Laberinto

Jeremias Reyes

Leandro Navarrete

Hernan Escalante

Tecnicatura Superior en Programación.

Índice

Nuestro grupo va a desarrollar el juego del laberinto.

El juego va a consistir en la generación de un laberinto donde el usuario intentara resolverlo, también se mostrara los caminos posibles.

Resumen.

Nuestro grupo comenzó desarrollando un juego donde el objetivo era escapar de un enemigo a través de una matriz que podía variar de tamaño. Nos comenzamos a plantear la idea de que quizás en esta matriz pudiéramos generar un laberinto y después de un análisis decidimos comenzar con este nuevo proyecto “Laberinto”. Optamos por un algoritmo en particular y decidimos agregarle detalles como la demostración de la solución y que el jugador decida cuantas filas puede tener su laberinto.

Introducción

Mas allá del cambio de proyecto se decidió mantener la forma de trabajo que veníamos utilizando en la cual el alumno Jeremias Reyes es el Scrum master , Leandro Navarrete el Product owner y Hernan Escalante del equipo de desarrollo. Cada uno aportando su parte del código. La idea no fue concentrarse tanto en el aspecto visual si no mas en el algoritmo generador.

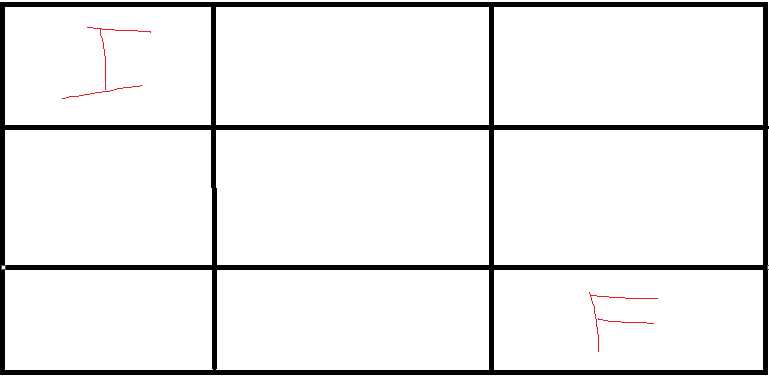
Método

Algoritmo generador del laberinto

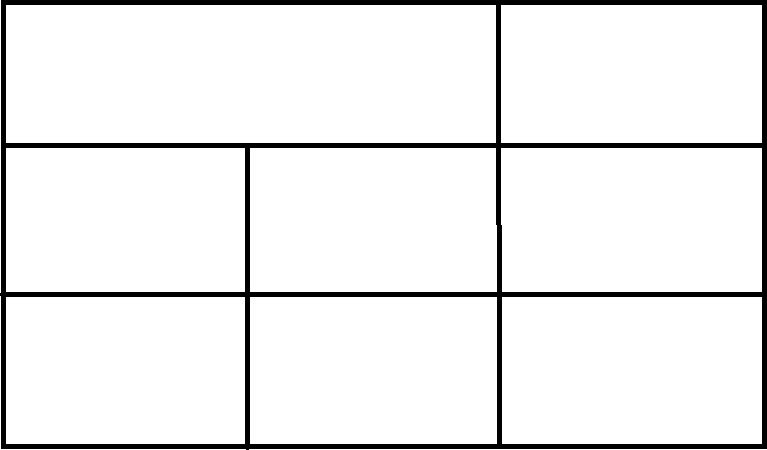
Es un algoritmo que funciona con iteraciones, arboles ternarios y matrices que contienen datos de tipo nodoArbol, este algoritmo genera un laberinto cuadrado de n\*n, con una entrada y una salida preestablecida, la entrada siempre es el principio de una matriz, y la salida siempre es el final de la matriz, este recorre todas las celdas de una matriz de nodos que pertenecen al árbol derribando "paredes" las cuales son representadas mediante los enlaces del árbol en el caso de que se llegue a un punto donde no hay salida el algoritmo recuerda los pasos que dio por lo que retrocederá hasta llegar a un punto donde si haya salida.

A continuación, para la explicación del algoritmo se utilizará un ejemplo para generar un laberinto de 3\*3.

