Integrantes:

Rodrigo Martín Díaz Figueroa – 97400

[rodrigomdiaz@live.com](mailto:rodrigomdiaz@live.com)

Federico Germán Agra – 94186

[federico.agra@gmail.com](mailto:federico.agra@gmail.com)

DESARROLLO DE IA PARA AGENTES EN UN ENTORNO VIRTUAL

2do Cuatrimestre – Año 2017

Inteligencia Artificial

Proyecto de Cursado – Etapa 1

Índice

|  |  |
| --- | --- |
| Descripción ………………………………………………………………. | 2 |
| Descripción del entorno …………………………………………………………. | 2 |
| Descripción de las entidades ………………………………………………….. | 2 |
| Discretización del terreno ………………………………………………………. | 2 |
| Objetivo del juego ………………………………………………………………….. | 2 |
| Consideraciones ……………………………………………………………………… | 3 |
| Objetivo …………………………………………………………………… | 3 |
| Estrategia …………………………………………………………………. | 4 |
| Actualización de creencias ……………………………………………………… | 4 |
| Buscar plan de búsqueda ……………………………………………………... | 5 |
| Descripción de Predicados ……………………………………….. | 5 |
| Consideraciones generales ………………………………………. | 6 |

Descripción

Descripción del entorno

El juego se desarrolla sobre un mundo virtual, en el cual se simulan las leyes de la física del mundo real tales como la gravedad, la fricción con el suelo, etc. Este mundo virtual consiste en una porción de terreno con elevaciones, obstáculos, edificaciones, objetos y agentes que interactúan con el terreno para satisfacer diferentes metas. El modo de juego es multijugador, tanto cooperativo como competitivo, ya que cada agente trabajará con un equipo de agentes de manera cooperativa, para juntos competir con otro equipo de agentes.

Descripción de entidades

Entre las edificaciones se distinguen: las *Posadas*, en donde los agentes pueden descansar para recuperar puntos de vida (*life*); las *Tumbas*, que suelen encerrar tesoros en su interior; y los *Homes,* los cuales constituyen bases para los agentes.

Los tesoros y pociones comprenden al grupo de objetos que se pueden encontrar en el terreno. Los tesoros incrementan la puntuación del agente y/o equipo que logre apoderase de ellos. En cambio, las pociones se utilizan para abrir las *Tumbas* y así poder obtener los tesoros que yacen dentro.

Los personajes (agentes) contienen atributos (como puntos de vida, de experiencia, puntaje total, etc.) y capacidades (como la de atacar a otros agentes o recoger objetos, moverse sobre el terreno, etc.).

Discretización del terreno

Sobre la superficie del terreno se define un grafo con una distribución de nodos y arcos en forma de malla. Cada entidad del mundo se encuentra asociada, en un instante dado, a un nodo del grafo.

Los agentes conciben el mundo de acuerdo con esta discretización, y tienen la capacidad de desplazarse de un nodo a otro adyacente. Este desplazamiento puede ser eficiente si se emplea algoritmos de búsqueda sobre grafos.

El agente percibe (con una visión de 360° y de alcance limitado), en cada instante, la porción del grafo (nodo y arcos) y las entidades que se hallan dentro de su rango de visión.

Objetivos del juego

Los miembros del mismo equipo tienen el objetivo común de recolectar la mayor cantidad de tesoros y depositarlos (*drop*) en su *Home.* Este objetivo entra en conflicto con el de los agentes de otros equipos. Esto hace del juego en un entorno colaborativo y competitivo a la vez.

Al igual que las *Tumbas*, los *Homes* pueden ser saqueados utilizando una poción de apertura para liberar los tesoros que se encuentran en su interior. Luego de recibir un hechizo de apertura el *Home* volverá a quedar cerrado, listo para guardar tesoros de su equipo.

El equipo que posea más tesoros en su home al momento de finalizar la partida será el ganador.

Consideraciones

Las *Posadas* permiten a los agentes descansar y recuperar energía, pero ésta no será instantánea, sino que se producirá gradualmente y a una velocidad constante mientras el agente se encuentre dentro de ella.

Las *Tumbas* son recintos que suelen encerrar objetos de valor. La manera de acceder a dichos objetos es lanzando un hechizo de “*apertura*” contra la *Tumba*, que la abrirá dejando los objetos que posee tirados en el suelo.

