State Pattern

When I first designed the AI for my project, its behaviour was fairly straightforward but also quite rigid. The AI would detect the player and immediately switch to shooting at them, which was all hard-coded directly into the AI blueprint. While it worked for a basic setup, it felt limiting, and I knew it would make future expansions or tweaks much harder to implement.

To improve the system, I decided to adopt the **State Pattern**. Now, instead of jumping straight from detection to shooting, the AI follows a more structured flow. When it detects the player, it first transitions into a "travel" state. This means the AI calculates the player’s location and moves toward it before switching into the "attack" state. This approach not only feels more realistic but also makes the AI logic more modular, so I can add or tweak states without having to overhaul everything.

I’ve been experimenting with adding a "roam" state to make the AI more dynamic. In this state, the AI would wander around the map or patrol specific points when it doesn't have a player to engage with. Right now, I’ve sketched out the roam behaviour in the blueprint, but it hasn’t yet been integrated into the Behaviour Tree. It’s something I’m planning to focus on next because I believe it’ll add more depth to how the AI interacts with the environment.

For the attack mechanic, I added a layer of complexity. The AI doesn’t just fire blindly at the player once it’s close enough. Instead, it has to meet two specific conditions: it must be looking directly at the player, and the player must be within a certain radius. If either of these conditions isn’t met, the AI won’t shoot. This small addition makes combat feel more engaging, as players have the chance to avoid being targeted by staying out of the AI’s line of sight or quickly moving out of range.

As I continue refining the system, I’m already brainstorming some additional features.

What if the AI had a “search” state triggered when the player briefly enters its detection radius but then moves out of sight? It could look around the area or move to the last known location of the player.

I’m also considering adding a “retreat” state. If the AI’s health drops too low, it could try to find cover or even call for reinforcements from other bots in the area.

Another interesting idea would be a “flanking” behaviour. Instead of moving directly toward the player, the AI could evaluate the environment and try to approach from an unexpected angle.

Each of these ideas could integrate seamlessly into the existing state system, and because the foundation is no longer hard-coded, adding new states would feel much more manageable. I’m excited to see how these changes enhance both the gameplay and my understanding of AI design.

While I wasn’t able to implement to much state mapping to my state engine, I am happy with the overall results with the state engine from what it was to what it has become. I will continue to build the state engine, this will be useful in other projects if I ever need an AI with in those projects.