

INTELIGENCIA ARTIFICIAL

E.T.S de Ingenierías Informática y de Telecomunicación.

Práctica 3

Métodos de Búsqueda con Adversario (Juegos)

Alumno: José Arcos Aneas
D.N.I: 74740565-H

Descripción del problema

El objetivo de esta practica es diseñar e implementar un agente deliberativo que pueda llevar a cabo un comportamiento inteligente dentro del juego de las aspiradoras.

El juego de las aspiradoras es un juego por turno de dos jugadores capaces de moverse en un tablero de $N \times N$ casillas.

Cada casilla puede estar sucia, contener un obtaculo o estar vacía.

El objetivo es conseguir mayor cantidad de unidades de suciedad que el adversario, considerando cada casilla con suciedad puede contener varias unidades de basura, y llevando a cabo movimientos que eviten obtaculos.

Cada jugador puede llevar a cabo los siguientes movimientos.

actFORWARD: desplaza la aspiradora a la casilla adyacente en la dirección que indica su orientación.

actTURN_LEFT: cambia la orientación de la aspiradora con un giro hacia la izquierda, manteniéndose en la misma casilla.

actTURN_RIGHT: cambia la orientación de la aspiradora con un giro hacia la derecha, manteniéndose en la misma casilla.

actIDLE: la aspiradora no realiza ninguna acción.

actSHOOT: la aspiradora lanza un disparo en la dirección en la que esta está orientada.

Estos movimientos están sujetos a diferentes restricciones.

Un movimiento siempre tiene que llevar a una casilla destino que no represente un obstáculo, que no este ocupada por otro jugador y que este dentro del tablero.

Cada vez que un jugador abandona una casilla origen para llegar a una casilla destino, el entorno se modifica de manera que aparece un obstáculo.

No se permiten mas de 4 giros de forma consecutiva.

Para que la acción disparar sea aplicable, se tiene que dar una de las siguiente condiciones.

- La aspiradora esta orientada hacia un obstáculo destructible.

- La aspiradora esta orientada hacia la otra aspiradora.

Las condiciones de parada o de finalización del juego se deben a alguna de las siguientes condiciones.

- Se recoge la ultima unidad de suciedad del mapa.

- Se supera el limite máximo de jugadas que esta fijado en 1000.

- Un jugador se queda en una situación en la que no puede aplicar ninguna acción.

- Un jugador alcanza en su marcados ella suciedad objetivo(osea la mitad mas uno de la suciedad inicial del tablero).

Descripción algoritmo a usar

A partir de estas consideraciones iniciales, el objetivo de la práctica es implementar un algoritmo MINIMAX, con profundidad limitada (con cota máxima de 10), o el algoritmo de PODA ALFA- BETA, con profundidad limitada (con cota máxima de 16) de manera que un jugador aspiradora pueda determinar el movimiento más prometedor para ganar el juego, explorando el árbol de juego desde el estado actual hasta la profundidad máxima, dada

como entrada al algoritmo.

Descripción algoritmo Minimax

La estrategia minimax es una estrategia de búsqueda exhaustiva mediante un árbol de búsqueda. Este algoritmo considera el caso de 2 participantes a los que se les denomina Max y Min

El que inicia el juego es Max y existe una alternancia en la participación del juego. Por lo tanto lo que tiene que hacer Max, es determinar la secuencia de jugadas que conduzca a un estado Terminal ganador o favorecedor.

Por otro lado, esta búsqueda muchas veces tiene que evaluar ramas innecesarias que no le traerán beneficio alguno para obtener el mejor resultado. Por lo tanto, es aquí donde se plantea la solución mediante la poda de esas ramas que no beneficiaran en nada a la mejor jugada.

Descripción poda alfabeta

Es una técnica de búsqueda que reduce el número de nodos evaluados en un árbol de juego por el algoritmo Minimax.

El problema de la búsqueda Minimax es que el número de estados a explorar es exponencial al número de movimientos.

Partiendo de este hecho, la técnica de poda alfa-beta trata de eliminar partes grandes del árbol, aplicándolo a un árbol Minimax estándar, de forma que se devuelva el mismo movimiento que devolvería este, gracias a que la poda de dichas ramas no influye en la decisión final.

Descripción de procedimientos

Procedimiento minmax de la práctica

Establecemos un límite de profundidad como condición de parada.

Proceso recursivo de cálculo de minimax

Determino las posibles acciones

Sino se puede realizar ninguna acción uno de los pierde dependiendo del nivel que se este jugando en ese momento.

Inicializo el mejor según el nivel.

Generamos los posibles movimientos.

Mientras podamos seguir generando posibles tableros

evaluamos el valor de los hijos dependiendo de quien juegue.

Generamos el nuevo tablero.

Procedimiento poda alfabetica.

Para realizar la poda alfabetica he modificado el algoritmo minimax que se entregaba en la plataforma de la asignatura.

Le pasamos dos variables alpha y beta que no ayudan a decidir cuando podar.

Cuando el jugador Max esta jugando comparamos el valor que guarda la variable "mejor" con alfa y la actualizamos en caso de que sea mejor.

Si "mejor" es mayor que beta podamos, por el contrario seguimos expandiendo.

En caso que juegue Min compramos mejor con beta y actualizamos beta si es menor o igual que beta.

Si "mejor" menos que alpha hacemos poda, en caso contrario seguimos expandiendo.

Función de valoración del estado del tablero.

Función heurística.

Esta función calcula el valor heurístico de un estado de la frontera de búsqueda.

Devuelve el valor heurístico asociado al estado "actual" desde el punto de vista del jugador "jug"

Procedimiento

Establecemos el tamaño de la matriz y el estado del mundo.

Escogemos un jugador dependiendo del jugador que se pase como argumento y asignamos posiciones a el y al rival.

Buscamos las casillas con suciedad

Y guardamos la que este mas cerca

devolvemos la menor distancia.

Mi función heurística toma como parametros el estado del tablero y el jugador que va a jugar.

En primer lugar asignamos una posición dependiendo del jugador que se pasa como parámetro.

Recorremos el estado del tablero buscando la suciedad mas cercana.

El valor de la heurística será del que depende la valoración, incluyendo un valor multiplicado por el marcador del propio jugador para que elija suciedad dando mas valor a las posiciones que en estados posteriores nos den un resultado mejor.

Luego el valor de la heurística esta multiplicado porque nos da un valor pequeño y nosotros queremos que ese estado este mejor valorado.