



por los **ítems recogidos** antes de que se acabe el tiempo.

## Menú Principal

### 1. Cargar Laberinto desde Archivo CSV

- Leer un archivo CSV que describe los **escenarios**: sus descripciones, decisiones (con direcciones posibles), ítems disponibles (nombre, peso, valor) y conexiones.
- Construir el **grafo (explícito)** a partir de esa información.

Archivo de ejemplo: [graphquest.csv](#)

### 2. Iniciar Partida

- Comienza el juego desde el escenario inicial. Se muestra el estado del jugador y el menú de opciones correspondientes.

## Menú del Juego

Cada vez que se entra a un nuevo escenario o se realiza una acción, se muestra el siguiente menú:

### Estado Actual

- Descripción del escenario actual.
- Lista de ítems disponibles en este escenario (con nombre, peso y valor).
- Tiempo restante (puedes inicializarlo en 10 o el tiempo que estimes conveniente).
- Inventario del jugador (ítems recogidos, peso total y puntaje acumulado).
- Acciones posibles desde este escenario: direcciones disponibles (arriba, abajo, izquierda, derecha).

### Opciones del Jugador

#### 1. Recoger Ítem(s)

- El jugador puede seleccionar uno o más ítems del escenario para agregarlos a su inventario. Se descuenta 1 de tiempo.

#### 2. Descartar Ítem(s)

- El jugador puede eliminar ítems de su inventario para reducir peso y moverse más rápido. Se descuenta 1 de tiempo.

### 3. Avanzar en una Dirección

- El jugador elige una dirección válida.
- Se actualiza el escenario actual, el inventario se conserva, y se descuenta el tiempo usado según el peso total transportado:

$$T = \left\lceil \frac{\text{Peso total del inventario} + 1}{10} \right\rceil$$

- Si se alcanza el escenario final, se muestran los elementos del inventario y el **puntaje final**.
- Si el **tiempo se agota**, se muestra un mensaje de derrota.

### 4. Reiniciar Partida

- Se reinicia el juego desde el escenario inicial, con inventario vacío y tiempo completo.

### 5. Salir del Juego

- Finaliza la partida y cierra la aplicación.

## Opciones adicionales (en caso de trabajar en pareja)

### Modo multijugador colaborativo por turnos (para 2 jugadores):

- Antes de cada turno se muestra el número (o nombre) del jugador correspondiente.
- Se muestra el estado del jugador actual.
- Los jugadores pueden realizar 1 o 2 acciones por turno (acciones: recoger ítem, descartar ítem o avanzar).
- **Conflictos de ítems:** Si un jugador toma un ítem, el otro ya no lo verá (ítems son únicos por escenario).
- **Fin de partida:**
  - Termina cuando **ambos jugadores lleguen al escenario final o se queden sin tiempo**.

- **Si ambos jugadores llegan a la salida** se muestran los items de ambos inventarios y el puntaje total obtenido.

## ¿Cómo empezar?

En este repl hay código que lee el csv de ejemplo e imprime todos los campos de los escenarios.

## Consideraciones

1. La tarea se puede realizar de manera **individual o en pareja**.
2. Utilicen librerías estándar de C (ver aquí).
3. Deben subir su tarea a un repositorio github o compartir su proyecto en repl.it. El link lo copian en el aula virtual.
4. Además deben anexar un archivo README.md (ejemplo) indicando:
  - a. cómo compilar y ejecutar la tarea;
  - b. las opciones que funcionan correctamente y las que no lo hacen indicando posibles causas;
  - c. un ejemplo mostrando como interactuar con la aplicación;
  - d. detalle de las contribuciones realizada por cada integrante (sin son dos).
5. **Copia será calificada con puntaje 0.**
6. **Tareas que no compilen serán calificadas con nota 0.**

## Rúbrica de evaluación

### Tarea 3 -old