

Game AI, Spring 2013

Project 1

Due: 1/22/12 at 11:55pm



Knowing the best place for an agent to move at any given time is a key element to almost any game AI. First person shooters, sports games, real-time strategy games, turn-based strategy games, etc. all share this common problem of needing to evaluate the current situation and decide where in space to move next and what action to take. This individual assignment focuses on movement in a semi-continuous 2D environment: Ms. Pac-Man.

This project will be using the IEEE Ms. Pac-Man vs. Ghost Team infrastructure for implementation. The description and distribution of the software can be found here: <http://www.pacman-vs-ghosts.net/>

This is an individual student project. You are to implement the 4 ghosts in a way that replicates the behavior of the original game.
See <http://gameinternals.com/post/2072558330/understanding-pac-man-ghost-behavior> for a description of the original ghost behavior AI. (Note that when you read about the ghosts you will see that there was an error in the code resulting in ghosts that occasionally path planned to the wrong location. You may choose to replicate the bug as a feature, or correct the bug in your own implementation. In your write up, please indicate whether you are implementing the bug/feature or not.)

A single controller object that inherits from GhostController controls all ghosts. Modify the game/controllers/examples/MyGhosts.java so that it implements the behaviors of the original ghosts. In the version of MyGhosts provided, all ghosts are programmed to want to travel to Ms. Pac Man's current location. The MyGhosts constructor takes a Boolean that, when true, should draw a line of the appropriate color (Red, Pink, Blue, or Orange) from the ghost to the cell on the game board that it wishes to travel to. This is for grading purposes; you will receive an "F" if you cannot draw the appropriately colored lines when the Boolean is true.

One of the skills involved in this class is learning how to deal with integration issues with pre-existing engines. You are responsible for obtaining the code and using the resources provided with the competition. Please use Piazza for all online conversations and questions for the class community.

You will be given 2 weeks for this assignment with weekly-suggested milestones:

W1: Download and run game. Experiment with different example ghosts and understand how they work. Implement at least the Red ghost.

W2: Implement the remaining 3 ghosts.

Get up and running

1. Unzip [mspacman v2.1.2 cs4731.zip](https://research.cc.gatech.edu/inc/sites/edu.inc/files/cs4731/mspacman_v2.1.2_cs4731.zip)
(https://research.cc.gatech.edu/inc/sites/edu.inc/files/cs4731/mspacman_v2.1.2_cs4731.zip)
2. Build the game engine: `javac -cp . game/Exec.java`
3. Run the game engine: `java -cp . game/Exec`
4. Modify `game/controllers/examples/MyGhosts.java`

Hints

- Look at the examples in `game/controllers/examples`, `AttractRepelGhosts.java` is especially helpful.
- See <http://www.pacman-vs-ghosts.net/software> for the game engine API.
- All ghosts are implemented by the same object by filling out a `directions[i]` array such that each element is the next direction ghost 0..3 should go.
- Modify `main()` in `Exec.java` to use your Ghosts controller class. You may also want to turn on Human control of PacMan: `exec.runGameTimed(new Human(), new MyGhosts(), true);`
- Use `GameView.addLines(game, Color.RED, game.getCurGhostLoc(i), myComputedGhostDestination);` to visualize where your ghost is going.
- Path planning is precomputed, use `getNextGhostDir()` with the appropriate distance metric. You may find it convenient to map between the node indexing scheme built into the engine and an X,Y coordinate system. There are functions to convert from node indices to X,Y coordinates. But there is no convenient way to convert back to node indices. You may want to implement your own technique for doing so.

Writeup

Include a .doc or .pdf file that describes the following:

- Exact instructions for how to run your agent (if these are not included, you may receive an F).
- Any assumptions you made when interpreting the descriptions of ghost behaviors.
- The major challenges you met during the project and how you overcame them.
- Answer the following questions:
 - Explanation why the design of the ghost AI was effective, or why it was not. In your answer, give consideration to the tradeoff between fun vs. challenge and the constraints of computer processing power at the time.
 - Given the processing power of modern computers, would you re-design the ghosts? Why or why not?

Submission

Submit via T-Square everything needed to run your agent, including the native pacman code, and your writeup.

Grading

This project is worth 10% of your final grade.

- Ghost implementation: 4 points (1 point for each ghost)
- Writeup: 6 points
 - Instructions to run your code: 0 point. But you may receive an F if you do not specify exactly how to run your code.
 - The assumptions you made when interpreting the description of the ghost behaviors: 1 point.
 - The major challenges you met and how you overcame them: 1 points.
 - The two questions: 2 points each.