Computer Games Development CW208
SRS and Project Report Year III

Student Name: Martin Farrell
Student Number: C00157047

Contents

Acknowledgments

I would like to thank Ross Palmer, my supervisor for providing me with useful articles, documentation and direction.
I would like to thank Quintin Furlong for being my “rubber-duck” when I ran into issues with my implementation.

Project Abstract

Pathfinding over the last few decades has resulted in a lot of different approaches while many solutions are more applicable depending on the computer application my particular deep dive into it will be finding the most efficient of these to use in a game and by extension compare them to each other.

Project Introduction

I will investigate several steering approaches and compare them to context-sensitive steering. The solutions I have chosen are seek, wander and context-free.

Project Description

My project will contain a grid and various obstacles that will serve to obstruct the various pathfinding implementations. The pathfinding AI will be represented by triangles and each behaviour will be clearly labeled. There will also be highlighted goals in the level that the AI will try to find. However not all of the AI implementations will be aware of where the goal is on the level and will have to wonder about until they detect it, such as wander.

Background

There aren't really any games that come to mind but I've found practical applications of steering algorithms and complex systems such as in context-sensitive steering to be quite interesting as trying to do it any other way results in a lot of dirty code from even the cleanest "by-the-book" programmers as a lot of instances have to be hard-coded.

Overview

Philosophy:

Philosophical point #1

With this implementation of a context-sensitive steering my goal is to compare it with previous solutions to steering for AI and determine how it fares against it.

Philosophical point #2

The "game" will not have multiple menus or any kind of menu system/splash screen etc. this is because it is purely a research project in which I wish to find quantifiable comparisons between other solutions.

Philosophical point #3

It is important that each AI implementation works correctly and accurately as the results will be corrupted otherwise.

Philosophical point #4

Measurements carried out must be a quantifiable measurement and not a subjective measurement or opinion of how they operate. So the time taken, execution time, collisions and path length.

Define the Application

The application will be an accurate simulation and comparison of several possible AI solutions: “Context-free” steering, “Context-sensitive”, "wander" and "seek". Context-free steering does not have any knowledge of it’s environment and interacts with obstacles as it collides with them. Context-sensitive steering has numerous systems that calculate the danger and desirable directions and compute in several directions simultaneously, representing the value of each direction numerically.

What is the Application Supposed to do?

This application is supposed to measure how long each steering algorithm takes to traverse the obstacles towards the goal. Using time and other quantifiable measurements such as execution time, path length and number of collisions to determine which is the most efficient.

Who is going to use this application?

I will be the only person using the application.

Context Diagram and Use Cases

Metrics: If I successfully implement context-free steering, context-sensitive steering, seek, wander and attain quantifiable information that will determine which is the most efficient then I will have been successful in my endeavour.

Metrics:

The main metrics I will use to measure my results are the time efficiency, execution time, number of collisions and path length. The time efficiency will quite simply be a measurement of the time taken for a given AI to complete the simulation i.e. find the goal, measured in seconds and accurate to several decimal points. The execution time will be calculated by finding the length of time it takes for the AI to compute a steering/movement decision and divide that number by the current number of frames. The number of collisions will simply be an account of the number of times and AI has bumped into an objects and the finally the path length will be measured in pixels using the distance formula and a previous position vector which will store the last postion of the AI when the calculation was performed and calculate the distance between it and the AI's current positon.

These metrics will be added together to create a score using the scoring method provided on the "Steerbench" paper however instead of using their formula which is "Si = wc,i(αC) + wt,i(βT) + we,i(γE) =C' + T' + E' " I will adopt a simplified version of my own which also incorporates a 4th variable in path length so S = T + E + C + P. The most "efficient" AI will have the lowest score using this formula and the least efficient will have the highest.

Is there a precedent for this application? (Your inspiration)

There has been a lot of research done into pathfinding and the effects it can have on gaming and programming. AI is commonly what the players will complain about when it comes to games and efficient solutions to steering in dynamic environments are quite convoluted and hard to read or even explain to other programmers. While there is somehwhat agreed upon bench mark formulas for scoring AI the weightings and methods of normalisation are abitrary such as the one mentioned on the paper "SteerBench: a benchmark suite for evaluating steering behaviors" by Shawn Singh*, Mubbasir Kapadia, Petros Faloutsos and Glenn Reinman.

Project Milestones

16-10-19: Started Project.
23-10-19: Added base classes and created game loop, constructors and render functionality.
30-10-19: Added grid based maze and obstacles.
11-11-19: Implemented context-free steering and collision detection.
09-12-19: Added Context-sensitive prototype and drafted SRS and TDD.
12-02-20: Reformatted simulation and changed AI functionality.
18-03-20: Lockdown in effect, had some issues with new machine had to change the dimensions of everything. Additional work on context-sensitive steering. Added a vertex array representing 8 directions(north, north-east etc).
25-03-20: Created the functionionality that would allow raycasting to objects as well as calculating distances and getting a unit vector.
22-04-20: Implemented seek and wander AIs and added labels to each of the AIs.
29-04-20: reformatted the entire simulation so instead of the AI all running around at once they take turns, also calculated scores and displayed them between each simulation and compared all at the very end.

Project Review and Conclusions

While the project was a learning success in many aspects I did not achieve what I set out to do. My implementation of context-sensitive steering was flawed and there were several other aspects that corrupted my results to some degree such as using a counter instead of delta time for execution time as the number of updates per second will vary with machines and this is not a very accurate measurement.

Were I to do the project again I would make the entire environment dynamic to further accentuate the difference between context-sensitive steering and the other solutions. I would also create more professional implementations of the other AI as some like the context-free AI I implemented tend to favour one direction should they collide.

References:

Pathfinding, wikipedia accessed at: https://en.wikipedia.org/wiki/Pathfinding (accessed 21st Oct 2019).

Pathfinding simulator accessed at: https://qiao.github.io/PathFinding.js/visual/ (accessed on Nov 6th 2019).

A* comparison at theory stanford accessed at: http://theory.stanford.edu/~amitp/GameProgramming/AStarComparison.html (accessed on 9th Nov 2019).

Craig Reynolds GDC paper 2019 accessed at: https://www.red3d.com/cwr/steer/gdc99/ (accessed on 9th Nov 2019).

Understanding steering behaviours at gamedevelopment.tutplus.com accessed at: https://gamedevelopment.tutsplus.com/tutorials/understanding-steering-behaviors-pursuit-and-evade--gamedev-2946 (accessed on 9th Nov 2019)

Context sensitive steering Andrew fray Accessed at: https://andrewfray.wordpress.com/2013/03/26/context-behaviours-know-how-to-share/(accessed on 26th Oct 2019)

SteerBench: a benchmark suite for evaluating steering behaviours, accessed at: http://www.cs.uu.nl/docs/vakken/mcrws/papers_new/Singh%20et%20al%20-%202009%20-%20SteerBench.pdf_ (accessed on April 29th 2020)