

HERNAN JAVIER
AGUILAR CRUZ

ALEJANDRO
MESA GOMEZ

LABERINTO

1. INTRODUCCIÓN

Mediante la necesidad de elaborar un algoritmo funcional para generar el dichoso laberinto, se tuvieron en cuenta diferentes maneras, pero la que se adaptó más a nuestra manera de pensar y así mismo a nuestras necesidades fue el uso de backtracking.

Esta técnica de programación nos sirvió para desarrollar de una manera más práctica nuestro proyecto, pudiendo así aplicar conocimientos relacionados al curso. De igual manera, se hizo un trabajo en conjunto con la librería “p5” de Daniel Shiffman (más conocido por el canal de YouTube de “The Coding Train”), que facilitó el entorno gráfico y el desarrollo de la aplicación en general.

Además, para poder complementar nuestro trabajo, se utilizó como herramienta Atom, (mediante su aplicativo de jsdoc) para poder realizar el manual técnico de la aplicación. Por supuesto, para poder consolidar nuestro proyecto como propio, quisimos subirlo a un repositorio de GitHub desde el cual se podrá acceder al mismo.

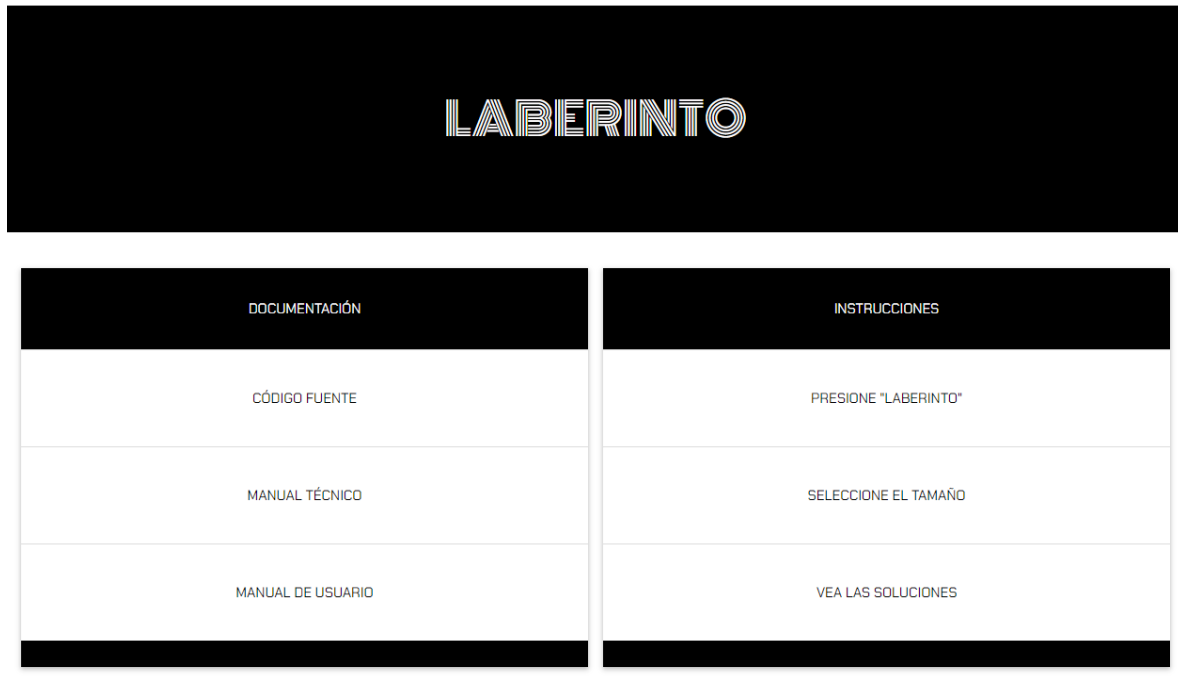
2. FUNCIONALIDADES

En el trabajo, se utilizan diferentes scripts para html que facilitan la parte visual (como la librería “p5” mencionada anteriormente), así como a la parte funcional (por ejemplo, las screenshots con la librería “html2canvas”). Esto, para hacer que el ambiente de la aplicación sea una más amigable con el usuario; permitiendo así un buen entendimiento y una mejor estética.

