TEC | Tecnológico de Costa Rica

Instituto Tecnológico de Costa Rica
Escuela de computación

Proyecto 1 de Inteligencia Artificial

Semestre II

Realizado por:
Jonathan Calet Guillén Campos 201261579
Ricardo Víquez Mora 2015013504

Abril 2021

Índice general

Capítulo 1

Introducción

El proyecto consiste en crear una aplicación multi agente que juegue la versión clásica de Pac-Man. El video juego de Pac-Mac, para los que no han jugado, consiste en que Mr. Pac-Man (un semicírculo amarillo) que tiene que comer unas bolas de energía que están repartidas a lo largo de un laberinto, mientras escapa de fantasmas que habitan en el laberinto.

El objetivo de cada nivel es limpiar el laberinto completo, es decir recoger todos los puntos distribuidos en el piso del laberinto. Si alguno de los fantasmas logra alcanzar a Mr Pac-Man este muere, y pierde una vida hasta agotarlas todas.

Capítulo 2

Ambiente de Desarrollo

2.1. Lenguaje

El lenguaje usado para este proyecto es Racket, un lenguaje funcional basado en Lisp.

2.2. Ambiente

El ambiente de desarrollo que se utilizo es DrRacket.

2.3. Manejo de versiones

Para este proyecto se decidió el uso de github como herramienta para manejo de versiones. Para poder tener la tarea en cualquier máquina para ejecutar proyecto se puede optar por descargar el proyecto desde la página o bien clonando el proyecto.

Para descargar el proyecto se puede hacer por medio de la página de Github, este este link. Este le descargara el proyecto comprimido en un archivo .zip.

En caso de que no se pueda o no se quiera descargar el proyecto desde la página se puede clonar el proyecto. Para esto se tiene que acceder a la terminal de Linux, después moverse a la carpeta que quiere clonar el proyecto e ingresar el siguiente comando:

```
git clone git@github.com:Aragox/pacman-racket.git
```

Listing 2.1: Link para descargar el proyecto.

Capítulo 3

Descripción del Sistema

El sistema utiliza el cálculo de la función Minimax con Poda alfa - beta para determinar el movimiento de los agentes en el laberinto. Cada agente toma uno de los turnos del Minimax, para el caso de este proyecto sería Max-Min-Min para Mr. Pac-Man y los 2 fantasmas respectivamente.

El movimiento de los fantasmas sigue el algoritmo de la función "Manhattan-move"para calcular la distancia que existe entre Mr. Pac-man y los Fantasmas. Para poder generar el árbol de movimientos del fantasma se utiliza el algoritmo de Branch and Bound para seleccionar la ruta de menor distancia.

Para el desarrollo del juego contamos con un interfaz proporcionado por la librería universe, la cual permite el manejo de la interfaz del escenario y de los agentes. Para manejar las imágenes de las bolas se utiliza la librería image que genera composiciones de imágenes y sobreponer la imagen de las bolitas en el laberinto.

Capítulo 4

Descripción de la Función Eval

Antes de usar la función Eval se requiere definir varias funciones previas.

Una función necesaria es calcular que tan cerca están los fantasmas; dependiendo de qué tan cerca esta el fantasma, retorna un valor mayor o menor negativo, y si los fantasmas están vulnerables retornara un valor mayor o menor positivo. También se tiene una función que calcula qué tan cerca esta Mr. Pac-Man de las esquinas, ya que se prioriza el agarrar las bolas grandes que están ubicadas en cada esquina, esto para que Mr Pac-Man se pueda comer los fantasmas. Además, se tiene una función que determina qué tan cerca están las bolas para poder recolectarlas, y así ganar más puntos.

Todas estas funciones se utilizan dentro de la función Eval para poder hacer el cálculo de mover a Mr. Pac-Man en el laberinto.

Capítulo 5

Documento de Pruebas

En esta sección se realizaron las pruebas de rendimiento en diferentes profundidades de exploración por niveles. Cada prueba se realizó para obtener el tiempo promedio de la partida y la eficiencia del código. Se tiene que:

1. El promedio de corrida de una jugada para una profundidad de 1 nivel es de 17ms con 28 muestras.
2. El promedio de corrida de una jugada para una profundidad de 2 niveles es de 26ms con 32 muestras.
3. El promedio de corrida de una jugada para una profundidad de 3 niveles es de 25ms con 45 muestras.;

