Simple 3D AI for a Stealth Game

Masood Taheri

Table of Contents

[Introduction 3](#_Toc138824079)

[How To 3](#_Toc138824080)

[How to use the AI system 4](#_Toc138824081)

[How to add new behaviors 6](#_Toc138824082)

[How to set up the patrol route 7](#_Toc138824083)

[how to setup sound detection radius 8](#_Toc138824084)

[Features Implemented 9](#_Toc138824085)

[FSM state Machine 10](#_Toc138824086)

[base class of enemy 10](#_Toc138824087)

[field of view 11](#_Toc138824088)

[patrolling on nodes 12](#_Toc138824089)

[Guard enemy 13](#_Toc138824090)

[Noise Alert device 13](#_Toc138824091)

[Player run foot sound 13](#_Toc138824092)

[spawnpoint controller 14](#_Toc138824093)

[Needs more Time 14](#_Toc138824094)

# Introduction

This is a documentation of a simple AI for a Stealth game that is made in Unity and c# without using any external libraries.

This a document to shows How to use some of the systems in the game and also a documentation of the subsystems that are implemented;

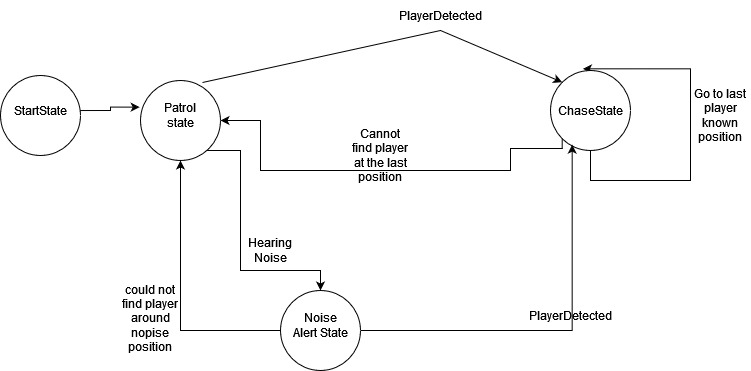
Thank you for your consideration.

Masood Taheri.

# How To

## How to use the AI system

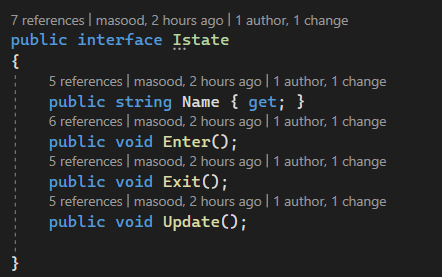
The AI system is based on the Finite State Machine Architecture(FSM). There are some different states that have different behavior and based on their decision, we can go to the other state.

The current diagram is like below:

There is a specific class name “State Machine Controller”. Its duty is to handle the states. by defining a variable of “State Machine Controller” you can have access to states and change them.

## How to add new behaviors

Every state is derived from “Istate”.



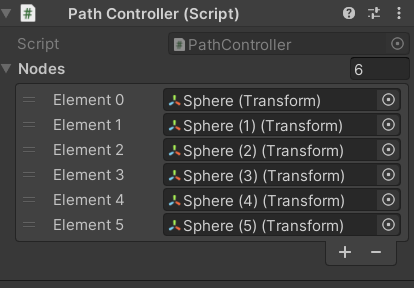
To have a new state, it's enough to create a new class and derived it from “Istate”.

Of this, you have to Implement some methods like Enter, Exit, and Update.

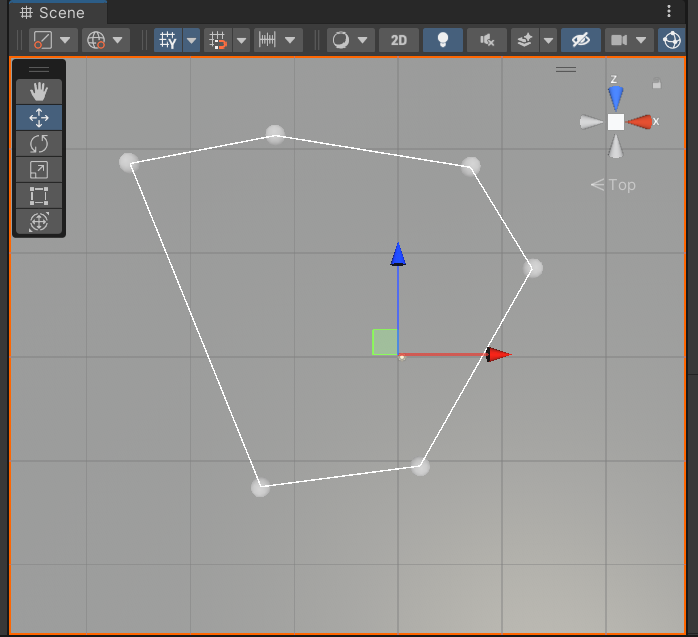
And finally, you should make a reference to this class to the “StateMachineController” class by adding a variable of its class

