

UNIVERSIDAD CARLOS III DE MADRID

APRENDIZAJE AUTOMÁTICO

COMPUTER SCIENCE ENGINEERING

Tutorial 1: Plataforma PacMan

Authors:

Daniel MEDINA GARCÍA

Alejandro RODRÍGUEZ SALAMANCA

February 8, 2016

Contents

1 Preguntas:	2
1.1 Pregunta 1	2
1.2 Pregunta 2	2
1.3 Pregunta 3	2
1.4 Pregunta 4	3
1.5 Pregunta 5	4
1.6 Pregunta 6	4
1.7 Pregunta 7	4

1 Preguntas:

1.1 Pregunta 1

¿Qué información se muestra en la interfaz? ¿Y en la terminal? ¿Cuál es la posición que ocupa PacMan inicialmente?

En la interfaz se muestra el tablero con PacMan. En la parte inferior podemos observar los indicadores de qué fantasmas han sido comidos, así como la puntuación de la partida y cuatro números, cada uno del color de un fantasma, que se corresponden con la distancia que hay desde PacMan a los fantasmas

En la terminal no se muestra ninguna información.

La posición que ocupa PacMan al inicio de la partida depende del mapa usado. En ciertos mapas, PacMan se encuentra en la esquina superior izquierda, en otros, PacMan sale en la fila interior, centrado.

1.2 Pregunta 2

Según tu opinión, ¿qué datos podrían ser útiles para decidir lo que tiene que hacer PacMan en cada momento?

Los datos que podrían interesarnos para decidir qué dirección debe tomar PacMan son:

- **Distancia de PacMan a los fantasmas:** sabiendo la distancia que hay de PacMan a los fantasmas, podremos decidir en cada turno qué objetivo es el más cercano. Dado que en algunos mapas existen paredes que PacMan no puede atravesar, habría que encontrar una función que tuviese esto en cuenta a la hora de calcular la distancia, ya que puede que PacMan y un fantasma se encuentren a una distancia de dos casillas en el mapa, pero entre medias exista una pared, por lo que la distancia real sería mayor.
- **Dirección en la que se encuentran los fantasmas:** Si PacMan sabe en qué dirección se encuentran los fantasmas, podría dirigirse directamente en esa dirección. La distancia únicamente, no ayudaría a PacMan a decidir qué dirección tomar.
- **Densidad de fantasmas en una zona:** Con este parámetro, podríamos saber si en cierta zona del mapa se encuentran más fantasmas, y tomar la decisión de dirigirnos hacia esa zona en vez de hacia otra zona en la que solo se encuentre un fantasma, pese a que éste se encuentre a una distancia menor, ya que existiría la posibilidad de eliminar más fantasmas.

1.3 Pregunta 3

Revisa la carpeta layouts. ¿Cómo están definidos los mapas en estos ficheros? Diseña un mapa nuevo, guárdalo y ejecútalo en el juego.

Están definidos por caracteres en archivos de texto con la extensión .lay. Cada carácter representa una casilla en el mapa.

- **% - Paredes.** Son casillas que PacMan no puede atravesar
- **G** - Representa la posición inicial de los fantasmas.
- **P** - Representa la posición inicial de PacMan.
- **.** - Casilla que PacMan puede comer. En esta implementación estas casillas no son tenidas en cuenta.
- **o** - Casilla que PacMan puede comer. En esta implementación estas casillas no son tenidas en cuenta.
- Además, si el formato de la penúltima fila del mapa es del tipo % % % % % % % % % % % % % % % %, en el juego se verá qué fantasmas han sido comidos por PacMan en la parte inferior del mapa.

El nuevo mapa creado por nosotros es el siguiente:

```

%%%%%%%%%
%                %
%   %%%   %%%   %
%   %G     G%   %
%   %%%%%%%%%%   %
%               P   %
%   %%%%%%%%%%   %
%               %   %
%   %%%%%%%%%%   %
%   G   %   G   %
%       %       %
%       %       %
%               %
%%%%%%%%%
% % % % %%%%%%%%%%
%%%%%%%%%

```

1.4 Pregunta 4

Analiza el fichero game.py. ¿Qué información ofrece este código sobre el estado del juego en cada turno? De esta información, ¿cuál crees que podría ser más útil para decidir automáticamente qué tiene que hacer PacMan?

Los datos almacenados en cada turno son:

- Posición y dirección de los personajes del juego (Pac-Man y fantasmas).

- Estado del agente (configuración, velocidad, ...)
- Lista de fantasmas que han sido comidos y los que todavía siguen vivos
- Puntuación de la partida
- Historial de movimientos

1.5 Pregunta 5

Programa una función que imprima en un fichero de texto toda la información del estado del PacMan. Esta función servirá como una primera versión de la fase de extracción de características que será imprescindible en las siguientes prácticas.

- *Por cada turno de juego, se debe guardar una línea con todos los datos concatenados del estado del juego que se calculan por defecto, separados por el carácter coma (,).*
- *Además, cada vez que se inicie una nueva partida o se abra el juego de nuevo, las nuevas líneas deben guardarse debajo de las antiguas. Es decir, que no se debe reiniciar el fichero de texto al empezar una nueva partida.*
- *Por tanto, el fichero de texto resultante tendrá que tener tantas líneas como turnos se hayan jugado en todas las partidas.*

1.6 Pregunta 6

Implementa manualmente un comportamiento para el PacMan. Para ello se debe modificar la clase del agente GreedyBustersAgent que se encuentra dentro del fichero bustersAgents.py. Esta clase es sólo una plantilla que sirve como punto de partida. PacMan debe perseguir y comerse a todos los fantasmas de la pantalla.

1.7 Pregunta 7

El agente programado en el ejercicio anterior no utiliza ninguna técnica de aprendizaje automático. ¿Qué ventajas crees que puede tener el aprendizaje automático para controlar a PacMan?