Bomber Chef State Machine

Acronyms:

AI(Artificial Intelligence)

GOAP(Goal Oriented Action Planning)

Summary:

For the AIs state machine, the main purpose of the state machine is to give a much more random feel to the AI, so the player doesn’t always know what they are going to do. The state machine is a public class that does not derive from any others.

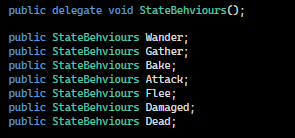
In this state machine, I created a base delegate that returned void and had zero parameters to be universal in use and expandable if prompted in the future. This delegate would serve as the base for having several instanced delegates all corresponding to the specific state in which the AI is being set to. The state machine also has a public method that is called on start to set the delegates to the corresponding methods in which the AI will use.

The State machine uses an Enum **AIStates** to determine what state the AI is currently in, this Enum contains the following states:

* Wander
* Gather
* Bake
* Attack
* Flee
* Damaged
* Dead

There is also a method **SetState** that takes in a **AIStates** as an argument sets the state and performs a certain action depending on the state using a switch statement.



A black screen with blue and green text

Description automatically generated

A computer screen shot of a black screen

Description automatically generatedA screen shot of a computer

Description automatically generated

AI Agent Implementation:

The **GoapAgent** class is what corresponds with the behaviors on the **CombatantActions** class and uses GOAP to determine the state that the AI can be in. Each agent has an instanced version of the **FiniteStateMachine** class to be able to set the states.

The Goap Agent also has a property current state which gets the instanced state machine’s state and sets the state using the **SetState** Method to create ease of use withing the class and other references of the agent.

Within the **Awake** method the agent assigns the state machines delegates using the **SetDelegates** method that takes in the following:

* **SateWander**
* **StateGather**
* **StateBake**
* **StateAttack**
* **StateFlee**
* **StateDamaged**
* **StateDead**

On **Start** the agent will be set to the wander state which when finished will transition into the **PickState** method.

The **PickState** method will go through some of the possible states the AI could do based on priority level which is set by top down order and will chose that state if applicable, if none of the states are it will either chose to wander or gather given a random chance to pick either however if gather is not available it will choose wander to prevent the AI from getting stuck.



A black screen with white text

Description automatically generatedA screenshot of a computer program

Description automatically generated



A computer screen shot of a program

Description automatically generatedA computer screen with text on it

Description automatically generated

Feedback:

I only ended up getting feedback from one person as the other said they understood it perfectly from what I explained it as and they did not think that I needed to improve it. However, the one I did receive feedback from said that it was a very simple to understand and function state machine as the state machine itself is not fulfilling any of the AI logic otherwise it is determining from the AI logic that what can be done. The changes they suggested were to change the method names that would turn into the delegate inside the state machine for a more accurate a depiction of what they do which I did end up implementing and added the **State** keyword in front ofeach of the methods for easier readability. However, there wasn’t any feedback on this, but I did notice while going through the state machine that I was still using **SetState** rather than using the current state property and just setting it as they behave the same way. I do think that that would be a worthy change down the line.

Difficulty:

However not implemented the difficulty settings that I would apply to the state machine would be a lower chance to wander and higher to gather and to add little voice lines on the lower difficulty, so the player knows what they are doing at any given time.