## How to set up the patrol route

by making an empty game object and assigning a “path controller” to it, you can have a patrol path.

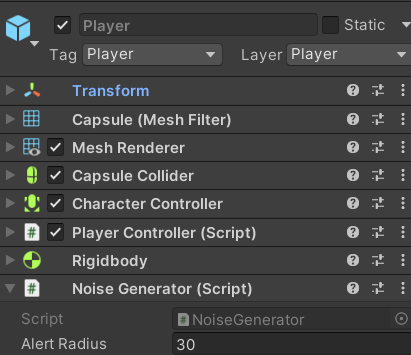


It's enough to add some new empty game objects as nodes and add them to the Nodes list. It automatically shows the position of the node in the editor and connects them to show the path like this.



## how to setup sound detection radius

player prefab has a Noise Generator component. In that component, there is an “Alert radius” field that you can modify it to change the sound detection radius:

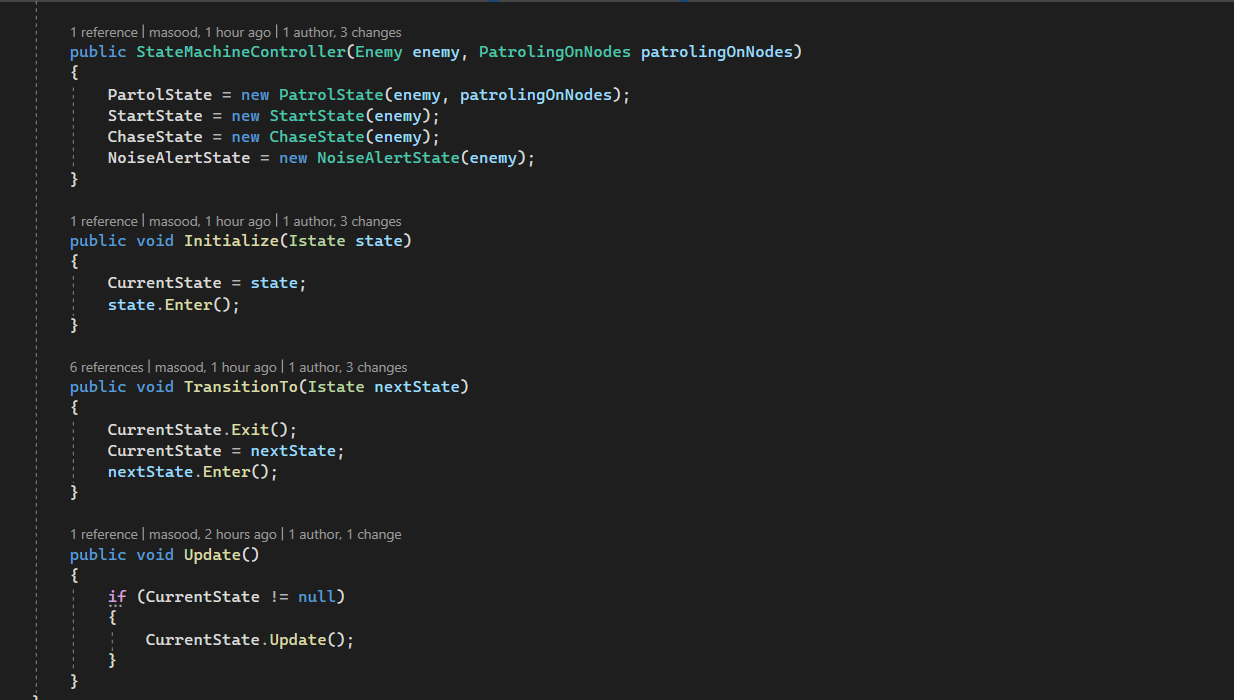


# Features Implemented

## FSM state Machine

Creating an FSM system to handle The AI behavior of Enemies.

