CSE 2010, HW6, Fall 2019

Due Mon Nov 19 at the start of your lab section; Submit Server: class = cse2010, assignment = hw6S*x*Individual

Due Mon Nov 19 at the end of your lab section; Submit Server: class = cse2010, assignment = hw6S*x*GroupHelp

x is 2 or 3—your section number (or j for java).

In the game of Pac-man, the player controls the Pac-man in a maze to earn points by consuming dots. To foil Pac-man's apetite, ghosts roam around trying to consume Pac-man.

HW6 explores graph algorithms to simulate a simpler version of the Pac-man game. Given a starting position in a 2D grid world, a player's goal is to move Pac-man to consume the most dots (each dot earns one point) without being consumed by one of the ghosts. Naturally, once consumed, a dot is no longer available. At each step, the player can move Pac-man up, down, left, or right to an adjacent empty cell or a cell with a dot. Similarly, at each step, the ghost can move in those four directions to an adjacent cell that is empty, with a dot, or with Pac-man. Ghosts do not consume dots and ignore them. For simplicity, the ghosts move at the same speed as Pac-man.

The player moves Pac-man first, then each ghost (in alphabetical order) will move. Trying to reach Pac-man quickly, each ghost decides which direction to move based on the shortest path from its cell to Pac-man's cell. The distance from one cell to an adjacent cell is 1. For easier debugging and testing, during Breadth-First Search for the path, consider the valid adjacent cells in this order: up, down, left, and right. Each cell can have Pac-man, a ghost, a dot, or an obstacle, or can be empty. *Note that since ghosts do not consume dots, a cell could have both a ghost and a dot, but only the ghost is shown.* The game stops after no dots remain, or one of the ghosts reaches Pac-man (cell).

HW6: one round of the first move from Pac-man and the first move from the ghosts.

HW6 Extra Credit 1 via hw6_extra1.c (or HW6Extra1.java) [10 points]: multiple rounds of moves to the end of the game.

HW6 Extra Credit 2 via hw6_extra2.c (or HW6Extra2.java) [10 points]: Smarter ghosts know that Pac-man likes dots and will more likely move to cells with dots than those without dots. Hence, the ghosts would prefer their paths to have fewer empty cells. One approach is to increase the "distance" between adjacent cells that are empty. For Extra Credit 2, the distance between two adjacent cells is:

- 1 if both have dots
- 2 if one has a dot and the other does not
- 3 if both do not have dots

To find the shortest path, use Dijkstra's algorithm. One round of the first move from the player and the first move from the ghosts.

HW6 Extra Credit 3 via hw6_extra3.c (HW6Extra3.java) [10 points]: same as Extra Credit 2, but multiple rounds of moves to the end of the game.

We will evaluate your submission on code01.fit.edu. Hence, we strongly recommend you to test your program on code01.fit.edu.

Input: Command-line argument for hw6.c (or HW6.java) is a filename of the 2D grid world—the first line has number of rows and columns of the world, the following lines have the initial world represented by these characters:

- P represents Pac-man
- . represents a dot that Pac-man likes to consume
- G, H, O, S represent ghosts (up to 4) with a dot in the cell
- g, h, o, s represent ghosts without a dot in the cell
- # represents a stationary obstacle
- a space represents empty

During the game, via the keyboard, the player can input u, d, l, and r to indicate moving up, down, left, and right to an adjacent cell. If the input is invalid (incorrect letter or the adjacent cell is not empty), prompt the player to re-enter.

Output: Output goes to the standard output (screen):

1. the world with row numbers on the top and column numbers on the left
2. Please enter your move [u(p), d(own), l(elf), or r(ight)]:
3. the world with row numbers on the top and column numbers on the left
4. Points:
5. For each ghost (in alphabetical order), display its move (u/d/l/r), length of its shortest path to Pac-man, and cells on the shortest path starting with the ghost cell before the move and ends with Pac-man's cell:
   Ghost g: *move shortestPathLength* (*row*1,*col*1) (*row*2,*col*2) ...
   ...
   Ghost s: *move shortestPathLength* (*row*1,*col*1) (*row*2,*col*2) ...

Row 0 is at the top and column 0 is on the left. Sample output (with player keyboard input) is on the course website.

For extra credit, the program repeatedly displays the output above and terminates after:

1. no dots remain: the program displays the final world, points, and "Pac-man is full!", or
2. one of the ghosts at Pac-man's cell: the program displays the final world and "A ghost is not hungry any more!"

Sample input files and output are on the course website.

Submission: Submit hw6.c (or HW6.java) that has the main method and other program files. Submissions for Individual and GroupHelp have the same guidelines as HW1.

For Extra Credit 1, 2, and/or 3, submit hw6_extra1.c, hw6_extra2.c, hw6_extra3.c (HW6Extra1.java, HW6Extra2.java, and/or HW6Extra3.java) that have/has the main method and other program files. GroupHelp and late submissions are not applicable.

Note the late penalty on the syllabus if you submit after the due date and time as specified at the top of the assignment.