

**uc3m**

---

Universidad  
**Carlos III**  
de Madrid

# Tutorial 1. Pacman

Aprendizaje Automático

Grupo 50

David Palomero Méndez 100333208

Juan Ortega Salinas 100332835

# Pregunta 1

La interfaz muestra los límites del mapa, los fantasmas, las comidas, el Pac-man, la puntuación y la distancia a cada fantasma (por colores).

Por la terminal no se muestra nada al inicio, pero mueve Pac-man comienza en la posición definida en cada mapa con la letra P. Si en un mapa se definen dos P, aparecen dos Pac-man, pero solo se mueve uno de ellos y al otro lo cuenta como fantasma.

Cuando se come una comida en la terminal aparece un Remove y en la interfaz desaparece la comida.

# Pregunta 2

La posición del pac-man, la posición de cada fantasma, para calcular la distancia mínima y la puntuación. En caso de que se quiera conseguir la máxima puntuación posible habría que tener en cuenta la posición de cada comida.

# Pregunta 3

Los laberintos están definidos por caracteres, de forma que cada caracter se asocie a un tipo de elemento del laberinto:

- Muro: %
- Pacman: P
- Fantasma: G
- Comida: .
- Comida especial: o



## Pregunta 4

La interfaz es la misma que en el primer apartado, pero la terminal muestra más información que en caso previo. Por cada estado del juego, que se corresponde con un tic muestra:

- Dimensiones del tablero
- Posición del pac-man
- Acciones legales
- Dirección en la que se mueve
- Número de fantasmas
- Número de fantasmas vivos
- La posición de los fantasmas
- La dirección de los fantasmas
- La distancia a las fantasmas
- Número de comidas
- Distancia a la comida más cercana
- Representación del mapa

Consideramos más relevantes los datos de:

- Posición del pac-man
- Acciones legales
- Posición de los fantasmas
- Fantasmas vivos

## Pregunta 5

El método que se nos pide implementar se llama *printLineData* y lo que hace es generar un string con los datos que consideramos relevantes para el funcionamiento del pacman. Los datos que recogemos son: posición del Pacman, acciones legales, fantasmas vivos, posición de los fantasmas y distancia a los fantasmas.

En cada tic del juego, llamamos a la función y escribimos la cadena generada en un fichero, seguido de un salto de línea.

## Pregunta 6

En este apartado se nos pide implementar un comportamiento para que pacman resuelva el mapa.

Nuestra solución busca el fantasma más cercano y va a por él, hasta terminar con todos los fantasmas y en caso de quedarse atascado se moverá aleatoriamente hasta el final.

Para esto lo primero que hacemos es comprobar que fantasmas están vivos, los guardamos para tener almacenadas sus posiciones y con dichas posiciones calculamos la distancia mínima. Una vez está calculada la distancia, asignamos a Pacman la dirección de movimiento hacia el fantasma correspondiente. Si se encuentra con un obstáculo en la trayectoria que se le ha asignado, se pondrá en modo aleatorio hasta que finalice el juego.

## Pregunta 7

Una de las posibles ventajas que podría tener es la capacidad de interpretar los mapas de forma predictiva, de forma que fuese capaz de aprender los mapas y las posibles configuraciones, para optimizar los movimientos y los cálculos de las distancias en función de la posición de los muros. Este caso del mapa se puede extender a la versión con fantasmas móviles, en la que el pacman podría utilizar los muros para “escondarse” de los fantasmas y realizar así movimientos más seguros.

Por otro lado, la implementación heurística de la solución requiere una capacidad de cómputo muy elevada, que se incrementa notablemente a medida que se complica el mapa y se aumenta el número de fantasmas.