

# **Práctica 1 Inteligencia Artificial**

## **Agentes Reactivos**

**Javier Martín Gómez**  
**2ºC**

La práctica se divide en 7 comportamientos distintos, donde interactúo con los distintos elementos del mapa, sea la naturaleza, npcs u objetos, también implemento el dibujo del mapa gracias a los sensores y el uso de una matriz auxiliar para conseguir más dibujo de mapa y el reinicio del agente.

Intenté implementar el mapaPulgarcito para que pudiese recorrer casi la totalidad del mapa, pero la implementación no daba lo que yo quería y al faltarme tiempo tuve que abandonar esta idea.

A raíz del problema anterior quise al menos implementar un movimiento aleatorio cada X pasos pero daba violaciones de segmento y tuve también que descartar la idea.

Al final me he quedado con estos comportamientos dando porcentajes de más del 70% de aciertos en los mapas de 100.

## **1. Brújula y movimiento aleatorio**

La brújula nos indica a dónde estamos orientados en cada momento, este trozo de código nos lo daban en el tutorial.

También he creado una función llamada 'movimientoAleatorio' para que cuando en ciertas situaciones donde tenga que girar a la izquierda o a la derecha, pueda hacerlo de manera aleatoria. De esta manera puedo asegurarme no entrar en bucles y así recorrer más mapa.

## **2. Comportamiento ante la naturaleza (bosque, agua, precipicio...)**

Mi agente reactivo toma decisiones simple ante los elementos de la naturaleza que tiene delante.

Si tiene bosque delante se adentra en él si tiene el objeto 'zapatillas' equipadas, en caso que no las tenga en la mano pero sí en la mochila se las equipa y si no las tiene gira a la derecha o izquierda de manera aleatoria con una llamada a la función 'movimientoAleatorio'.

Si es agua lo que tiene delante realiza el mismo procedimiento que el bosque pero usando el objeto 'biquini'.

En caso de tener una puerta delante también realiza el mismo procedimiento

anterior pero con el objeto 'llaves'

Por último para los elementos 'precipicio' y 'muro' si se los encuentra delante gira a un lado o a otro de manera aleatoria con la llamada a la función 'movimientoAleatorio'.

### **3. Comportamiento ante objetos**

El agente realiza dos acciones muy simples en caso de que tenga un objeto delante de él.

Si no tiene nada en la mano lo coge con la acción 'actPICKUP' pero en caso de que tenga algo en la mano lo guarda en la mochila con la acción 'actPUSH' y recoge el objeto que tiene delante.

No he implementado el caso cuando la mochila esté llena y tenga un objeto delante ya que suelo reiniciar antes que pase o he usado un objeto.

Si lo tuviese que implementar, al colisionar con el objeto tiraría o el que tengo en la mano o en la mochila y cogería el que tuviese delante.

### **4. Comportamiento ante NPCs**

Hay 2 npcs simples en el juego, el lobo y el aldeano.

Mi agente, al tener un lobo delante comprueba si tiene el objeto 'hueso' en la mano. En caso de tenerlo se la da al lobo con la acción 'actGIVE' y el lobo desaparece.

En caso de tener el hueso en la mochila, tira lo que tengo en la mano y me equipo el hueso para luego dárselo al lobo.

En caso de no tener un hueso en la mano ni en la mochila, esquivo de manera aleatoria a la derecha o izquierda con una llamada a la función 'movimientoAleatorio'.

Para el aldeano como no reacciona ante ningún objeto, simplemente lo esquivo de manera aleatoria con la función 'movimientoAleatorio'.

### **5. Dibujar mapa según lo que vean los sensores**

En un principio mi agente dibujaba lo que tenía bajo sus pies pero ahora dibuja todo lo que vea los sensores.

Cuando el agente alcanza una casilla 'K', el agente reconoce en qué posición del mapa está y empieza a dibujarla.

Con un switch-case de la brújula, dependiendo de la orientación puedo dibujar todo el mapa que rastrea los sensores consiguiendo así un porcentaje de mapa amplio.

## **6. Mapa auxiliar, volcar el mapa auxiliar en el mapa resultante**

Se desperdicia mucho mapa y tiempo antes de encontrar una casilla 'K' por lo que he creado una matriz auxiliar del doble de tamaño que la original.

En esta matriz he ido desde el primer momento metiendo lo que los sensores han rastreado y estoy en una casilla 'K' vuelco todo lo que he encontrado en la matriz auxiliar en la matriz resultante consiguiendo así descubrir mucho más mapa y desperdiciando el menor tiempo posible.

## **7. Reinicio**

Cuando el personaje se queda sin vida un sensor llamado reset se pone a true.

Cuando esto ocurre entro en un if que pone las variables al valor que tiene el constructor en el .hpp Para poder así comenzar de nuevo y seguir recorriendo el mapa con normalidad