Cada agente tiene asociado un único *Home*, pero varios agentes pueden compartir el mismo *Home*, lo que en la práctica significa que forman parte de un mismo equipo.

objetivo

* Adoptar una representación en Prolog de las creencias que el agente tiene del mundo (y que conforman, en parte, el estado interno del agente). Desarrollar además el subprograma para la actualización de dichas creencias a partir de la percepción corriente (”módulo” *actualizar concepción del mundo).*
* Desarrollar el subprograma de decisión para dotar al agente de la capacidad de recolectar eficientemente aquellos tesoros que yacen en el suelo. No se requiere recolectar los tesoros encerrados en *Tumbas*, ni llevarlos al home. Para ello se deberá implementar un predicado auxiliar buscar\_plan\_desplazamiento(+Metas, -Plan, -Destino) que dado un conjunto Metas de nodos metas, basándose en las creencias del agente acerca de su ubicación actual y la configuración del territorio, encuentre el nodo meta Destino al que cueste menos energía/tiempo llegar a partir de dicha ubicación actual, además de la secuencia Plan de acciones de movimiento (move(Node)) de costo mínimo hacia él. Este predicado permitirá al agente seleccionar destinos de menor costo () por ejemplo, ubicación del tesoro más cercano), y encontrar *Planes de desplazamiento* optímales hacia dichos destinos. Para la implementación de buscar\_plan\_desplazamiento/3 deberá emplearse la estrategia de búsqueda *A\**, adoptando una heurística adecuada.

estrategia

Actualización de creencias

Para el manejo de creencias se decidió desglosar el predicado para actualizar las creencias en sub-predicados, los cuales realizan una determinada tarea dependiendo el tipo de relación de cada percepción.

Primero actualiza su base de creencias a partir de las percepciones que recibe, luego recorre todos los nodos percibidos para actualizar o corroborar la base de creencias sobre objetos encontrados anteriormente.

Para la parte de actualización de creencias, se toman en cuenta los casos en que:

- Una entidad cambia de la posición en la que creía que estaba.

- Se actualiza la descripción de una entidad.

- Una entidad que creía estaba en el piso, ahora está en poder de otra.

- Una entidad que creía que estaba en poder de otra, ahora está en poder de una tercera.

- Una entidad sigue en poder de otra entidad.

- Una entidad que creía que estaba en poder de otra, y ahora está en el suelo.

- Una entidad sigue estando en la misma posición que creía que estaba.

- Un agente que tiene una moneda nueva en su poder.

- El nuevo tiempo.

- Los nuevos nodos percibidos.

Para la parte de avistamiento de nodos se contempla el caso:

- Si tenía la creencia de que en un nodo N yacía una entidad y al percibir de nuevo ese nodo verificar que dicha entidad no está en el nodo como se creía.

Buscar plan de búsqueda

Cuando el agente recibe su primera percepción, luego de actualizar sus creencias, pasa a buscar un plan de búsqueda que sea óptimo para la recolección de los tesoros esparcidos en el terreno. Para ello, el agente realiza una serie de acciones para lograr encontrar un camino óptimo desde el nodo actual hasta el tesoro visto más cercano.

Como primera instancia, se generan todos los nodos vecinos del agente (nodos adyacentes al nodo actual), los cuales se utilizan para actualizar la frontera de visión del agente. Una vez actualizada la frontera, mediante un algoritmo de ordenamiento Quick Sort, ésta se ordena según una **función de estimación de costo**1 desde el nodo actual hasta el nodo donde se encuentra el mejor tesoro, o sea, el más cercano en cuestión de costo. Luego de haber seleccionado el nodo al que se quiere llegar (el primero de la lista ordenada anteriormente), se genera un plan (camino) optimal para poder alcanzar dicho nodo utilizando el algoritmo A\*. Una vez encontrado el plan, el agente comienza a realizar las acciones del plan generado en orden.

Descripción de predicados

busquedaAEstrella(+Metas,+NodoActual,+Frontera,+Visitados,-Plan):

El predicado busquedaAEstrella consta de dos partes:

* Caso base: El nodo actual que entra como parámetro está contenido en el conjunto Metas de metas, es decir, que el agente está parado en una meta. Luego el plan es ese único nodo.
* Caso recursivo: Si en nodo actual no está contenido en el conjunto de metas, entonces selecciona el primer nodo de la frontera y lo agrega al conjunto de nodos visitados. Se generan los vecinos del nodo actual y se agregan a la frontera obtenida en el nodo anterior. Luego se ordena frontera obtenida en el paso anterior teniendo en cuenta la función de estimación de costo (de menor a mayor). Se selecciona el primer nodo de la frontera ordenada y se vuelve a llamar recursivamente a busquedaAEstrella con el nodo seleccionado.

generarVecinos(+Visitados,+Nodo,-VecinosSinRepetir):

