



Universidad de Alcalá

Escuela Politécnica Superior

Universidad de Alcalá

PECL2

Juego de la cantera

Inteligencia Artificial

Grado en Ingeniería Informática – Curso 2018/2019

Marcos Barranquero Fernández – 51129104N

Eduardo Graván Serrano – 03212337L

INTRODUCCIÓN

El juego consiste en cortar el ancho, alto o profundo de un bloque de mármol. Dos personajes, el jugador y la máquina, se van alternando turnos. En cada turno, eligen una dimensión y cuánto cortan, empezando con un bloque de $M \times N \times D$ y acabando en un bloque de $1 \times 1 \times 1$. El que no pueda realizar más cortes, pierde.

EXPLICACIÓN GENERAL DEL PROGRAMA ENTREGADO

La ejecución del programa se podría estructurar en los siguientes pasos secuenciales:

1. Pide dimensiones del bloque en formato M N D. (Por ejemplo, 5 4 5). Si se introduce algo no válido, devuelve error y vuelve a pedirlo.
2. Se elige al azar si comienza el jugador o la máquina.
3. Se entra en el bucle de juego, donde se irán alternando turnos hasta que el bloque sea de $1 \times 1 \times 1$:
 - a. En el turno de la máquina, se crea un estado basado en el estado del mármol y se ejecuta el algoritmo minimax con poda- $\alpha\beta$, eligiendo el mejor movimiento realizable.
 - b. En el turno del jugador, se pide bloque a cortar y cantidad a cortar, se verifica el input y se realiza la operación.

EXPLICACIÓN MODULAR DEL PROGRAMA

FUNCIONES DE INPUT

GET DIMENSIONES INICIALES

Función que, comprobando el input, devuelve el bloque con las dimensiones del comienzo de juego. Está preparada para capturar las excepciones, en caso de que se metan valores incorrectos o incompletos, y volver a pedir las dimensiones hasta que se introduzcan correctamente.

PEDIR INPUT

Función que, para el turno del jugador, recibe el bloque a cortar y la cantidad que se quiere cortar. Es similar a la anterior: mientras no se meta un input válido seguirá pidiendo datos.

CLASE ESTADO

La clase estado representa un estado del árbol de estados en el que se ejecuta el algoritmo minimax con poda- $\alpha\beta$. Contamos con 4 atributos:

- esMax: booleano que indica si es un estado asociado a un maximizador. Si es False, entonces es un minimizador.
- estado: es una lista que representa el bloque de mármol.
- sucesores: es una lista de estados sucesores o hijos de ese estado.
- esHoja: booleano que indica si es hoja, es decir, si no tiene sucesores y su estado es 1x1x1.

Dentro de la clase estado distinguimos varias funciones:

SOY HOJA

Verifica si es una hoja o no. Sirve para dar valor al atributo esHoja.

STR Y REPR

Devuelve una representación por pantalla del estado. Repr es usado para poder imprimir esa representación en las listas.

GENERA SUCESORES Y GENERA SUCESORES AUXILIAR

Ambas funciones generan todos los posibles estados sucesores de ese estado. Genera sucesor auxiliar devuelve todos los estados posibles disminuyendo uno de los lados del bloque hasta uno. Genera sucesores llama a la función auxiliar con cada dimensión del bloque, de forma que cubra todos los casos.

Solo se cargan los sucesores resultantes de realizar un solo movimiento. Es decir, con profundidad 1, no carga sucesores de sucesores.

MINIMAX CON PODA ALFA-BETA

Ejecuta sobre ese estado el algoritmo minimax con poda- $\alpha\beta$. Distingue entre tres casos:

- Si el estado es una hoja, devuelve 1 o -1 en función de si es Max o Min.
- Si el estado no es una hoja, se evalúa recursivamente sus sucesores, llamando para cada uno de estos al algoritmo minimax con poda alfa beta.
 - Si es max, se busca en cada uno de ellos el valor máximo. Si se encuentra el máximo, se recoge el estado asociado que lleva a ese máximo para poder devolverlo. Además, se asigna el valor a alfa.
 - Si es min, se busca en cada uno de ellos el valor mínimo. Cuando se encuentra, se recoge el estado asociado que lleva a ese mínimo para poder devolverlo. Además, se asigna valor a beta.
 - Si beta es menor que alfa, se realiza la poda.
- Finalmente, si es el estado desde el que se había iniciado la búsqueda, se devuelve el estado asociado que lleva al mejor resultado. Si es un estado hijo, sólo se devuelve el mejor valor encontrado.

FUNCIONES DE JUEGO

CORTAR BLOQUE

Dada la lista de bloques, un bloque y una profundidad, se corta la cantidad especificada del lado elegido.

BUCLE JUEGO

Se tiene una variable que indica si el juego ha terminado, inicialmente a false.

Lo primero que se hace es calcular quién comienza primero, utilizando la librería de random.

Después, mientras el juego no esté terminado, se ejecuta el juego. Si es turno del jugador, se pide el bloque a cortar con pedirInput y se corta con cortarBloque. Si es el turno de la máquina, se crea un estado asociado a la lista de bloques, se ejecuta minimax con poda- $\alpha\beta$, y se devuelve el estado que lleva a la mejor puntuación posible en ese momento. Tras esto, se intercambian los turnos y se verifica si el juego ha terminado.