Project Evaluation Report

Artificial Intelligence for Games

Academy of Interactive Entertainment

Roy Batty, August 17 2019

# Project Overview

Use this section to give a high-level overview of your project and its development.

**Briefly (in one or two sentences) describe what your project is.**

The project is a simple simulator that allows the user to spawn in agents and other objects as well as a build a level. The user was able to modify agent’s attributes. Some of these attributes include movement speed and movement costs.

Then address the following questions in report form (i.e., write well-formed paragraphs that have a logical flow, taking note to avoid spelling or grammatical errors).

* **Did development adhere to your pre-planned timeline?**

Development for the most part did indeed adhere to the original pre plan. Minor changes were made such as the project transitioning from a maze solver app to a maze runner like app with enemies.

Originally, the simulator was supposed to take an image of a maze. The A\* algorithm will then solve that maze and show the steps to solve it.

Now there a different agent and powerups such as a speed boost, player agent, zombie agent and chaser agent. The player agent uses A\* to go to wherever the user left clicks.

* **What A.I. algorithms did you implement, or attempt to implement?**
* **What difficulties did you have in implementing these algorithms?  
  Possible difficulties worth mentioning might include:**
  + **Difficulty in understanding all details of the algorithm**
  + **Difficulty with programming / debugging**
  + **Performance issues, including memory management**
  + **Unexpected or incorrect agent behaviour**

A\* was the chosen algorithm to implement. Writing out the code was the easy part; the more challenging part was fixing up bugs. A\* was the chosen algorithm as it seemed at the time to be the most suitable.

The reason for this was because an agent would always have a start and end point. Therefore, there was no need to check a large quantity of nodes for no reason, just the neighbouring nodes from point A to point B.

Some bugs which needed to be fixed during implementation was the path node distance calculations and making sure bad nodes were skipped when getting neighbour costs.

Since a little bit of blackboard behaviour was implemented, it also meant the number of agents to blackboard had to be limited. Whilst testing many agents (around 300), FPS would have a major drop once every single agent was told to calculate a path to the player.

One solution which was used at the time was to just limit the number of agents that could be blackboarded at that given time. This meant that out of the 300 agents, only 100 would move to the player’s position.

# Performance Analysis

Use this section to analyse the performance of your algorithm(s) or techniques.

**Provide a brief description of the memory footprint of your agent class(es). Explain if this is efficient or could be improved upon.**

**Analyse you A.I algorithm and identify any performance bottlenecks or places for improvement. If possible, list the efficiency of your algorithms using Big O notation.**

Possible topics for inclusion in this section are:

* **Is it efficient for a lot of agents to use the same pathfinding algorithm?**
* **Should pathfinding be done every frame?**
* **How can you improve the performance of your pathfinding algorithm in the context of your game?**
* Flow Fields
* **Are all your algorithms efficient? Why/why not.**
* **How many agents could you have in your game before you start seeing performance issues, and have you tested this?**

# Future Improvements

**Did you get enough time to completely implement your A.I. as planned? What work did you not complete (and why)?**

**Can you see ways to improve your program/algorithms?**

**Were your algorithms good choices? Do you plan to use them in future projects (why/why not)?**

There is no word limit for this report, but it is expected that you provide enough detail for 1 to 2 full pages.