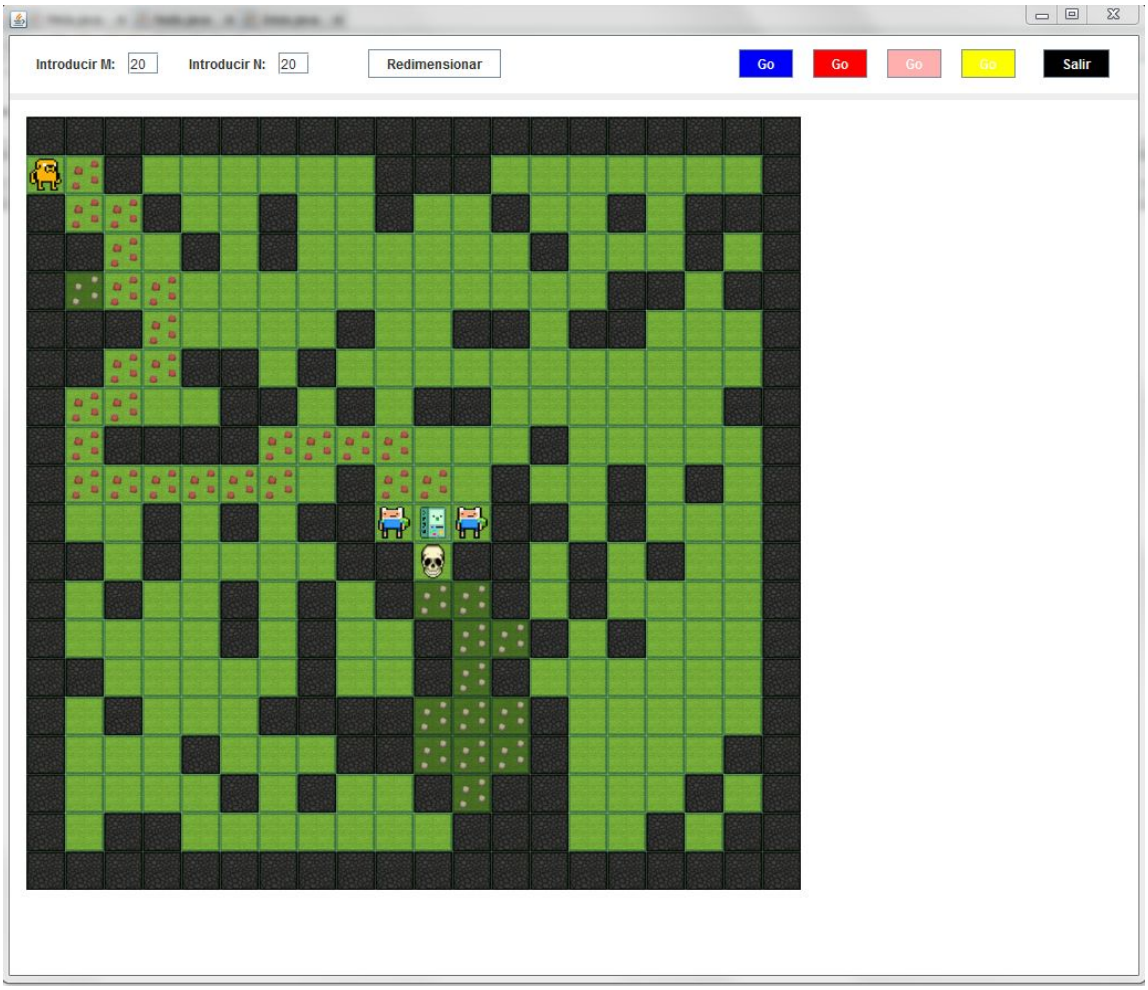
Informe del proyecto de Búsqueda Heurística

“The Maze Runner”



Integrantes: Natalie Dajakaj, Alexis Daniel Fuentes Pérez, Luis David Padilla Martín

Índice

Introducción ……1
Interfaz Gráfica……2
Estrategia……..2
Información extra……..4

Introducción

Decidimos el tema de nuestro proyecto y el lenguaje en el que lo realizaríamos. Cada integrante del grupo pensó en un escenario posible, siendo estos:

- Una pecera en la que los peces deberían buscar la comida mientras esquivaban los obstáculos.
- Una charca en la que las ranas tendrían que llegar de una orilla a otra saltando por los nenúfares.
- Un laberinto en el que varios corredores buscarían la salida y marcarían el camino que recorren.

