheral cones to detect where the player is. If the player is in front of the AI then it stops moving on the given path and chases the player around. It can only chase the player on which the nav mesh is made. The chasing of the player stops when
- ❖ **Projectile:** The projectile mechanic was incorporated later in the project to help give the game more structure. The .cpp and .h projectile files are responsible for the projectile mechanic where you shoot the projectile with the left click of the mouse to reduce the health of the AI enemy and kill them to win the game and essentially restart the game. The sphere component and static mesh are used to generate the projectile. The set direction function helps in identifying the camera and where the forward direction of the camera is so that a projectile can be fired in that direction only for the proper working of the shooting mechanism. The Tick function is called in the Projectile class to interpolate a projectile object at the position of the player which gets updated every frame.
- ❖ **Restart:** The restarting mechanic is implemented in the application to restart the game after the objective of the game is completed which is to kill both the enemies. When the NumAI variable becomes 0 in the program, it recognises that both the enemies have been killed and it restarts the game by spawning the player and the enemies back to their default positions. The way that the NumAI becomes 0 is when the enemy gets hit with the projectile and it reduces the health of the AI. The OnHit function made in the projectile class detects whether to decrease the health of the AI. If the AI is hit with one projectile, its health goes down by 50% and it dies in 2 hits. The game restarts when both enemy's health go down to zero.

Class Diagram



Techniques used in the development Process

The development process of the application made required some research regarding what counted as a unique game mechanic. The decision was made to develop a state-based AI game where the enemy characters chase the player if the player is spotted by one of the enemies traversing through the map. The enemy would chase the player to all the accessible areas of the map. The accessible areas in the map are made by using the actor type nav mesh bounds volume. It defines the area in the scene where the nav meshes are generated these in turn help the AI to identify which areas they can move in.

For the beginning of the development of this project, the process was started straight from C++ without much help from the blueprints provided to us in the engine. Initially the AICharacter class and header were developed in C++. The enum states defined in the header are to give specific values in the idle, patrolling and chasing states.

The AICharacter class and header is are the main classes where the bulk of the project is written in and the working of the application smoothly relies on the good structure and writing of the code inside these files.

The functions defined in the header of these files help in running the application. Before calling all the functions all the default values are also set in the AICharacter class to set the scene for the player which spawns all the obstacles, enemies (AI) and the player model onto the third person scene. To move the AI enemies from one point to another the function was used to identify empty actors in the scene which would serve as co-ordinates for the AI enemies to identify which path to go through in the map. These empty actors are known as patrol points which are stored in a list for the AI to identify and recognise which way, they must go to patrol the area.

The beginplay function in the class helps to start the game. This sets the scene because it helps in spawning the player character and the enemies on their default positions. The enemies are spawned on the first empty actor co-ordinate and the player is spawned in the middle of the map.

The tick function in the character class provides us with the functionality of helping the AI what it is looking at and perform their actions accordingly. If the player is in the periphery of the AI enemy then it starts to chase the player otherwise if there is nothing in front of the AI bot it just keeps traversing from one patrol point to the next. This function keeps updating us with the information of the positions of the player and the enemy every frame because of the movement of both the enemy and the player.

The OnHit function in the class identifies if the enemy AI has been hit with a projectile or not. If the function detects it to be true, then it reduces the health of the enemy by 50 units. Since the Overall health of the enemies is set as 100, it takes two collisions with the projectiles for the enemy to fully die.

The handlecolliderendoverlap function checks if the player is in the sight radius of the enemy or not. If not, then the enemy travers on the pre-defined path if the player is in sight, they keep on chasing him.

The projectile class in the project helps in defining and the creation of the projectile being fired at the enemy by the player.

The class creates the projectile as a sphere object which can collide with other objects and actors in the environment. This helps in the projectile working as an object which can produce an effect on another object in the environment. In this case it damages the health of the enemy AI which is required by the game to complete it.

Conclusion

In conclusion of the coursework, I would say that this project helped me understand the working of the unreal engine much better than what I would have done by myself. The development phase of the application provided me with huge challenges because I had not used the unreal engine before but the tutorials and the lectures throughout the year made the learning less daunting than what I expected. The coursework brief required from us to build a unique and complex game mechanic. The perfect definition of unique and complex was a bit blurry to me as I had no prior idea on what to do in a task like this. But seeing the examples of previous courseworks submitted by the students before me helped me a lot as they gave me a clearer idea on what could be done to fulfil the requirements in the brief. The decision to make an AI state-machine based mechanic was chosen by me because I thought it matched the coursework brief requirements and could best demonstrate the techniques and skills acquired during the running of this module. The initial idea while implementing the AI based system mechanic was just about enemies chasing the player around the map when the player gets spotted. To do so, I had to look at different types of stealth games that I had seen and played in the past. This led my decision to make the game a third person mode game instead of a first-person mode game. To make an AI actor move around the map, the nav mesh volume seemed to be the perfect choice because it could cover almost all the map and let the AI enemy roam freely wherever they wanted to. This helped in making the chasing more realistic as their movement was not restricted and they were able to follow the player anywhere the player went. To chase the player, the AI enemy first had to witness the player. This was made possible by implementing perspective cones for the enemy. The implementation of the cones worked as expected because it closely resembles the field of view for an average human being. As soon as the player was in the peripherals of the enemy, it started to work, and the enemies started to chase the player as intended. After the chasing mechanic was working, I wanted to add a shooting mechanic to the player. This was done to give the application a more game like feel and a way to reset all the elements in the level. The shooting was implemented as a basic mechanic where the player threw a projectile in the direction where he was facing. The shooting also worked as expected because it helped in colliding with the AI enemies. The collision detection helped in reducing the health of the enemy and dealing damage to them.

Overall, the application provided satisfactory results but there are still some elements that could've been polished and added to the project. If I had more knowledge about the engine and more time on my hand, I would've implemented an element of player death where if the enemy bots came too close to the player, they would've killed the player with melee attacks if the player doesn't kill them first with the projectiles.

I also should've added a gun to the player model to make it seem more realistic when the player is shooting projectiles at the enemy to make the application look closer to reality.

An element of hearing could've been added to the enemy AI so that the player wouldn't be able to sneak up on them.

References

YouTube. 2022. CMP302 Coursework - Patrolling Guard system - YouTube. [ONLINE] Available at: <https://youtu.be/LUq7kTTnFNk>. [Accessed 18 January 2022].

YouTube. 2022. CourseWork CMP302 AI Patrol Bot - YouTube. [ONLINE] Available at: <https://youtu.be/Jgl5xoylKPo>. [Accessed 18 January 2022].

docs.unrealengine.com. 2022. No page title. [ONLINE] Available at: <https://docs.unrealengine.com/4.27/en-US/Resources/Templates/ThirdPerson/>. [Accessed 18 January 2022].

docs.unrealengine.com. 2022. No page title. [ONLINE] Available at: <https://docs.unrealengine.com/4.26/en-US/API/Runtime/NavigationSystem/NavMesh/>. [Accessed 18 January 2022].