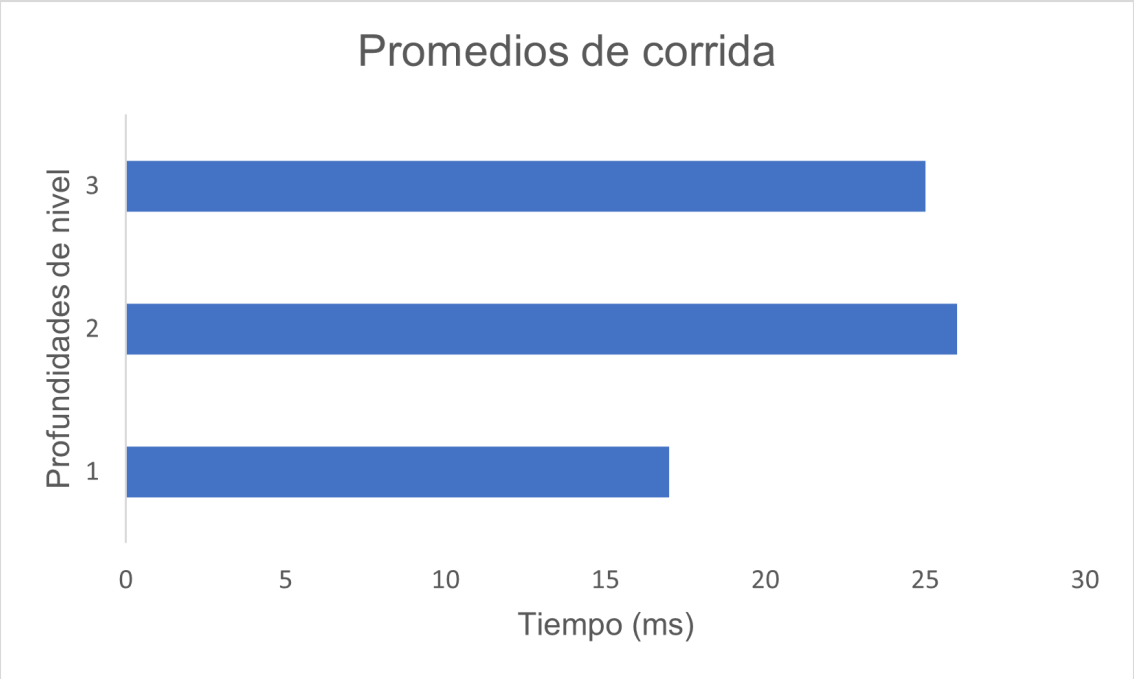

Figura 5.1: Rendimiento de la función Eval en tiempo promedio de duración por jugada, de acuerdo con la profundidad explorada de nivel.

Capítulo 6

Descripción de los Agentes

Nombre del agente	Mr. Pac-Man
Descripción y objetivo del agente	El agente es un semi circulo de color amarillo que se mueve a través de los pasillos de un laberinto. El objetivo del agente es recolectar todas las bolitas de energía que están en el laberinto mientras evita ser tocado por los fantasmas.
La función del agente	Recolectar bolas de energía y morir al encontrarse con un fantasma.
Observable	Total
Determinístico	Determinista
Episódico	Si
Estático	No
Discreto	Si
Mono agente	Multi Agente
Medida de Desempeño	Cantidad de bolas obtenidas, Número de vidas
Actuadores	Función "$change - pos$"
Sensores	Funciones "$get - positions$", "$Eval$".

Nombre del agente	Fantasmas
Descripción y objetivo del agente	El agente es un fantasma que vive en el laberinto, el cual persigue a quien quiera que entre en el laberinto. El objetivo del agente es perseguir y atrapar a quien sea que entre en el laberinto.
La función del agente	Perseguir a Mr. Pac-Man y huír de Mr. Pac-Man cuando esta en estado vulnerable
Observable	Parcial
Determinístico	Determinista
Episódico	Si
Estático	No
Discreto	Si
Mono agente	Si
Medida de Desempeño	Comer a Mr. Pac-man, Acercarse a Mr. Pac-man, Alejarse de Mr. Pac-Man
Actuadores	función "$change - pos$"
Sensores	Funciones "$get - positions$", "$stop - ghost - count$", "$vulnerable - ghosts$"

Capítulo 7

Instrucciones para ejecutar el programa

7.1. Prerrequisitos

Para la presente tarea es necesario contar con Racket instalado, por lo que es un requisito correr los siguientes comandos en una terminal de Ubuntu.

```
sudo add-apt-repository ppa:plt/racket
sudo apt-get install racket
```

Listing 7.1: Comandos para instalar Racket en Linux.

7.2. Compilación y Ejecución

Para poder compilar y ejecutar el proyecto se usa el ambiente de desarrollo de DrRacket, con el cual puede ejecutar directamente el proyecto.