3. ASPECTOS FUNDAMENTALES

Para realizar la aplicación, se tuvo en cuenta 2 páginas en total. Estas son: “index.html” y “laberinto.html”. Con ellas, se trabajaría tanto interfaz de inicio como interfaz de juego. Se optó por tener un menú, puesto que se necesita de una pequeña introducción al uso del programa (instrucciones para su entendimiento), y también por una pequeña bibliografía en la que se tuviera acceso a la documentación del programa.

3.1. INTERFAZ DE INICIO



Como interfaz de inicio, el programa cuenta con el título del proyecto, y dos contenedores en los que se tiene la documentación, y las instrucciones. En el caso de la documentación, cada uno de sus ítems se redirigirá a la pestaña de la cual se habla.

3.1.1. DOCUMENTACIÓN

En el código fuente, se trasladará al repositorio de GitHub en el que está montado el proyecto. El manual técnico, será una pestaña donde se trate de la documentación, y finalmente el manual de usuario (el que se está leyendo en este momento). Cada uno de ellos, se mostrará de la siguiente manera:

3.1.1.1 CÓDIGO FUENTE

cosmolejo update	
Js	update
lib	update
out	update
README.md	Initial commit
index.html	update
laberinto.html	update
README.md	
<h1>MazeRunner</h1>	

Estará el repositorio de GitHub en el que se encontrará todo el proyecto.

3.1.1.2 MANUAL TÉCNICO

[Home](#)

Classes

Cell

Global

Colores
draw Toma los datos de la generación hecha en setup para graficarlos.
Estados
getCell Permite conseguir la posición actual.
getCellUnderCursor Ayuda a indicar la celda para colocar la posición inicial y final.
getRndNeighbor selecciona un vecino al azar de la celda enviada
highlight colorea la celda según el color enviado
infoText Escribe en la parte superior las indicaciones.
isFenced Permite saber si no existe un camino válido, para así devolverse.
keyPressed Ayuda a marcar la posición elegida con el cursor.
mouseClicked Ayuda a saber cuál fue la posición en el lugar de inicio fin.
removeWall Función utilizada para remover muros en la creación del algoritmo.
setup En esta función se genera el laberinto.
show permite visualizar la celda dentro del canvas
showRoute Muestra el recorrido del laberinto paso por paso.
stroke crea un rectángulo dentro de la celda, del color enviado
Tamaños

Estará subida la documentación de la clase Cell, y de las funciones hechas en sketch.

3.1.1.3 MANUAL DE USUARIO

HERNAN JAVIER
AGUILAR CRUZ

ALEJANDRO
MESA GOMEZ

3.1.2. INSTRUCCIONES

En el caso las instrucciones, son sólo informativas. En ellas, se dan las pautas para poder ver el funcionamiento del laberinto. Como dice la primera, para hacer funcionar el programa se deberá presionar “LABERINTO”. Esto pasa al ser presionado:

TAMAÑO

18x18

20x20

22x22

24x24

26x26

✕

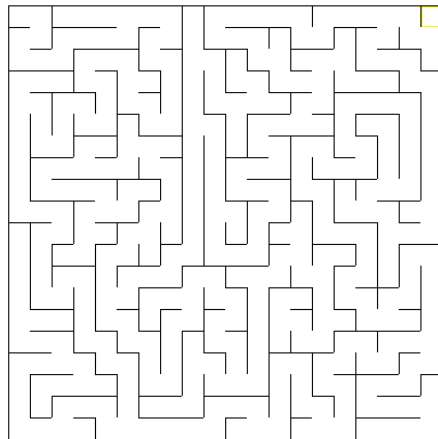
Aquí se seleccionará el tamaño para poder ver el laberinto.