It took 2-3 hours.



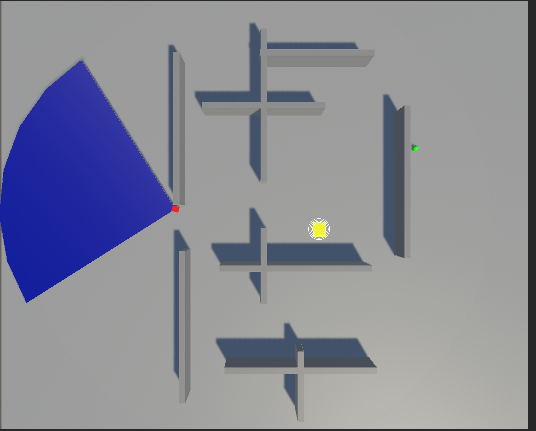
## base class of enemy

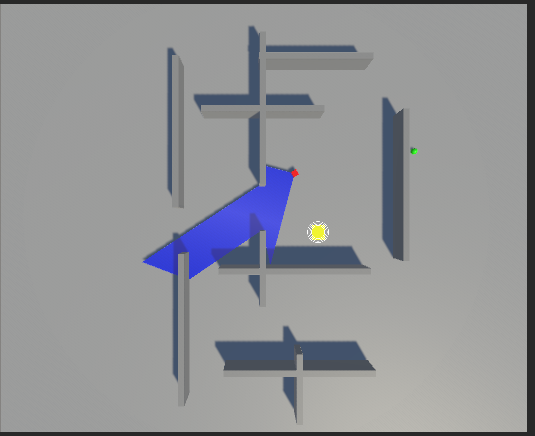
this is an abstract base class for all enemies. It has a handler for navmesh to use navmesh for moving and a player detection method to act as an eye of the enemy.

Any other enemy character should inherit from this class.

## field of view

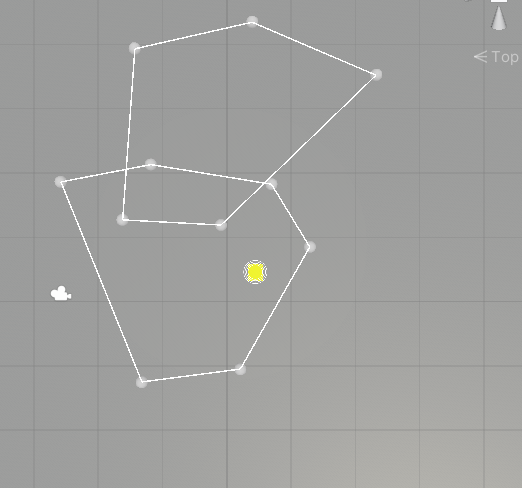
n this project, I implement a cone of player view to show how the enemy can see their world.I create a new mesh by trying to create a part of a circle mesh. And after that, I will change the vertexes to the hit point if there were a blocking object is near.It was really time-consuming. it needs 6 hours to implement.





## patrolling on nodes

as you can see, we can have different paths and each enemy can patrol on one of them. It can show the nodes and the connection between them.



## Guard enemy

A guard enemy is an enemy that can patrol on a path and also (because it inherited from the enemy) it can see around it and if the player is in its sight, it can change its state to chase him.

## Noise Alert device

There is a Yellow box on the level. If a player hit it, it will make a noise and if enemies were around it, they will notify and go to the sound generator



## Player run foot sound

The player can Go faster by using the left shift but will make a loud noise and if the enemy were around ,they will notify

## spawnpoint controller

this class will guarantee that enemies or players will not be spawned in the same position. Their random position will be unique till all the spawn points are used;

# Needs more Time

Since I had little time to implement and test this project I hadn’t enough time for working on other items that may be really useful to work on:

1. Make some test

Although I tried to make things as non-monobehaviour ( FSM classes) to make some tests on them I didn’t have time to add the test to this project

1. Separate player input handling to another class which could be replaceable by other types of getting input.
2. A better visual for the Field of view
3. Some more states like Idle or searching around for enemies to make their behavior more humanistic.
4. More realistic sound generation and detection.
5. A better level with better design of walls and obstacles.
6. Some sound, music, UI, and camera behavior
7. Some lighting on the level to make the sight of view more acceptable.
8. Add a reason for the player to hide and tried to get out of enemies sight. For example, collecting things or opening locks, or killing enemies