Artificial Intelligence for Games

Assignment 1

In this assignment I have been tasked with creating an interactive application that demonstrated the use of different AI components such as finite state machines, various steering behaviours and path finding. I have developed an application that simulates basic traffic flow to try to exhibit these methods. For this task I have used SFML or Simple and Fast Multimedia Library (Milchev, 2005), as it speeds up development due to the easy to use application programming interface (Pupius, 2015).

Finite State Machines

A state machine is simply a model of behaviour that uses decisions, based on input, to generate signals representing actions (Wagner, 2006). A finite state machine (FSM) can only be in one state at a time that makes transitions from one state to another (Buckland, 2005). Each of the cars in the application uses an FSM that performs an action. The diagram below demonstrates how the states transition.

The car starts in an Idle or Spawn state. In this state the car simultaneously decides a random spawn location as well as a random destination location. The car transitions from a Driving state to a Waiting state when it has reached a junction. The car then performs a calculation to determine the next path, and repeats this process until the car has reached the destination. The states are designed to be easily distinguishable so the user could understand exactly what the car is doing and what to expect from the car at each state. The states are a collection of tidy functions that easily relate to the name of the state, for example, the Driving state simply transitions the car from junction to junction and the Waiting state produces the next direction (Ndjountche, 2016).

Path Finding

There are many forms of pathfinding used in game AI such as Dijkstra’s algorithm, which checks every vertex around itself to find the shortest distance to a point (Pearson and Bryant, 2004). A\* algorithm is similar to Dijkstra’s in that it will always find the shortest distance, but instead of checking every point around itself, it only checks the points progressively towards the destination to shorten the calculation time (Millington and Funge, 2009). I decided that neither of these methods for path finding would be suitable for the application I created as they would have been too excessive, these methods would not have been necessary as the cars travel down a linear path as opposed to detecting collision. Instead I programmed the car to complete a vector calculation at each junction to determine the shortest possible distance to the next junction. This is because none of the conventional path finding algorithms were not as efficient. Each junction checks the cells surrounding it, based on north, south, east and west coordinates, and then decides which cell the car should travel to.

Steering Behaviours

Steering behaviours are used in games AI to create the illusion of intelligent and natural movement (Kyaw and Swe, 2013). There are many types of steering behaviours that I could have employed for the birds for example seeking, flocking or wandering. In this application I have created birds that can be spawned by the user that will inherit a seeking steering behaviour. The seeking behaviour moves towards a target location using a steering force that turns towards the target. I decided to use seek as this behaviour is fundamental when developing more sophisticated behaviours, such as flocking (Champandard, 2004). I did not end up developing the flocking behaviour due to the birds being secondary to the traffic simulation. If the birds were too overbearing the application would become too cluttered too quickly. The application has the option to gain information about each of the cars. The statistic menu on the right side of the screen shows the current state of the car, the current coordinates of the car and finally the destination coordinates.

To conclude I have created a traffic simulation game that utilises finite state machines to calculate each subsequent path for the cars. I have created birds that fly over the scene using a seeking steering behaviour. I have also programmed the ability to obtain information about each car upon button press to assemble a fully interactive AI application.

Bibliography

Buckland, M. (2005). *Programming game AI by example*. 1st ed. Plano, Tex. Wordware Pub.

Champandard, A. (2004). *AI game development*. 1st ed. Indianapolis, Ind.: New Riders.

Kyaw, A. and Swe, T. (2013). *Unity 4.x Game AI Programming*. 1st ed. Birmingham: Packt Publishing.

Milchev, Milcho G. *SFML Essentials*. 1st ed. Packt Publishing Ltd, 2005. Print.

Millington, I. and Funge, J. (2009). *Artificial intelligence for games*. 1st ed. Burlington, MA: Morgan Kaufmann/Elsevier.

Ndjountche, T. (2016). *Digital electronics*. 1st ed. John Wiley & Sons, pp.10-330.

Pearson, D. and Bryant, V. (2004). *Decision maths 1*. 1st ed. London: Heinemann.

Pupius, Raimondas. *SFML Game Development By Example*. 1st ed. Packt Publishing Ltd, 2015. Print.

Wagner, F. (2006). *Modeling software with finite state machines*. 1st ed. Boca Raton, FL: Auerbach.