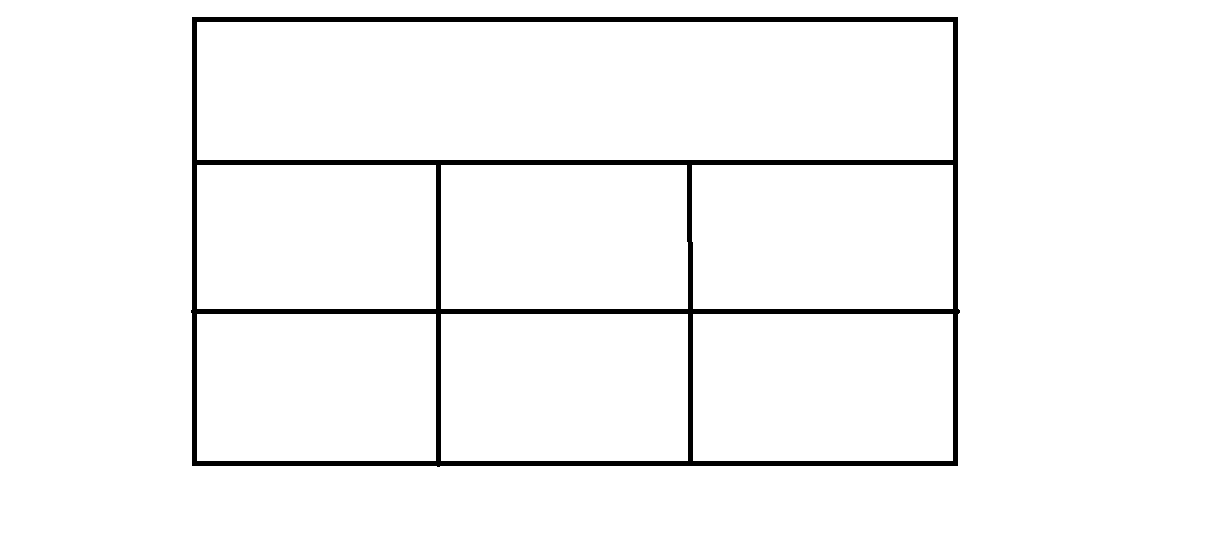
Matriz inicial (I=Inicio F=Fin)



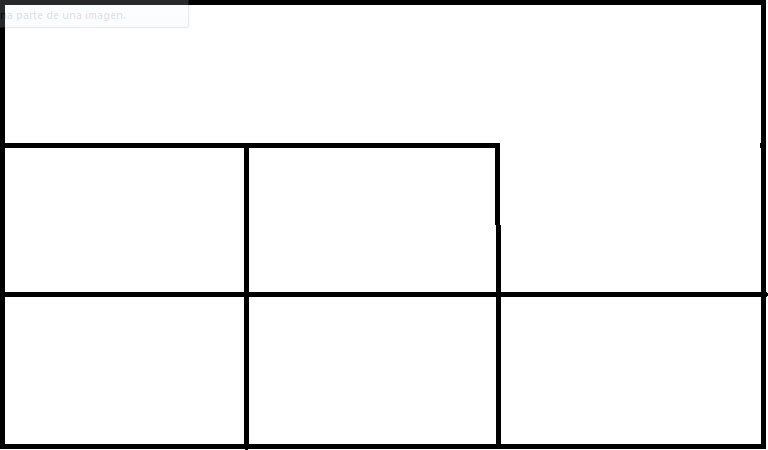
1. Desde el punto de salida solo se tiene dos posibles variantes, la pared derecha y la pared de abajo, se debe elegir una al azar en nuestro ejemplo se elegirá la pared de la derecha.
2. Una vez elegida la pared se avanza en dirección hacia la pared derribada se ve si las casillas adyacentes no se han visitado previamente, (en el caso de que si se haya visitado previamente significa que hay una pared por lo que se debe retroceder a la casilla donde se estaba previamente y así sucesivamente hasta encontrar una salida, también se debe marcar la casilla como visitada para no volver a pasar por ella.



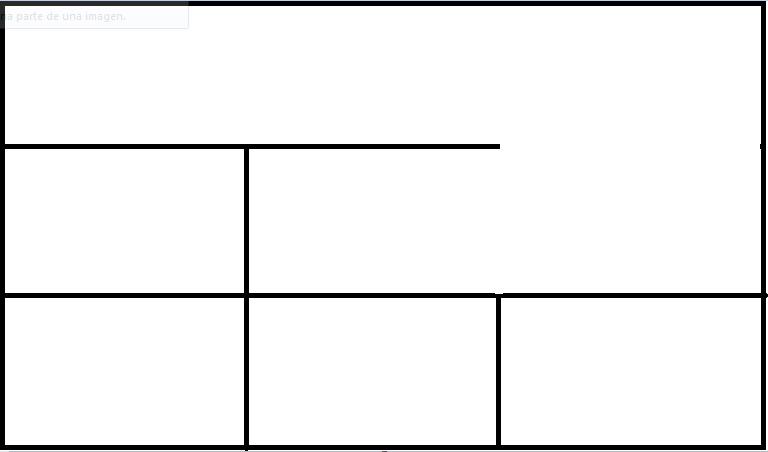
1. Nuevamente se repite el paso dos, ahora elegimos la pared derecha.



