

Práctica 2
Agentes Reactivos
El robot trufero
Inteligencia Artificial
Grado en Ingeniería Informática
Granada, 12 de Abril de 2015.

Datos del estudiante

Fernández Bosch, Pedro
76422233-H

1. Descripción de la solución planteada.

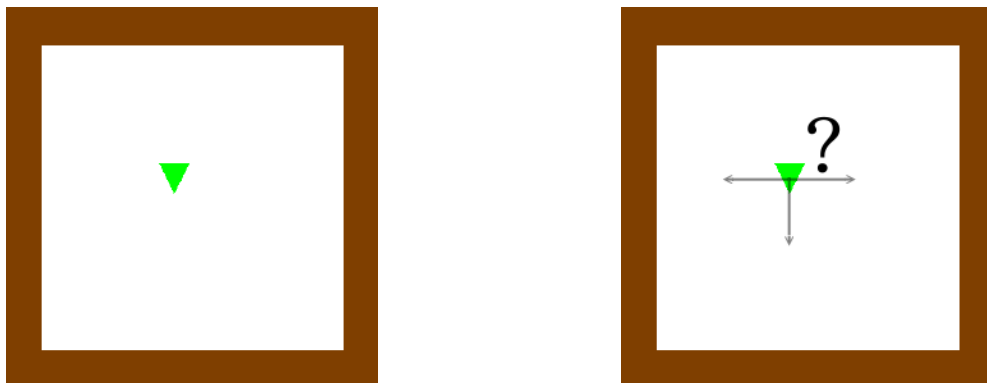
Introducción:

Esta práctica consiste en el diseño e implementación de un agente reactivo que resuelva el problema del robot truero de la mejor forma posible, asumiendo como complejidad adicional, que el entorno es dinámico. Por este motivo, el agente ha de ser capaz de percibir el ambiente y actuar de acuerdo a unas reglas simples predefinidas.

Especificaciones de diseño:

Asumimos que el entorno puede representarse en una matriz de tamaño máximo 10x10 (mundo icónico) cerrado y sin salida, donde los bordes de la matriz sólo pueden ser obstáculos. Dado que se va a trabajar sobre un entorno cuadriculado, podremos establecer un sistema de rangos a cada posición de la matriz.

Este robot truero va a basar sus reglas de movimiento en el desplazamiento hacia la posición vecina (frente, izquierda, derecha) de menor rango, es decir, la posición vecina que el robot lleva más tiempo sin visitar.



En el momento en que el robot avanza hacia una nueva posición de la matriz, establece el rango de dicha posición a 0, y aumenta en una unidad el rango del resto de posiciones que conforman la matriz de 10x10. Con este sencillo método, es posible reconocer las posiciones con mayor o menor rango, evitando pasar por las últimas posiciones recorridas por el robot truero y realizar nuevas rutas.

Dado que al comenzar la ejecución no se puede saber en qué posición de la matriz va a situarse el robot, siempre se va a denotar la posición inicial como la (0,0) y se ha optado por trabajar con posiciones mod 10.

Fase de actualización:

En primera instancia, cuando comienza la ejecución el robot truero, se inicializan todas las posiciones de la matriz (mapa) con rango valor 1000. Esta inicialización, denota que ninguna de las posiciones haya sido explorada.

Al ejecutar una iteración, el programa Incrementa el rango de todas las posiciones mayores que 0, es decir, todas aquellas posiciones que no son obstáculos.

La posición actual del agente, es actualizada con valor 0, porque acaba de ser visitada.

En caso de que el agente choque con un obstáculo, lo representamos su posición en la matriz con el valor -1 (un valor menor que 0 que no se va a volver incrementar jamás).

Fase de decisión:

Con los datos actualizados, el agente ha de ser capaz de tomar decisiones lo más acertadas posibles y ejecutar acciones en consecuencia.

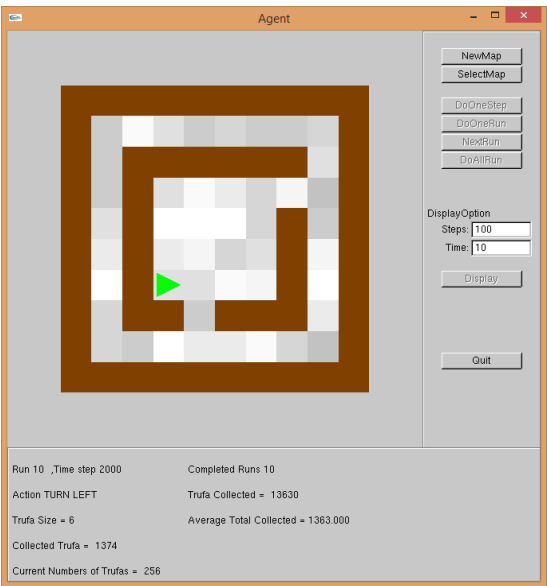
Si en la posición actual, el robot detecta más de 10 trufas, las extrae (actEXTRACT). Si en la posición actual, el robot no sabe cuantas trufas hay, olfatea (actSNIFF). Y si tiene que desplazarse, compara los valores de sus vecinos (Arriba, Derecha, Abajo, Izquierda) y elige el más adecuado para avanzar.

