



UNIVERSIDAD DE GRANADA

GRADO EN INGENIERÍA INFORMÁTICA

Práctica 1: Agente Reactivo

Autor:
Yeray López Ramírez

Curso: 2º C
Asignatura: Inteligencia Artificial
Fecha: 06 de Abril de 2022

Índice

1. Objetivo de la Práctica	2
2. Comportamiento general	2
3. Resumen de ejecuciones	3
4. Observaciones	4

1. Objetivo de la Práctica

En esta práctica se nos presenta a un jugador que se sitúa en un mundo de BelKan con 5 niveles diferentes. Mi objetivo principal es conseguir que el jugador descubra el máximo mapa posible antes de agotar los ciclos con la restricción de que su comportamiento sea esencialmente reactivo. Por ello describiré el comportamiento general de mi jugador y lo desarrollaré después.

2. Comportamiento general

El jugador que he diseñado consta de 4 fases: Preparar, Decidir, Ejecutar y Actualizar. Se explica detalladamente en los siguientes párrafos:

1. En su primera fase comprueba su estado previo y actual. Apartir de esa información actualiza el mapa, su posición y orientación. Siempre entra en esta fase.
2. En su segunda fase toma una decisión. Entrará en esta fase siempre que no tenga nada decidido, en este caso, que la cola “ejecutor” esté vacía. Las decisiones las toma entrando a las diferentes funciones que dotan de inteligencia al jugador. Su preferencia es:
 - Primero de todo busca una entrada, que viene a ser un hueco entre 2 casillas impene-trables (Muro o Precipicio) y transitable (S,T,A+bikini,B+zapatilla, etc). Con esto consigue entrar en “edificios”, o pasar tramos de muros con facilidad.
 - En caso de tener poca batería (por debajo de 1000), irá a por la casilla de recarga siempre y cuando la haya visto alguna vez.
 - Si se encuentra en la casilla de recarga, espera recargando.
 - Si sigue sin decidirse entonces busca una casilla de Interés. Esa casilla puede ser G,K,D,X pero debe respetar las condiciones de si tiene el objeto, está bien posicionado o le falta batería para así evitar que vaya a por ellas innecesariamente.
 - En caso de llegar al final y no tener decidido nada entonces procede a ponerse en modo Exploración. En este modo el jugador hará lo siguiente:
 - El jugador irá hacia delante cuando no haya obstaculos y cuando se “aburra”. El ju-gador se “aburre” cuando avanza adelante sin ver cambios en el mapa y entoces gira aleatoriamente. El sistema de “aburrimiento” le dota de un comportamiento aleato-rio.
 - En caso de que haya obstaculos girará hacia un lateral que esté libre. Elige aleatoria-mente si puede girar a ambos lados.
3. En su tercera fase ejecuta las acciones guardadas en la cola de ejecución. Se introduce el frente de la pila en la variable de Accion y se elimina de la cola para pasar a la siguiente orden. Siempre que la cola NO esté vacía, ignorará la fase 2 y entrará directamente en esta fase hasta finalizarla.
4. En su cuarta fase actualiza las variables de aburrimiento, ultimaAccion y recuerda el terreno a su alrededor para el ciclo posterior. También imprime el estado del jugador antes de finalizar.

3. Resumen de ejecuciones

He implementado y modificado una macro que permite ejecutar todos los mapas para testear la calidad del comportamiento del jugador:

```
#!/bin/bash

DIRECTORIO="Resultados"
output=$DIRECTORIO/resultados

#./practica1SG mapa semilla nivel fila column sentido

echo -e "\nMapa: ./mapas/mapaEspecial.map" >> $output.dat
for (( c=0; c<=4; c+=1 ))
do
    echo -e -n "Nivel $c: " >> $output.dat
    ./practica1SG ./mapas/cuadrícula.map 0 $c 50 20 0 | tail -n1 >> $output.dat
done

...

echo "Mapa: ./mapas/mapaXX.map" >> $output.dat
for (( c=0; c<=4; c+=1 ))
do
    echo -e -n "Nivel $c: " >> $output.dat
    ./practica1SG ./mapas/mapa30.map 0 $c 10 20 0 | tail -n1 >> $output.dat
done

...
```

Tal y como está el código, arroja el siguiente resultado:

```
Mapa: ./mapas/mapa30.map
Nivel 0: Porcentaje de mapa descubierto: 100
Nivel 1: Porcentaje de mapa descubierto: 100
Nivel 2: Porcentaje de mapa descubierto: 100
Nivel 3: Porcentaje de mapa descubierto: 100
Nivel 4: Porcentaje de mapa descubierto: 100

Mapa: ./mapas/mapa50.map
Nivel 0: Porcentaje de mapa descubierto: 99.8
Nivel 1: Porcentaje de mapa descubierto: 57.84
Nivel 2: Porcentaje de mapa descubierto: 57.84
Nivel 3: Porcentaje de mapa descubierto: 99.88
Nivel 4: Porcentaje de mapa descubierto: 99.92
```

```
Mapa: ./mapas/mapa75.map
Nivel 0: Porcentaje de mapa descubierto: 82.1689
Nivel 1: Porcentaje de mapa descubierto: 89.6
Nivel 2: Porcentaje de mapa descubierto: 89.6
Nivel 3: Porcentaje de mapa descubierto: 73.5644
Nivel 4: Porcentaje de mapa descubierto: 72.64
```

```
Mapa: ./mapas/mapa100.map
Nivel 0: Porcentaje de mapa descubierto: 18.26
Nivel 1: Porcentaje de mapa descubierto: 17.7
Nivel 2: Porcentaje de mapa descubierto: 17.7
Nivel 3: Porcentaje de mapa descubierto: 74.83
Nivel 4: Porcentaje de mapa descubierto: 63.36
```

```
Mapa: ./mapas/medieval.map
Nivel 0: Porcentaje de mapa descubierto: 68.09
Nivel 1: Porcentaje de mapa descubierto: 42.44
Nivel 2: Porcentaje de mapa descubierto: 42.44
Nivel 3: Porcentaje de mapa descubierto: 40.13
Nivel 4: Porcentaje de mapa descubierto: 54.09
```

4. Observaciones

Principalmente lo que he visto es que hay una diferencia entre ejecutar la practica en modo gráfico(./practica) y en modo terminal (./practicaSG).

Como se puede comprobar en los datos arrojados anteriormente, en el mapa100 los porcentajes caen en picado en los niveles bajos mientras que se inflan en el nivel 3 y 4.

Sin embargo al ejecutar exactamente lo mismo en el modo grafico se obtiene al rededor del 62 % en todos los niveles del mapa100. Espero que se tenga en cuenta eso.

También quisiera puntualizar que hay quizás cierto comportamiento No Reactivo ya que la función de busca batería no la consideraría como tal aunque se ha dado el visto bueno.

Por último, quedaría considerar que el código actual tiene bastante potencial de ampliación. Una mejor planificación por mi parte hubiera resultado en una mayor puntuación pero estoy satisfecho con el resultado.