Logan Ryan - AI Behaviour Evaluation

# Implementation of AI Elements

## Agent

The Agent implementation was first, and it was easy as this was already created from previous works. The class already contained the logic to set the behaviour, position and velocity of the agent. It also is a base class, so I could easily create derived classes from it.

## Player

The player class was easy to implement as again had been done from previous tutorials and most of the code was already in the agent class. The player had a simple keyboard controller behaviour.

## Enemy

Enemy was once again easy to implement because it was the same as the player class. Looking at it now I feel like that the player and enemy class were not needed as most of the code was identical and it lived in the Agent class.

## Pathfinding

The pathfinding algorithm itself was implemented easily as I had it done through previous tutorials. It is simply implemented in the Graph class and it will find a vector of points. A\* and Djikstras was implemented as well as the Manhattan Distance Heuristic.

I should have looked at implementing more heuristics as they would have helped with my pathfinding to a Chaos node or Master node.

Once the path is calculated the agent will then follow the path and when it gets close to the end of the path it will then clear the path and get a new path depending on what behaviour it is on. This was difficult as the agent need to know when they were at the end, so they can start to calculate the next path and know that they should not be going to the same node twice. The agent also needed to know if a Chaos Emerald was in the room, so it can calculate a path to the Chaos Emerald.

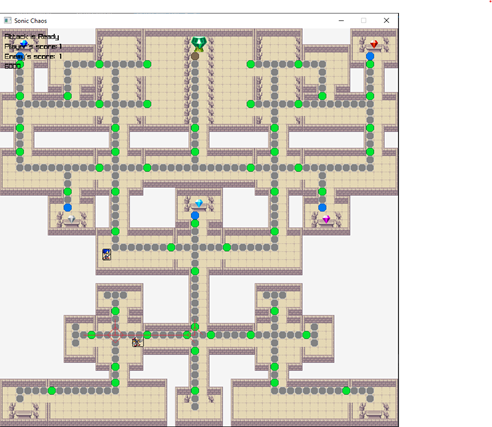


Figure 1 Door Pathfinding

It was tricky to set up pathfinding to the Chaos Emerald. The way I started doing it was by having the game check if the agent was a certain distance away from it but the distances weren’t consistent so I had to create a certain type of node for the agent to go to. Then this made pathfinding trickier as I had to get the agent to find a path to that specific node. In the end, I got it to work successfully and no errors came up.

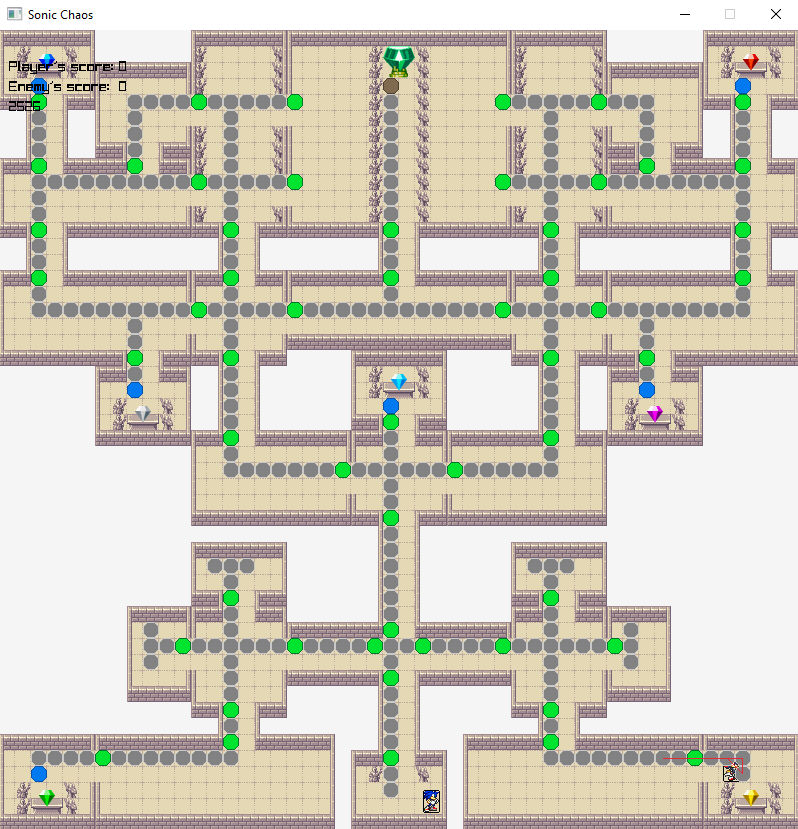


Figure 2 Chaos Node Pathfinding

## Collision Detection

Collision detection initially was hard as I was drawing rectangles using Raylib at specific spots at the start. This was very time consuming so to fix this my teacher showed me a trick I could do using the tile map creator “Tiled.” The trick was to create nodes on it with a specific colour, in my case I used black, and then used Raylib’s collision logic to stop the player and the enemy from going off the map.

# Performance

* The keyboard behaviour is working as expected.
* Pathfinding is also working well. The enemy agent is not going to the same node twice until it has been to all the other door nodes first. It is able to find the path to the Chaos Emerald and to the Master Emerald as well.
* Collision detection is also working well. The enemy and the player agent are not able to go through the walls.
* Switching Behaviours works ok but it only does it once it completes its path.

# Improvements

## Seek Health Packs

I was unable to implement health and health packs for the agents to use as I ran out of time to implement it. I am disappointed about this. This was to make it so if the agents health reduces to zero they would return to the start and all the Chaos emeralds they had would be transferred to the other agent.

## Field of Vision

Right now, the enemy agent uses distance to determine if the player is in view. I would like to change this so that it would only chase the player in front of it.

## Heuristics

I would invest more time, checking for different heuristics that could be used in A\* to optimize the pathfinding of the Agent. Currently only have 1, when I should have 3 because I have 3 different types of nodes.

## Behaviour Management

Right now, the agent changes its behaviour through if statements. I would like to change this so that the Agent can path find more efficiently. This could be done through a behaviour management object. The agent only changes behaviour once it completes its path so next time I do something like this I should spend more time implementing logic on how the agent will change behaviour.

## Another Agent

I would like to add another agent to the game to show that I have a good understanding on AI. Maybe I could have this agent behave differently to the one I already have.

# Post Mortem & Conclusion

Overall, I am happy with the result of the assignment. It worked just as I wanted it to. There is room for improvement. I am happy that I could achieve pathfinding as well as implement multiple behaviours for 1 agent. But at the same time, I feel like I could have done better.

I felt like I wasted too much time on creating the map that I did not make a lot of time for implementing the code for the AI. This has caused me to fall behind and not able to get started on coding the behaviours and not be able to seek the necessary help from the teacher. In future, I need to plan things better as there is so much to consider it is easy to visually see what needs to happen.

Maybe I am being too harsh on myself as I am still new to programming, but I feel if I am critical I will learn from my mistakes and improve in future projects.