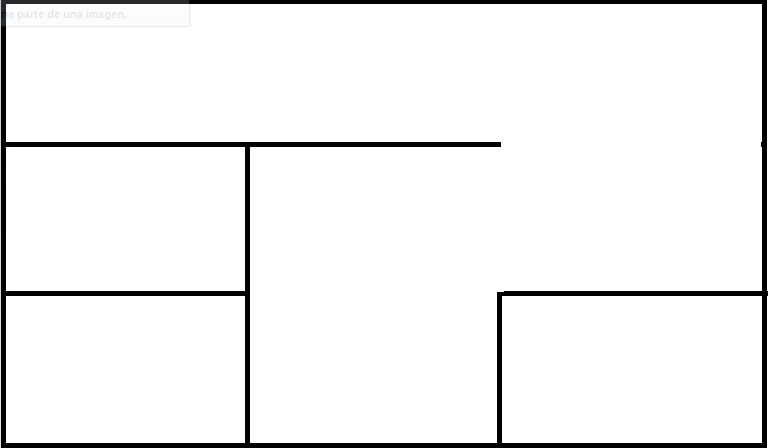
1. Repetiremos el paso 2, y elegiremos la pared de abajo.



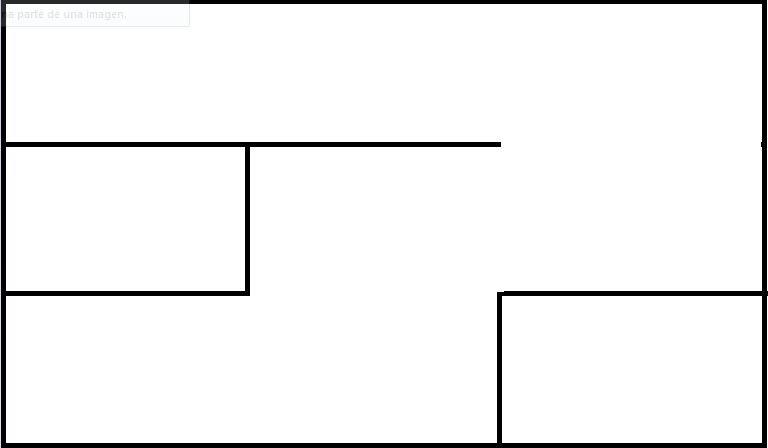
1. Repetiremos el paso 2 y elegiremos la pared de la izquierda.



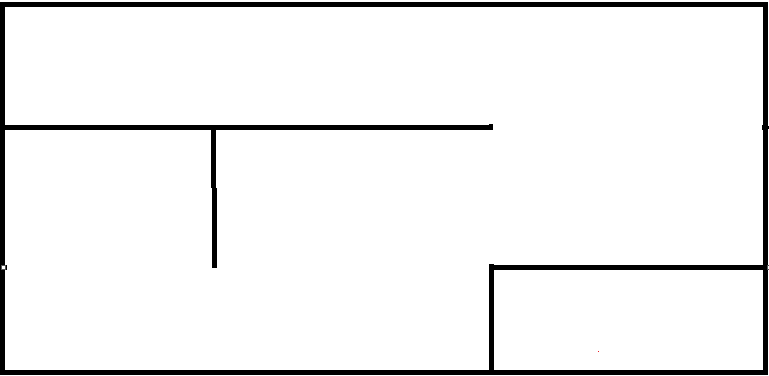
1. Repetiremos el paso 2 y elegiremos la pared de la abajo.



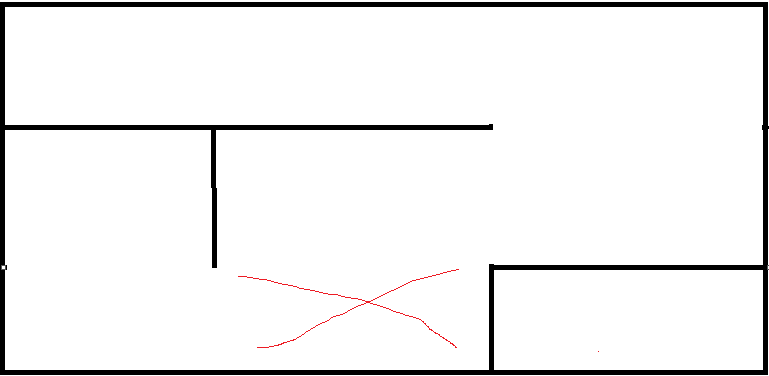
1. Repetiremos el paso 2 y elegiremos la pared de la izquierda.