3.2. INTERFAZ DE JUEGO

Una vez se ha seleccionado el tamaño del tablero a generar, esta será la pestaña que nos aparece, la cual corresponde a la interfaz de juego:



SELECCIONE EL INICIO



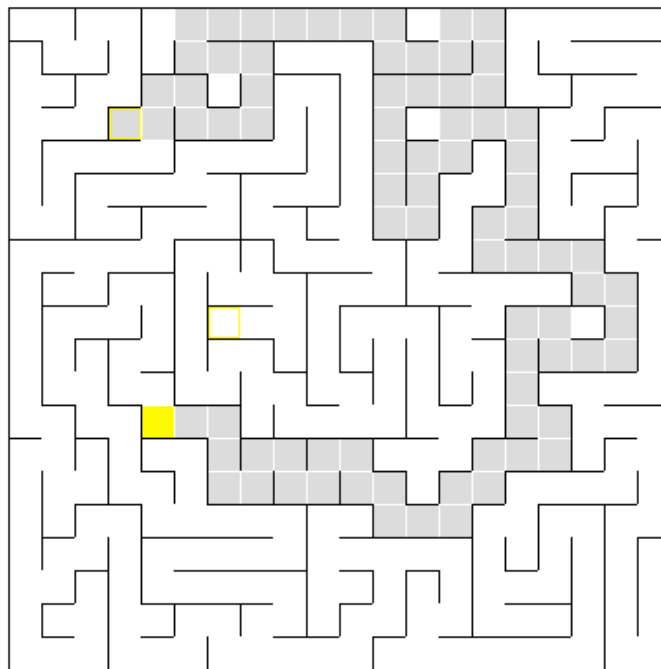
En la parte superior, podemos ver que dice “LABERINTO” si lo presionamos, tendrá la opción de volver al inicio del programa. También, debajo de esta, podemos ver un texto que dice “SELECCIONE EL INICIO”.

Este nos ayudará como indicador para saber lo que tenemos que hacer, sus etapas iniciales son:

SELECCIONE EL INICIO

SELECCIONE EL FIN

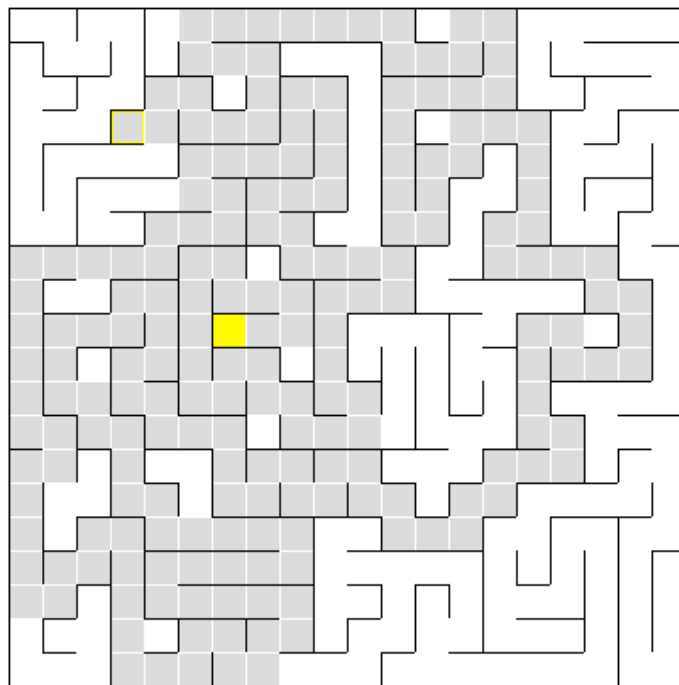
Cada una de ellas, también refleja un proceso en el laberinto. El laberinto, al tener una posición de inicio y una de fin a buscar, recorrerá caminos celda por celda hasta encontrar la solución.



En este caso, la etapa del laberinto pasa a ser:

RESOLVIENDO

Una vez encontrada la solución a buscar, se terminará el recorrido en el laberinto de esta manera:



Y se notificará así:

SOLUCIÓN ENCONTRADA

4. EXCEPCIONES

- ¿La posición de inicio ya ha sido elegida?
- ¿La posición de fin ya ha sido elegida?
- ¿La posición de inicio es válida?
- ¿La posición de fin es válida?
- ¿Es un camino cerrado?
- ¿La siguiente posición es válida?
- ¿Ya se encontró la solución?
- ¿La posición actual es el fin?