Este predicado recibe un conjunto de nodos visitados V y un nodo N, y agrega al conjunto de visitados V todos los nodos adyacentes de N que no hayan sido visitados antes. Para esto, sigue el siguiente algoritmo: Si el conjunto de vecinos V es vacío, quiere decir que es el nodo en el que comienza el agente (nodo inicial), entonces, el conjunto de vecinos resultantes es igual al conjunto de nodos adyacentes de N. Sino, busca todos los nodos adyacentes de N que no estén visitados y los retorna en una lista.

obtenerFrontrera(+NodoPadre,+ListaVecinos,+Frontera,-FronteraNueva):

A partir de un nodo N, un conjunto de pares [Ady,Costo] (siendo Ady adyacente a N y Costo es el peso del arco [N,Ady]) y la frontera actual, devuelve una nueva frontera que contenga los nodos de la frontera actual más los nodos del conjunto de pares [Ady,Costo], todos transformados a la forma nodo(ID,CostoCamino,Camino,Heuristica). Entonces la frontera resultante sería un conjunto de nodos, los cuales conocen su ID, el costo total junto con el camino de recorrer desde el nodo inicial (nodo en el que se encuentra actualmente el agente) hasta dicho nodo de id ID y la heurística de llegar a un nodo meta desde el nodo ID.

actualizarFrontera(+NodoPadre,+IDHijo,+CostoPaso,+Frontera,-FronteraNueva):

Este predicado calcula el costo del camino entre el nodo inicial hasta el nodo hijo, además calcula la heurística de este último, que es la estimación de costo optimo entre dicho nodo y un nodo meta. Luego actualiza la frontera con nodos de la forma nodo(ID,CostoCamino,Camino,Heuristica).

ordenar\_por\_f(+FronteraDesordenada,-FronteraOrdenada):

Utiliza el algoritmo de ordenamiento Quick Sort para ordenar la lista de nodos frontera ingresada por parámetro, según su función de estimación de costo y de menor a mayor.

Consideraciones generales

1Función de Estimación de Costo

La función de Estimación de costo de un nodo N es la suma del costo de llegar desde un nodo inicial NI hasta N, más la heurística h(N), el cual es una cota inferior de mínimo costo del paso más corto desde el nodo n a un nodo meta. La heurística utilizada es la basada en la distancia euclidiana, la cual contempla la distancia a la meta más cercana en línea recta desde el nodo N.

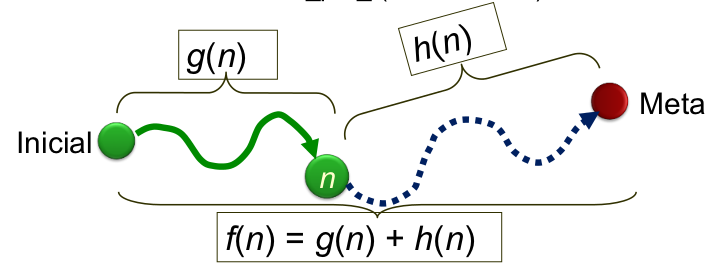


Ilustración - Función F(n)

Ilustración - Cálculo de la heurística Euclidiana

Nodo Inicial

Nodo N

Nodo Meta

h(N)

…

Representación de Nodos

En la implementación de creencias y de algoritmos de búsquedas, los nodos pueden tomar diferentes formas a modo de estructura para un mejor manejo de los mismos y mayor accesibilidad de los datos. En cuanto a creencias, se establece la notación provista por la catedra, donde un nodo se representa por la estructura:

node(ID, Pos, Conn); donde ID es el identificador del nodo, Pos es un vector de posición [X,Y,Z], y Conn es una lista de nodos adyacentes de la forma [[Ady1,C1], [Ady2,C2]…], donde Adyi es un nodo adyacente y Ci es el costo de llegar a él.

Cuando se utilizan nodos en conjuntos fronteras, estos se representan como:

nodo(ID,CostoCamino,Camino,Heuristica); donde ID es su identificador, CostoCamino y Camino son el costo y el camino respectivamente que conlleva llegar al nodo con identificador ID desde un nodo inicial (desde el que comienza el algoritmo de búsqueda de plan), y la heurística es una cota inferior de mínimo costo del paso más corto desde el nodo con identificador ID a un nodo meta.

Problemas encontrados

En ciertas ocasiones, al correr el juego con más de un agente, si alguno de los dos se demora demasiado en encontrar una meta o no existe dicho camino puede provocar un error de *“Out of global stack”* o *“Out of local stack”.* Desconocemos el origen exacto del problema pero creemos que puede ser por insuficiente memoria a la hora de realizar las llamadas de búsqueda.