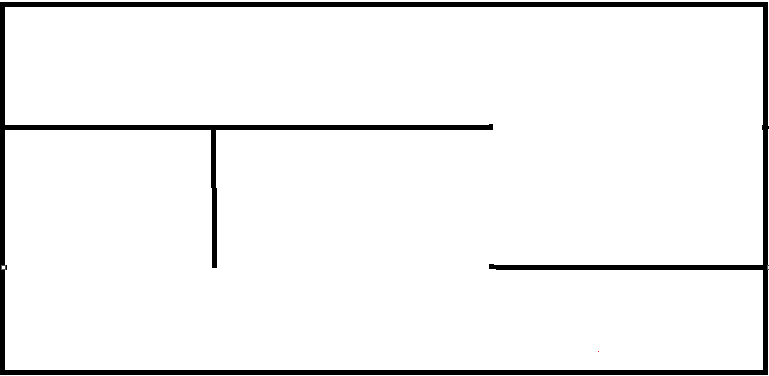
1. Repetiremos el paso 2 y elegiremos la pared de arriba.



1. En este punto no tenemos a donde ir, entonces retrocederemos hasta donde haya una salida (Se marca el punto con una cruz).

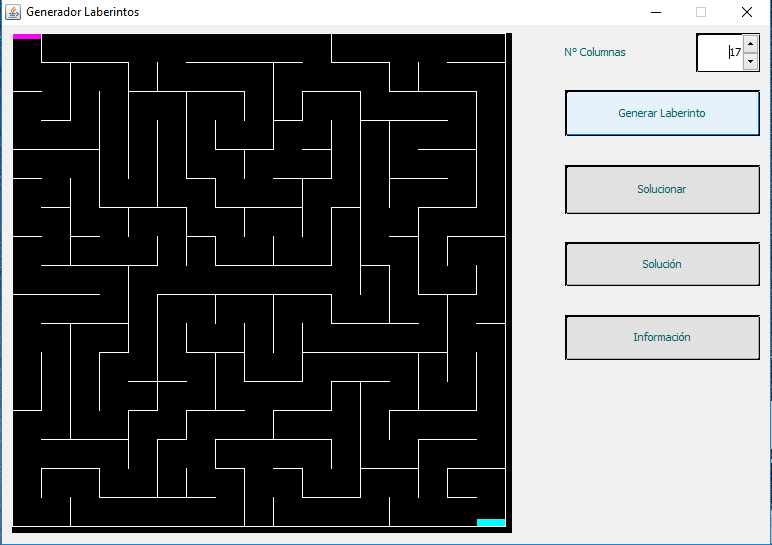


1. Repetiremos el paso 2 y elegiremos la pared de la derecha, y llegamos a la salida.



Resultados

Los resultados obtenidos en una fase donde el juego todavía se desarrollaba.



Se puede observar que dado el numero que un indique de columnas el laberinto ira incrementando su tamaño, también tenemos un botón que nos indicara la solución la opción solucionar para que el usuario intente resolverlo y un ultimo botón de información donde se mostrara un ventana con una instrucción.

En conclusión lo interesante de este juego es el generador del laberinto es curioso programar un juego con estas características ya que uno no se encuentra con muchas opciones a la hora de buscar información. Pero hay ejemplos curiosos donde uno puede darse una idea para seguir buscando y aprendiendo también distintos algoritmos que se pueden combinar como el de Recursión.

Fuentes utilizadas.

https://www.youtube.com/watch?v=v0ghwiepPwk&t=755s

https://serprogramador.es/programar-arboles-binarios-parte-1-introduccionclasesagregar-nodo/

http://elclubdejava.blogspot.com/

https://en.wikipedia.org/wiki/Maze\_generation\_algorithm

https://www.youtube.com/watch?v=HyK\_Q5rrcr4

https://www.youtube.com/watch?v=D8UgRyRnvXU

<https://en.wikipedia.org/wiki/Backtracking>

Proyecto Final metodología.

Alumnos: Navarrete Leandro

Jeremias Reyes

Hernan Escalante