Escogimos el escenario del laberinto y decidimos que en el centro del mapa aparecieran 4 corredores, con un obstáculo en medio para obligarles a ir por distintos caminos. Cada corredor va marcando su camino con piedras (cada uno tiene su propio color) y si encuentra un camino sin salida, lo marca como tal para que los demás corredores no tomen ese camino y encuentren la salida más fácilmente.

Percepciones

Paredes, otros corredores que aparezcan en el laberinto.

Acciones

Andar hacia adelante, atrás, o los lados.

Objetivos

Salir del laberinto.

Entorno

Un mapa con paredes (laberinto).

Arquitectura Software

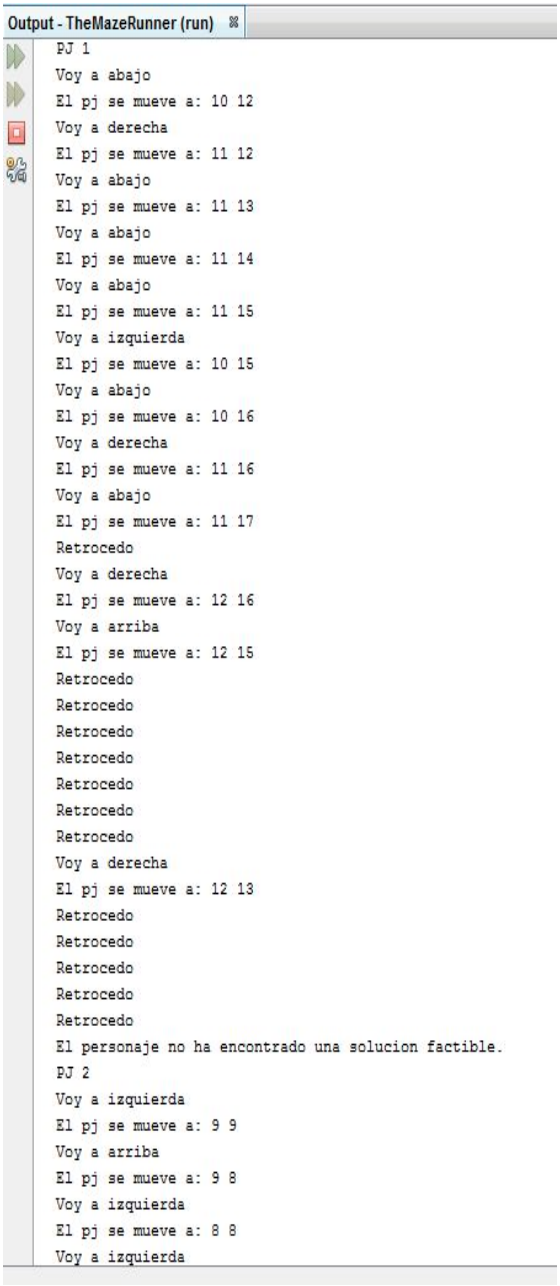
Arquitectura por capas, con tres capas. La más básica es la reactiva, ya que los corredores avanzarán según hayan obstáculos o no. La siguiente es basada en modelos que aprenden, porque si un camino no tiene salida y ya ha sido explorado, los demás corredores no tomarán ese camino. La última es la de objetivo, pues se quiere que todos logren salir del laberinto.