Las prioridades para elegir destino entre sus vecinos son:

- 1) Una posición que nunca hemos visitado antes.
- 2) La posición que hace más tiempo que hemos visitado.

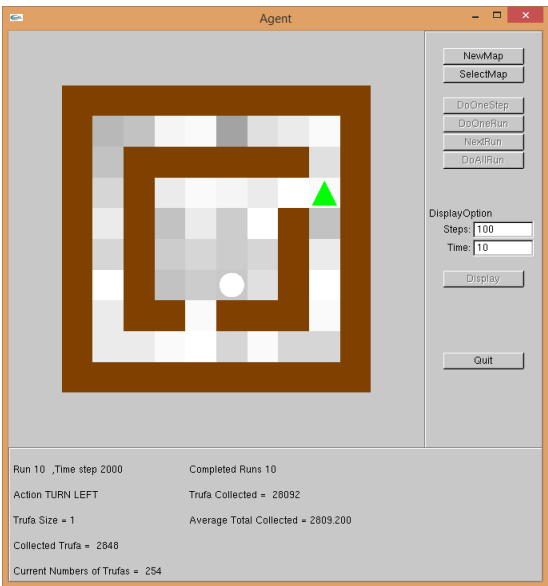
Cuando el agente elige la acción más apropiada (entre: actFORWARD, actTURN_R, actTURN_L), la realiza y guarda la última acción realizada para conocerla en la siguiente iteración.

2. Resultados obtenidos por la solución aportada en los distintos mapas que se entregan con el agente.



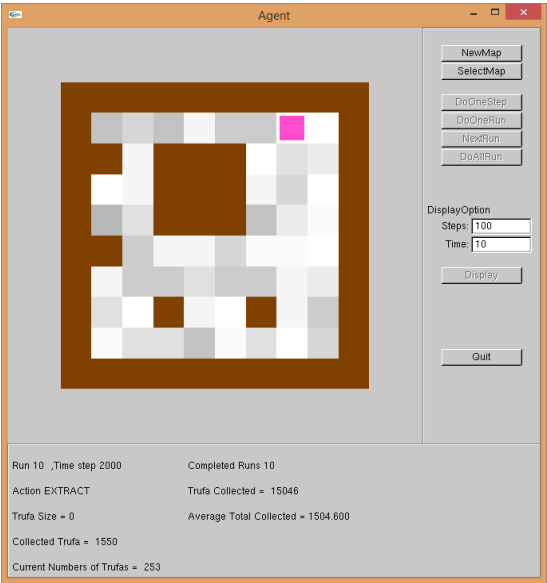
Agent.map

1 Run - Collected Trufa: 1.374
10 Runs – Trufa Collected: 13.630

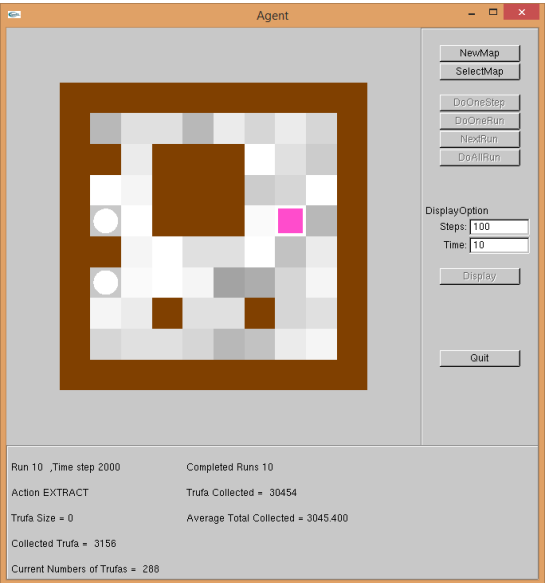


Agent_rap.map

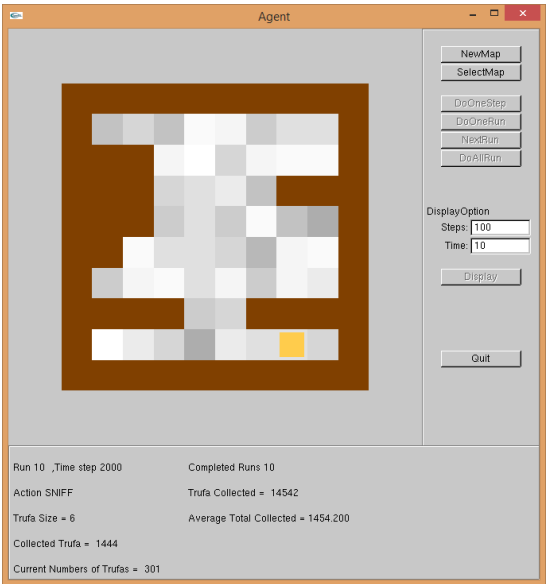
1 Run - Collected Trufa: 2.848
10 Runs – Trufa Collected: 28.092



