Andrew Rimpici

Jeff Timanus

EGP-410-101

09/24/2018

Assignment One Overview

Overall Architecture

Each steering algorithm, *SeekSteering* (already provided in the assignment), *ArriveSteering*, *FaceSteering*, *WanderSteering*, *WanderAndChaseSteering,* and *ArriveAndFaceSteering,* has its own class for each separate implementation. All of these steering objects are subclasses from the main *Steering* class. The book, *Artificial Intelligence for Games* by Ian Millington, goes into detail about *AlignSteering* as well, which has also been included, along with the other steering algorithms, to aid in the implementation of *FaceSteering*. The steering algorithms *Face, WanderAndChase* and *ArriveAndFace*, hold multiple steering algorithms and combine the results from them to get more complex output. For example, *FaceSteering* holds an *AlignSteering* object, *WanderAndChaseSteering* holds a *WanderSteering*, *SeekSteering*, and *FaceSteering* object, and *ArriveAndFaceSteering* holds an *ArriveSteering* and *FaceSteering* object.

In addition, there is now an *InputSystem* class that is responsible for all input events fired from SDL. These events are converted to Message objects and are processed accordingly using polymorphism. Each message type: *KeyboardMessage, MouseMessage, and RequestWindowCloseMessage,* has its own class and handles events in the *process()* function.

Challenges Faced In Development

The most challenging part of this assignment was implementing *WanderSteering* as I was having issues with it behaving in the correct way. I also had some trouble mapping the book's variable names to the existing project's variable names. On a positive note, I expected WanderAndChase and ArriveAndFace to be harder to implement than they were, but they ended up being the easiest since they are made up from the other steering algorithms.

Areas Of Improvement

FaceSteering seems to turn very slowly, which causes issues with the *WanderAndChaseSteering* algorithm as the enemies do not turn fast enough when they are seeking their target. Also, I think it may be a random number generation issue, but all of my enemy units take the same wander path.