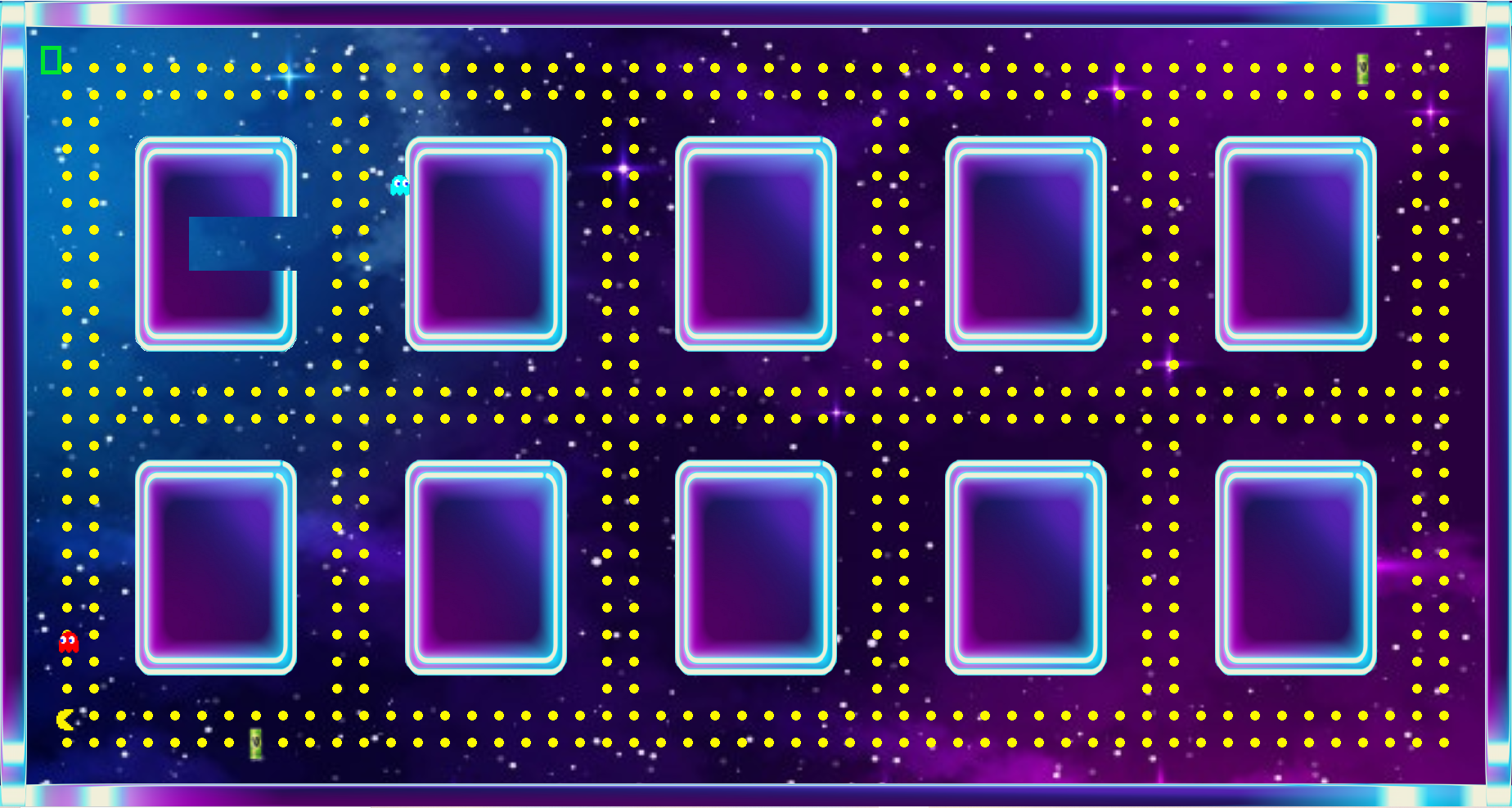
Project Evaluation Report

Artificial Intelligence for Games

Academy of Interactive Entertainment

Justin Katic, August 19, 2019



# Project Overview

The project I have created is a top down 2D version of Pacman where the aim of the project is for the player to eat as many of the yellow dots as possible while avoiding the ghost trying to hunt the player down.

During the pre-planned development, I was unsure if I was being too ambitious on my ideas or not for the timeframe allocated, however throughout production things started to piece together quickly. Early on in development I had plans and implementation ready for all the behaviours I desired for my game these included seek, flee, path-find and wander.

However, I came across some difficulties in the implementation stage of some of these behaviours. The first hurdle being wander. Originally wander behaviour was just seeking random positions on the map and seeking towards them but I decided this was not the look I was after for my game and so development of a new wander began. The new wander behaviour turned out to be much more difficult than anticipated due to a lot of math being involved in the algorithm. It entailed finding a random position around my player at the edge of the radius, but it also needed an offset, so it did not turn back on itself so suddenly.

The second big hurdle I was faced with was my flee behaviour. My original flee behaviour was just propelling my entities away from each other which led to an ugly look due to wall collision and was far from the flee behaviour I had hoped for in my result. The solution to this issue was to create a flee behaviour that used my pathfinding algorithm with the goal node being a safe place for the ghosts to escape. This was nearly perfect till I hit yet another big hurdle. The enemies would choose the quickest route to the safe area which sometimes led the enemies to go right towards the entity that it is meant to be running away from. My solution to this issue was a heat map, which at first appeared hard but on reflection, it was quite easy to understand what was happening. The map consisted of elements that matched my map position which contain the value of 0 and when my player was near any node in a 3x3 area those nodes would increase in value every second increasing their values up to 1. These values were then used to add weights to the nodes near my player encouraging the A.I to take an alternate route to get to its destination creating my flee like behaviour.

# Performance Analysis

I feel some of my algorithms could have been improved on especially my Astar and Dijkstra algorithms. The O notation of these functions could have been better in two easy ways. First, the amount of times the sort function is called could have been replaced with an insertion method, which involves inserting the node where it needs to go rather than sorting the whole list every loop. The second improvement could have been adding a second function with additional parameters since both algorithms are similar, but both were written twice with almost duplicate code.

Throughout my project I found the Raylib engine to be quite suitable for my needs. It gave me access to all the math functions I required the implement my behaviours. The Licensing of Raylib fit my project well due to its public access. I believe for the assessment criteria Raylib was fine however there are some better engine choices I could have used that would have made flow of development easier such as unity.

# Future Improvements

Overall, I felt the correct amount of time was given to create and implement my A.I as planned there were a few things I would have liked to improve. If I had more time, I would have liked to incorporate a game state manager to show a good pause screen or possibly menu screen. I would have also liked to improve the way my collision looks and feels with my wander behaviour, so it is not so jittery looking at times.

Given more time I would also have liked the opportunity to clean up some of my code since there is a lot of un-used code implemented from past behaviours that I did not implement in the final version. However, I feel like all the algorithms I implemented will be of great use in future projects and are written in a way that they can be easily imported to different projects in the future.