Josh Moore

1/22/2013

A1 Write-up - Ms. Pacman Ghost AI

Brian Magerko

Instructions

-------------------

1. Make a new project in Eclipse.

2. Import the compiled zip file into the project, making sure that the "data" and "images" folders are outside of the main source folder. (Eclipse has a problem compiling and running the project if this is not the case)

3. Run Exec.java using Eclipse. The player is set to HUMAN controlled, so please use the arrow keys to move Pacman. You can also press "1" to pause the game (it doesn't pause the threads though)

Assumptions

--------------------

There were a few assumptions that we had to make based on the disparity between the Pac-man Dossier and the Game Internals doc with the game system we were using ourselves. One was that we would always have the positions of the ghosts and Pacman at all times, and that reversals done in the ghost's AI. We also assumed that we had a tile grid system in order to calculate distances, which in reality, our system only had nodes (similar but not as robust).

Major Challenges

-------------------------

The first thing that I had trouble with was deciding on a structure I was happy with. I wanted to implement good practices into the system by isolating the AI from the State Change code. This is more for myself than for the sake of the project. My final decision was to separate the Chase and Scatter code into their own separate the targeting code out of the main ghost AI. I thought that if I ever wanted to change how the ghosts behaved, it would be on the targeting code, and thus I could easily switch up the classes in order to try different AI setups.

As with most everyone in the class, I had trouble with the targeting system itself. The base system was with Nodes. The problem with Nodes is that they were not easily mapped to the space on the screen. Each row of Nodes had a different number of Nodes in them, thus making it difficult to target specific nodes based on visual spacing. As such, I did not rely on the nodes to target, but the X and Y coordinates on the screen. I wrote an additional Euclidean Distance function for an X,Y scale, and re-wrote pathing logic in order to emulate what was said the documents (shortest distance is priority). However, this actually does make a difference in terms of accuracy. With nodes, the shortest distance was sometimes equal to another direction, causing a prioritized direction choice. With the XY system, this equality does not happen often due to the scale of the Euclidean Distance. A buffer could be set, some epsilon range, to provide a closer simulation, but I did not implement this.

I also had some problems with working with the game engine in general. First off I had to change some of the elements of the system in order to make it similar. The list of these include:

* Changing the lair exit directions to all left (done in Game.java) to match the documents.
* Add a pause function that suspends the graphics update. (However it doesn't suspend the ghost AI thread)
* Changed the pac-man control over to Human AI in order to test.

I also had to deal with the fact that there was no reverse function that could be properly used in the AI to time the reversals of the ghosts as they switched states. I attempted to disable the reversal functionality in the updateGhost function in Game.java, however when I tried to emulate it in the Ghost AI it did not propagate in game. I am not entirely sure on why this occurs, maybe something to do with the ways that the thread process the AI. Thus one of the obvious flaws with my system is that the reversals do not occur on state changes.

I also ran into a small state change problem, where my ghosts would become frightened and then never go back to their previous states after the timer ran out. It turned out that I was using the timer wrong and because of this didn't move from the frightened state properly. This was fixed though and worked properly.

Questions

-----------------------

1. I think that it was a good design of the ghosts AI. Mostly for the reason of the variety, yet simplicity of the AI. If all of the ghosts were similar to Blinky (Red), the challenge of the game would be next to nothing, as you could gather them all up behind you and practically win the level. That wouldn't be fun it all. Then you add "personality" and base the character around it. With some very simplistic choices (as in Clyde's AI), the ghosts remain predictable, yet as a GROUP they become a challenge to deal with. The only one that is sort of difficult to decipher is Inky (Cyan), as his target is very strange. It was also well designed for the CPU power at the time, because all of the targeting and pathing choices were based off of simple vector based choices. FSMs are very efficient systems for low processing power systems.
2. I would NOT re-design the ghosts. I feel that the AI gives just enough challenge to the player that it makes the game repeatable and fun. With modern computers, we could use a stronger AI methods to make better decisions, such as ambush tactics, or trapping movements based on level design. However, the player is still limited to the space that game gives them. If the ghosts were any smarter, the player is most assuredly to lose within the first few levels, which would not be fun for the player.