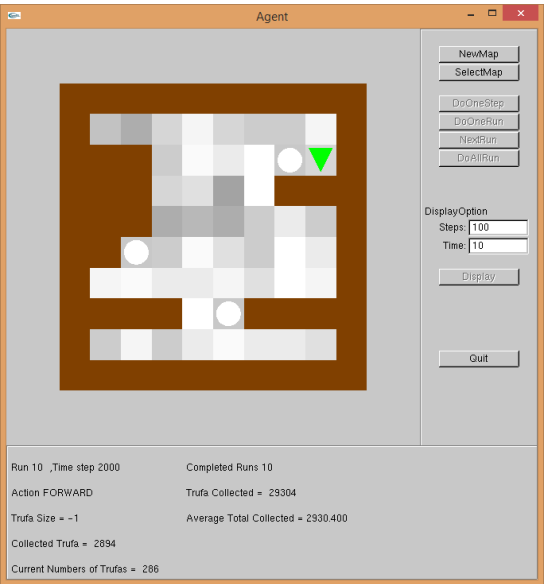
mapa1.map
1 Run - Collected Trufa: 1.550
10 Runs – Trufa Collected: 15.046



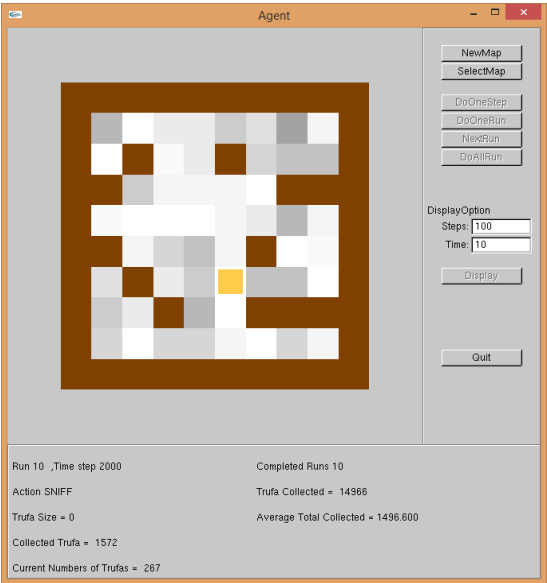
mapa1_rap.map
1 Run - Collected Trufa: 3.156
10 Runs – Trufa Collected: 30.454



mapa2.map
1 Run - Collected Trufa: 1.444
10 Runs – Trufa Collected: 14.542

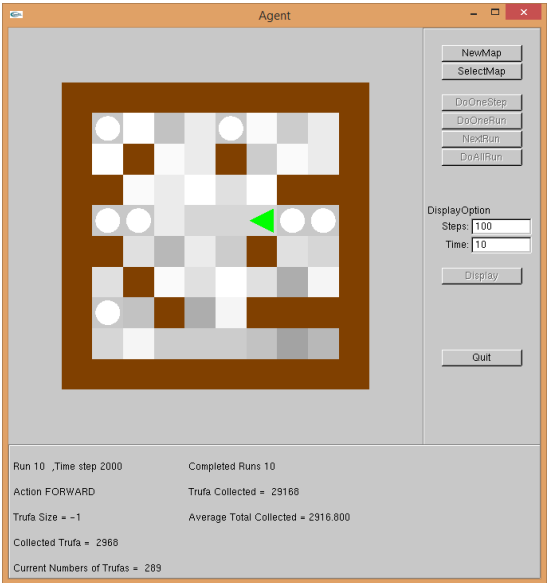


mapa2_rap.map
1 Run - Collected Trufa: 2.894
10 Runs – Trufa Collected: 29.304



mapa3.map

1 Run - Collected Trufa: 1.572
10 Runs – Trufa Collected: 14.966



mapa3_rap.map

1 Run - Collected Trufa: 2.968
10 Runs – Trufa Collected: 29.168

3. Descripción de otras estrategias descartadas.

- 1. En primera instancia se había barajado la posibilidad de que el robot trufero se desplazase a la esquina superior derecha de la matriz de posiciones nada más empezar, porque no se conoce la posición inicial del agente y por lo tanto parecía imposible dibujar correctamente el mundo icónico en el mapa de otra forma.

No obstante, después de las primeras pruebas se ha descartado esta opción debido a que identificar esta posición puede ser más complicado de lo que parece a priori, ya que algunos obstáculos podrían hacernos creer que se trata de la esquina de la matriz (un camino en forma de espiral, por ejemplo).

Tras descartar esta primera estrategia de posicionamiento, se ha diseñado el problema de tal forma que el agente siempre comience en la posición inicial (0,0) y las posiciones de la matriz son identificadas en base a mod 10.