Ms Pac Man Project

Michael Mateas   Peter Mawhorter

Department of Computer Science
University of California Santa Cruz

January 12, 2012

Project 1

Write three controllers (one for Ms Pac Man and two for the ghosts) based on techniques from the class. One of the ghost controllers should be designed to make the game fun to play, the other two controllers should try to maximize score (or minimize it for the ghost controller).

The Competition

- Code from the Ms Pac Man vs. Ghosts Team Competition, which started last year.
- Rules and code are available at `http://www.pacman-vs-ghosts.net/`
- Projects should follow all of the competition rules.

Controllers

- For the Ms Pac Man controller, try to maximize score.
- First ghost controller is trying to minimize score.
- Second ghost controller should be fun to play against.

The Code

- The `game` folder contains the code.
- `Exec.java` can be edited to run different tests.
- The `game/entries` folder is where you'll write code.

The Code

- `game/core/Game.java` is an interface for controllers.
- `game/core/G.java` implements that interface (if you need to see stuff under the hood.
- `game/controllers` has the controller interfaces.

Making a Controller

1. Write a class that implements either the `PacManController` or `GhostController` interfaces.
2. Use methods from `Game.java` to read the world state.
3. Return either one or four direction values for the current frame.

Running a Controller

1. Edit `Exec.java` to use your controller (remember to import it).

2. Use the `runExperiment` method to run lots of games quickly.

3. `Exec.java` has plenty of commented examples for different run modes.

The Game.java API

- Functions for getting information (`getLevelTime`, `checkPill`, etc.).
- Functions for computing information (`getNextPacManDir`, `getPathDistance`, etc.).
- Reference/walkthrough is at `http://www.pacman-vs-ghosts.net/software`.

Data Structure

- Everything is an integer.
- Directions are up - 0, right - 1, down - 2, left - 3.
- The maze is a graph (not all nodes are connected).
- Each node in the graph is represented by an integer.

Data Structure

- The pills and power pills are stored in arrays indexed by integers.
- The ghosts have integer indices.
- Just use the `Game.java` API and pretend you're passing objects around.

Important Functions: Pac Man

- `getCurPacManLoc` - Location
- `getCurPacManDir` - Direction
- `getPacManNeighbors` - Neighboring cells
- `getPossiblePacManDirs` - Movement options
- `getNextPacManDir` - Pathfinding to a location

Important Functions: Ghosts

- `getCurGhostLoc` - Location
- `getCurGhostDir` - Direction
- `getGhostNeighbors` - Neighboring cells
- `getPossibleGhostDirs` - Movement options
- `getNextGhostDir` - Pathfinding to a location
- `getEdibleTime` - How long until dangerous again
- `isEdible` - Can be eaten by Pac Man

Important Functions: Pills

- `getPillIndex` - Find pill from cell
- `getPowerPillIndex` - Find power pill from cell
- `checkPill` - Has it been eaten?
- `checkPowerPill` -Has it been eaten?
- `getNumberPills` - How many left?
- `getNumberPowerPills` - How many left?