A Dual-Mode Software Agent for Ms. Pac-Man

Kuan-Wei Chen, Chu-Ming Wang, and Tsung-Che Chiang

Department of Computer Science and Information Engineering,
National Taiwan Normal University, Taiwan, R.O.C.

sayuki_goku@hotmail.com, muming25@gmail.com, tcchiang@ieee.org

Overview

This paper presents a dual-mode software agent for the Ms. Pac-Man competition held in 2011 IEEE Conference on Computational Intelligence and Games (CIG 2011).

How to run the program

1. Run the game: Open any browser and start the Ms. Pac Man game by visiting http://www.webpacman.com/mspacman.htm.

2. Get the required information for screen capture:
   (1) Capture the screen by pressing “Print Screen” on the keyboard.
   (2) Open the Paint program in Windows.
   (3) Paste the captured screen by pressing “Ctrl” and “V.”
   (4) Move the cursor to the top-left pixel of the white frame and find the coordinate of that pixel. Fill the coordinate in the file location.txt (Figure 1).

3. Run the agent, the executable file MsPacMan.jar.

4. Start the game. Please wait until the game stops ten times.

Figure 1: How to get the location of game’s screen.
Proposed Agent

a. Dual-mode agent

Our agent has two modes. When the program starts to run, the agent runs in Normal mode. When there are two or more ghosts nearby and the agent runs in Normal mode, it switches to Cautious mode. When there is no ghost nearby and the agent runs in Cautious mode, it changes back to Normal mode. In other cases, the mode is not changed. Here the ghost refers to the normal-state (inedible) ghost. The state diagram is in Figure 3.

![State Diagram](image)

Figure 3: The state diagram of agent’s mode.

The Normal mode follows the rules in our previous version of agent, which participated in the competition in CIG 2009. Ms. Pac-Man uses the depth-first search (DFS) to search for the path to the pill, power pill, and ghost with a limit of search depth and calculates the distance between Ms. Pac-Man and the first encountered objects. Then, she selects a direction by the following rule:

1. If there is no safe direction, i.e. moving along every direction will encounter a normal-state ghost, choose the direction that will cause the farthest distance from the ghost.
(2) If there is at least one safe direction, choose the direction in the priority order: the direction that leads to an edible ghost, the direction that leads to a power pill, and the direction that leads to a pill. If there is more than one direction that leads to the same type of object, choose the direction that causes the shortest distance.

Some improvements are made in this version:

(1) We do not scan the entire game screen to obtain the current state of the game. Instead, we scan only the area around Ms. Pac-Man and ghosts, as illustrated in Figure 4(a). It helps to save computation time.

(2) When edible ghosts are close and form a big blue chunk, we estimate the number of edible ghosts by their volume. It helps to direct Ms. Pac-Man to move to capture more edible ghosts and more scores.

(3) When Ms. Pac-Man is chased by ghosts, she is disallowed to enter pre-defined dangerous areas, as shown in Figure 4(b).

(4) The button of last direction is not released until a new direction is decided.

(5) We extend the map to store the invisible part of the tunnel when Ms. Pac-Man and ghosts temporarily disappear in the tunnel.

(6) We add the map information of stage 2. We monitor the words “HIGH SCORE” (see Figure 5) on the top of screen. When it disappears, we change the map information to that of stage 2. It helps Ms. Pac-Man to play further in case she reaches stage 2.

Figure 4: (a) Searching areas (b) The dangerous area
The *Cautious* mode is newly introduced in our agent. In the *Normal* mode, Ms. Pac-Man searches along each direction until she encounters the first object, whereas in the *Cautious* mode, she searches along each direction with a fixed depth. We assign each kind of object a score, and the scores of all objects along the search path are accumulated. Ms. Pac-Man will then choose the path with the highest total score. During the search, when a main path branches into \( n \) sub-paths, the score on each sub-path is divided by \( n \) and the scores of all sub-paths are summed to the main path. This calculation is done recursively. There are six parameters in the *Cautious* mode:

1. the maximum search depth \( (D_1) \),
2. the score of power pill \( (S_1) \),
3. the score of encountering a normal-state ghost from its front \( (S_2) \),
4. the score of encountering a normal-state ghost from its back \( (S_3) \),
5. the score of encountering an edible ghost from its front \( (S_4) \), and
6. the score of encountering an edible ghost from its back \( (S_5) \).

b. GA-based parameter tuning

We use a Genetic Algorithm (GA) to find a suitable setting for these parameters. The integer coding is adopted, i.e. the chromosome consists of six integral variables. The flow of our GA is shown in Figure 6.

The details of each step are given in the following.

1. Initial Population: we generate six random individuals.
2. Evaluation: Our agent plays the Ms. Pac-Man game with the parameter values encoded on the individual for ten times. The third highest score among the ten runs is taken as the fitness of the individual.
3. Mating selection: We select two pairs of individuals as parents randomly.
4. Reproduction: The parents are temporarily transformed to the binary format and generate the offspring by the typical 2-point crossover. Four offspring are produced by crossover. One additional offspring is produced randomly, as the way we use in the initialization step. These five offspring are evaluated and assigned the fitness.
5. Environmental selection: The best six individuals with the highest fitness survive to the next generation.
Due to the time limit and submission deadline, we did not run the GA for many generations. We will keep the GA running and searching for better settings.