

Challenge 2

Steering & Movement

A document made to analyze the AI pathfinding and replicating it in Unity

Game Chosen:



Kingdom Hearts 1

Reason for choosing Kingdom Hearts

Kingdom Hearts is a game I've been playing for a very long time and have been replaying over and over. It has been such an inspiration, that I've been working on personal projects inspired about the gameplay and story telling. Additionally, the combat itself is very intriguing and engaging which makes or breaks the game. The enemies below have also intrigued me since I've been wanting to replicate it.

Video example of AI pathfinding:

https://drive.google.com/file/d/1QO-E5gXfVMFM9B_ma1QqWnkUbEeC7Ln9/view?usp=sharing

Initial thoughts:

First thoughts and I think this AI is both simple and incredibly clever. The main thing I notice is that the AI is able to be idle (almost like it's surveying the area with its animation), chasing the player, attacking the player, and moving around the place in a puddle. It can only attack/retreat

once it enters the chasing phase. From the gameplay, they seem to decide whether to attack after reaching their destination or hide in a puddle at random. Which is very interesting. Additionally, after attacking, they enter their idle state for a bit before engaging in the cycle.

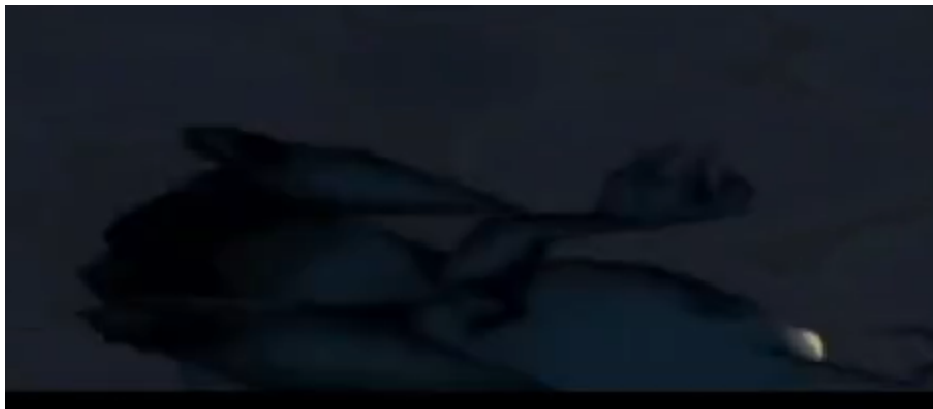
Steering Patterns:

Chase:



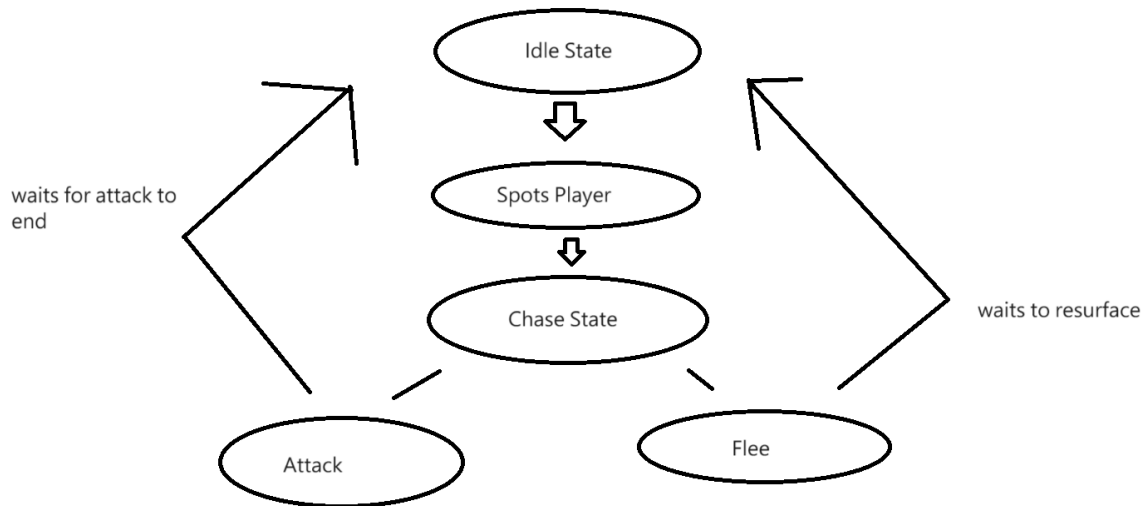
This state occurs when the enemy spots the player and walks around the player to get to them.

Flee:



This is a state where the enemy goes into the ground and wanders around, the AI isn't trying to go to the player but rather it is moving in random directions before leaving the state and going into the idle.

This diagram illustrated the states that I was able to spot when looking at the video:



It waits to be in the chase state before it chooses an action. Most enemies in the game do not follow the same AI for differentiating the enemies but most enemies will follow this same flowchart (of course subtracting the flee state for most of them).

While in the chase state, the AI adopts the A* pathfinding where the computer is able to process and output the best path for the AI to travel through which helps the enemies constantly follow the player and feel more oppressive. Once the enemy gets to the player it can transition into 2 different states:

- Attacking which makes the enemy jump or slash at the player before returning to the Idle State for a random amount of time.
- Flee by going into a puddle form and randomly goes in several directions, before resurfacing and returning to the Idle State for a random amount of time.

If the enemy were to go on a higher ground, they can still go to the player in the Chase State. There are a few frames in the video (0:22) where the enemy is on a higher ground but has no problem calculating the distance to get to the player so this is not an issue.

How well is it all integrated?

I believe the AI that is created and used as a basis is an excellent example of how to do an early game enemy. The enemy has 2 simple states it can go to when it is chasing and both tell the player 2 things:

- In the attack state, always keep your guard up and try to counterattack them if they leave themselves open.
- In the flee state, it tells the player that you can't keep attacking to win and instead of making the enemies counter you, they are briefly invincible while not hurting the player. This will better inform the player that it is best to have a strategy instead of mindlessly mashing.

The Finite State Machines presented by this enemy are clear and presented well to the player so they know how the enemies move and helps explain to the player how most enemies will move to attack the player. Its pathfinding AI also allows the enemy to reach the player as quickly as possible so that the enemies aren't too passive. Additionally, the delays between attacks so that the current enemy doesn't immediately start its loop so soon and keep attacking the player over and over.