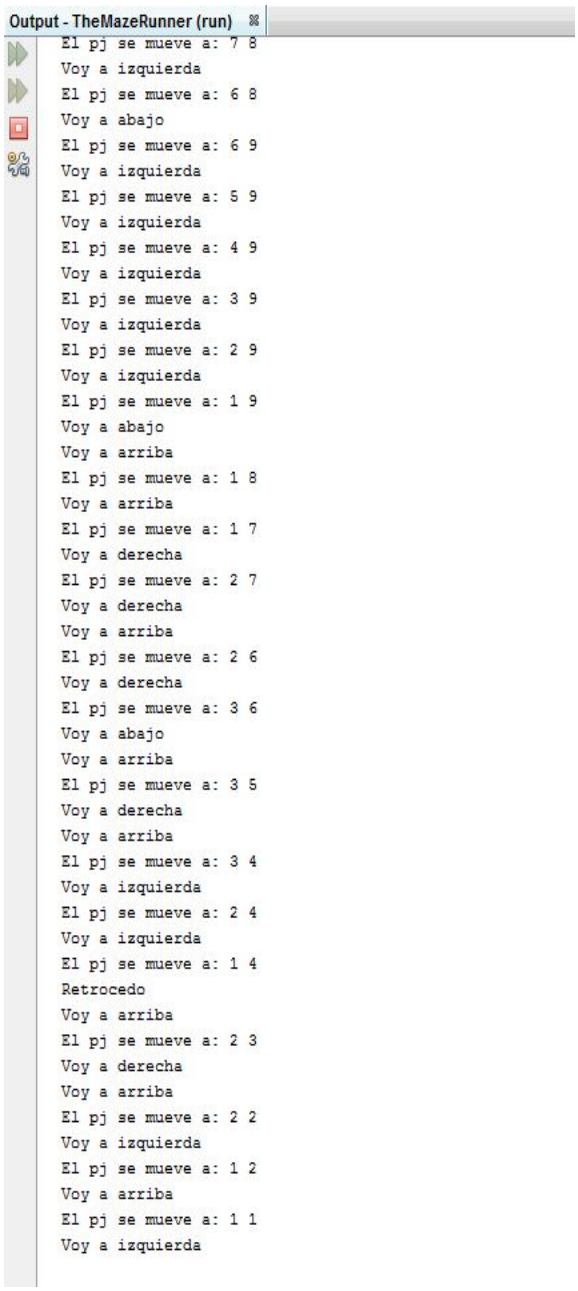
Interfaz Gráfica

Durante dos semanas trabajamos en el desarrollo de la interfaz gráfica. Usamos Java para escribir nuestro código y cada elemento de la matriz la representamos con un JLabel.
Hicimos nuestro laberinto, con distintas imágenes para cada cosa (suelo, obstáculo, salida…) y pusimos el código necesario para que si cambiábamos el tamaño del laberinto, los JLabel adaptaran su tamaño al de la matriz, de esta forma no se cortaba la matriz introducir valores muy grandes. En la primera semana los obstáculos aparecían de forma aleatoria.
En la segunda semana añadimos que a parte de aparecer aleatoriamente, los obstáculos se podían poner a mano, si clickeamos en una casilla con césped aparecerá un obstáculo y viceversa. Para esto usamos MouseListener.

