

# Proyecto 2: Generador de Laberintos

**Miércoles 25 de Abril**

## I. DESCRIPCIÓN GENERAL

Hay varios algoritmos para generar un laberinto. Para este proyecto usaremos un algoritmo descrito por David Matuszek<sup>1</sup>. Los laberintos solamente serán generados, grabados o leídos para ser desplegados, pero no serán resueltos. La interfaz debe ser gráfica. Toda la programación debe realizarse en C sobre Linux. No se pueden cambiar las especificaciones de este documento.

## II. INTERACCIÓN GRÁFICA

La interacción con el usuario se hará por medio de interfaces gráficas que deben ser desarrolladas con GTK y Glade. Además se debe usar Cairo para el despliegue del laberinto. La calidad estética de la salida será muy tomada en cuenta.

**Trabajo extra opcional 1:** hacer que el piso y las paredes del laberinto tengan alguna textura apropiada.

El programa tendrá la posibilidad de grabar archivos con los datos particulares de cada laberinto generado, para que puedan ser cargados de nuevo. El formato de este archivo queda a discreción de cada grupo de trabajo. Se espera que, el día de la revisión, cada grupo cuente con bastantes archivos de pruebas para mostrar las capacidades de su proyecto.

Habrán botones para las siguientes funciones:

- **Generar:** se abre una subventana que pedirá la cantidad de filas y columnas del laberinto (pueden ser laberintos de hasta  $2048 \times 2048$ ). Estos son campos que el usuario debe ingresar. El programa impedirá el ingreso de cualquier entrada ilegal.
- **Leer:** se carga y se despliega un laberinto previamente grabado.
- **Grabar:** se graba el laberinto actual en un archivo. Este botón estará deshabilitado en general, pero se activa cuando se genera un nuevo laberinto.
- **Resolver:** este botón sólo dará un mensaje que diga “No implementado”.
- **Salir:** se termina la ejecución del programa.

<sup>1</sup>Cualquier uso de otro algoritmo o de software tomado de Internet para generar laberintos se considerará fraude académico y será severamente castigado

## III. NAVEGACIÓN

Independientemente del tamaño solicitado del laberinto, este será desplegado completamente en la interfaz (puede que sólo se vea una mancha negra...).

Sin alterar el tamaño del área de la interfaz dedicada al despliegue del laberinto, el usuario podrá hacer acercamientos o alejamientos del laberinto (**Zoom in/ Zoom out**). Se puede llegar hasta a ver una sola celda del laberinto o hasta un despliegue total del laberinto. También podrá mover el área visible del laberinto hacia arriba, abajo, izquierda o derecha.

Toda esta interacción debe ser controlada con el *mouse*. Investigue las capacidades de GTK y de Cairo al respecto. Obviamente, esta funcionalidad sólo tiene sentido si hay un laberinto desplegado.

**Trabajo extra opcional 2:** implemente adicionalmente laberintos donde la casilla básica es un hexágono en vez de un cuadrado.

## IV. REQUISITOS INDISPENSABLES

La ausencia de uno solo de los siguientes requisitos vuelve al proyecto “no revisable” y recibe un 0 de calificación inmediata:

- Todo el código debe estar escrito en C (no C++).
- El proyecto debe compilar y ejecutar en Linux.
- Todas las interfaces deben ser gráficas.
- Se debe usar GTK, Glade y Cairo.
- No debe dar “Segmentation Fault” bajo ninguna circunstancia.

**Trabajo extra opcional 3:** hacer la demostración en una máquina que levante Linux de manera real (puede ser dual), es decir no usar máquinas virtuales.

## V. FECHA DE ENTREGA

Revisiones a las 11:30am el **Miércoles 25 de Abril**. Mande además un .tgz con todo lo necesario (fuentes, makefile, readme, etc.) a [torresrojas.cursos@gmail.com](mailto:torresrojas.cursos@gmail.com). Ponga como subject: A.A. - Proyecto 2 - Fulano - Mengano, donde Fulano y Mengano son los 2 miembros del grupo.