Assignment 6

CS 381: The Game Development Pipeline
Spring 2014
Max Score: 100

Simple AI for Games

You will work on implementing simple AI for your game engine. In this assignment you will implement move, follow, and predictive intercept for entities in your game engine. To do this, you will need to implement a command class, a unit AI aspect that manages sequences of commands, and the move, follow, and intercept commands.

I encourage you to use the solution to the prior assignment as your starting point. I will demonstrate a working example of this assignment in class.

0.1 AI Commands (50 points)

Design and implement a Command class in python. Think of this as an abstract class which ensures that subclasses that inherit from Command will implement (override) the tick method with their own implementation.

- **tick(dt)**, a method that is called every dt seconds to update the command. The Command class’ tick does nothing.

  Move, Follow, Intercept are the three subclasses of Command. Each of these will have a different tick implementation.

0.2 Command Invocation (30 points)

We will follow RTS game conventions and use left clicking the mouse for selecting entities. Right clicking the mouse specifies the command and command’s target.

- Right clicking over a target entity, $T$, while holding the left control key means intercept $T$
  
- Right clicking over a target entity, $T$, means follow $T$ at a distance of 100 units
  
- Right clicking over empty water means to move to the right-clicked location
The command applies to all selected entities. Thus when you left click to select an entity $E_s$ and then right click on an empty location, $P$, in the world, $E_s$ will move to $P$ and stop. More specifically, for Move:

1. You will detect the right click, realize that it is over the empty ocean an not over an entity, and thus create an instance of Move.
   
   ```python
   newCommand = Move(E_s, P)
   ```

2. You will tell $E_s$’s unit AI aspect to begin executing `newCommand`.

   Intercept and Follow will work the same way, but instead of taking a target position, $P$, as the second argument, they take a target entity as their second argument.

**Command sequences (10 points)**

You can also sequence commands using the shift key. Shift-move (holding down the shift key while right clicking the mouse over empty water) means add the corresponding move command to the list of commands managed by the Unit AI. Shift-intercept means add an intercept command. Shift-follow means add a follow command. Note that it is possible of a command (like follow) to never terminate, meaning that subsequent commands in the list for that entity will never be executed.

**0.3 Unit AI (10 points)**

This brings us to the last component that you will need to design and implement. The UnitAI aspect holds the currently executing command. Initially, the UnitAI aspect has no command and does nothing during its tick. Once a command is assigned to the UnitAI, as in item 2 above, it calls the new command’s tick. Every time you give a new command to this aspect’s entity, the currently executing command is replaced by the new(er) command.

**0.4 Quitting**

Hitting the escape key should shut down your running game engine. No other key should cause your engine to shut down.

**0.5 Design contraints**

Use my solution assignment five as your starting point. If you prefer your solution to mine and it works well, go ahead and use your assignment five solution but expect less debugging help from us.

Note that if more than one entity is selected (for example, when using the tab key), you will be assigning your command to all selected entities.
0.6  **Cumulative Extra Credit**

- Add group mouse selection (+10)
- Add wakes to all entities. This code should be added to Renderer (+20)

**Turning in your assignment**

Assume that this format will be used for all your laboratory assignments throughout the semester unless otherwise specified.

1. Demonstrate your working program in the lab on the due date.

2. In lab, turn in hardcopy to me with

   (a) Your FULL name, assignment number, and email address
   (b) Source code listing
   (c) Screenshots (if any)

Ask me (sushil@cse.unr.edu) if you have questions.

**Objectives**

1. Demonstrate an ability to apply knowledge of computing, mathematics, science, and engineering by learning and applying knowledge of Python to solve a problem (1)

2. Demonstrate an ability to analyze a problem, and identify, formulate and use the appropriate computing and engineering requirements for obtaining its solution (5)

3. Demonstrate an ability to use the techniques, skills, and modern engineering tools necessary for engineering practice (11)

4. Demonstrate an ability to apply design and development principles in the construction of software systems or computer systems of varying complexity (13)