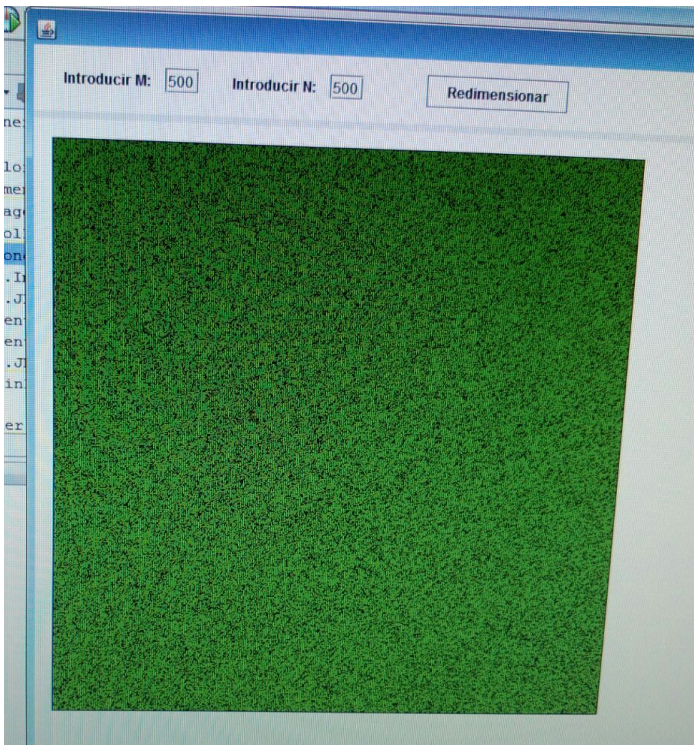
Estrategia

En la primera semana que dedicamos a la estrategia, escogimos cuál íbamos a usar. Decidimos usar la estrategia Hill Climbing o de escalada, según la cuál cada personaje escogería cada paso eligiendo la casilla vecina con menor distancia de Manhattan. Además, cada uno marcaba su camino y si retrocedía lo dejaba doblemente marcado para que los demás no fueran por caminos sin salida ya explorados. Esto lo conseguimos añadiendo la condición de que si una casilla está doblemente marcada se vería como un obstáculo y no se podría escoger.
La semana siguiente terminamos con la estrategia añadiendo la lista en la que cada personaje pondría su camino, así si se quedara en un camino sin salida este retrocedería sacando elementos de la lista hasta llegar a uno en el que pudiera tomar otra dirección.


Output - TheMazeRunner (run)
PJ 1
Voy a abajo
El pj se mueve a: 10 12
Voy a derecha
El pj se mueve a: 11 12
Voy a abajo
El pj se mueve a: 11 13
Voy a abajo
El pj se mueve a: 11 14
Voy a abajo
El pj se mueve a: 11 15
Voy a izquierda
El pj se mueve a: 10 15
Voy a abajo
El pj se mueve a: 10 16
Voy a derecha
El pj se mueve a: 11 16
Voy a abajo
El pj se mueve a: 11 17
Retrocedo
Voy a derecha
El pj se mueve a: 12 16
Voy a arriba
El pj se mueve a: 12 15
Retrocedo
Retrocedo
Retrocedo
Retrocedo
Retrocedo
Retrocedo
Retrocedo
Voy a derecha
El pj se mueve a: 12 13
Retrocedo
Retrocedo
Retrocedo
Retrocedo
Retrocedo
El personaje no ha encontrado una solucion factible.
PJ 2
Voy a izquierda
El pj se mueve a: 9 9
Voy a arriba
El pj se mueve a: 9 8
Voy a izquierda
El pj se mueve a: 8 8
Voy a izquierda


Output - TheMazeRunner (run)
El pj se mueve a: 7 8
Voy a izquierda
El pj se mueve a: 6 8
Voy a abajo
El pj se mueve a: 6 9
Voy a izquierda
El pj se mueve a: 5 9
Voy a izquierda
El pj se mueve a: 4 9
Voy a izquierda
El pj se mueve a: 3 9
Voy a izquierda
El pj se mueve a: 2 9
Voy a izquierda
El pj se mueve a: 1 9
Voy a abajo
Voy a arriba
El pj se mueve a: 1 8
Voy a arriba
El pj se mueve a: 1 7
Voy a derecha
El pj se mueve a: 2 7
Voy a derecha
Voy a arriba
El pj se mueve a: 2 6
Voy a derecha
El pj se mueve a: 3 6
Voy a abajo
Voy a arriba
El pj se mueve a: 3 5
Voy a derecha
Voy a arriba
El pj se mueve a: 3 4
Voy a izquierda
El pj se mueve a: 2 4
Voy a izquierda
El pj se mueve a: 1 4
Retrocedo
Voy a arriba
El pj se mueve a: 2 3
Voy a derecha
Voy a arriba
El pj se mueve a: 2 2
Voy a izquierda
El pj se mueve a: 1 2
Voy a arriba
El pj se mueve a: 1 1
Voy a izquierda

Superamos el limite a 500x500



Información extra

En nuestro código está comentado todo lo que vamos haciendo, explicando de una forma más exhaustiva los métodos que usamos, las funciones que hemos elegido, las variables que usaremos y todo lo que necesitamos para escribir el código de nuestro proyecto. El fichero en el que hemos puesto la parte del código más importante (parte gráfica, resolución de estrategia…) es Principal.java, en los demás se encontrarán clases que hemos usado en este fichero.