Laboratorio 2:

Profesor: Viktor Tapia
Ayudante de cátedra: Juan Pablo Varas y Martín Rodriguez
Ayudante de Tarea: Javiera Cárdenas y Nicolás Toro

Septiembre 2023

1 Reglas Generales

Para la siguiente tarea se debe realizar un código programado en lenguaje C. Se exigirá que los archivos se presenten de la forma más limpia y legible posible. Deberá incluir un archivo README con las instrucciones de uso de sus programas junto a cualquier indicación que sea necesaria, y un archivo MAKE para poder ejecutar el programa.

2 Tarea

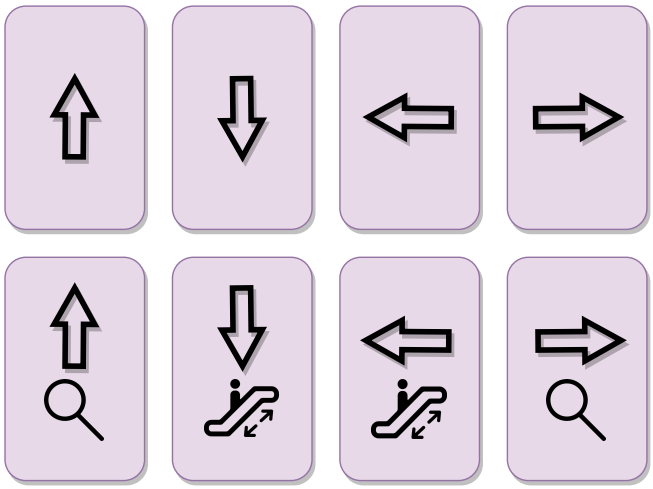
Se solicita recrear el juego de mesa llamado "Magic Maze", para esto se les pide que desarrollen el juego y además de que un participante puede jugar a través de la consola, se creen 3 jugadores extra para que el juego quede de 4 jugadores en total. Los jugadores extras deben ser creados con la función fork().

2.1 Conceptos básicos

Se especificarán los conceptos básicos, cualquier cambio a estos debe ser especificado en el README que entreguen junto con el programa:

1. Se tienen 8 cartas en total, en donde cada una representa una dirección, además 4 de estas 8 cartas contienen una acción que se puede realizar:
   (a) Buscar, representado por una B.
   (b) Escaleras, representado por una E.
   (c) Derecha, Izquierda, Arriba, Abajo, representando la dirección en la que se puede mover el jugador.

A modo de ejemplificar lo dicho anteriormente se muestran las cartas en la siguiente imagen:



2. El concepto clave del juego es la recolección de 4 tesoros, 1 por cada jugador y luego volver al laberinto inicial en una cantidad limitada de turnos para poder terminar el juego. En la mesa tendremos un tablero inicial donde los 4 jugadores empezaran y tendrán que ir descubriendo las otras partes del tablero usando las acciones de "Buscar" en las casillas respectivas, al momento de realizar la acción se debe agregar un laberinto que coincida con la dirección en la que apunta el jugador, donde existe una probabilidad de encontrar el tesoro o uno de los 4 tipos de eventos:

   (a) b_c, representando casillas donde existen cámaras, en el caso de existir 2 casillas con cámara activas, no podrán obtener mas turnos. Para eliminar las cámaras el jugador debe pasar sobre ellas.

   (b) b_t, representado casillas para obtener 5 turnos.

   (c) b_n, representado casillas para perder 3 turnos.

   (d) b_{tpN}, representado casillas para poder teletransportarse a otra casilla del mismo tipo, y N representando al jugador respectivo.

   Cabe recalcar que cada jugador **solo tiene 1 tesoro a recolectar** y que estos no pueden encontrarse dentro de la misma casilla.

2.2 El tablero

El juego consta de 9 laberintos distintos, 1 siendo el inicial el cual tiene 4 salidas en las 4 direcciones generales y 4 tableros en donde las salidas tienen forma de "L" y 4 tableros donde las salidas tienen forma de "T". Cada tablero tendrá un formato de 5x5 y serán entregados en un *archivo.txt* con la tarea donde:

1. 0, representa las casillas donde se puede mover el jugador.
2. $/$, representa las casillas donde existen muros que los jugadores no pueden atravesar
3. J_N, representa la casilla donde esta el jugador/bots.
4. b_c, b_t, b_n, b_{tpN}, representando las casillas mencionadas anteriormente, donde cada una tendrá 25% de probabilidad de aparecer en una casilla del laberinto. Estas casillas deben ser agregadas por **ustedes** en cualquier casilla al momento de descubrir un laberinto.

5. B, representando las casillas mencionadas anteriormente.
6. E, representando las casillas mencionadas anteriormente.
7. 1, 2, 3, 4, representando el tesoro a encontrar por cada jugador.

Un ejemplo de tablero puede ser:

$$\begin{bmatrix} / & / & / & / & / \\ / & \mathrm{J}_1 & 0 & \mathrm{J}_2 & / \\ B & 0 & 0 & 0 & B \\ / & \mathrm{J}_3 & 0 & \mathrm{J}_4 & / \\ / & / & B & E & 1 \end{bmatrix}$$

Tablero en forma de "T"

2.3 Inicio de la partida

- Se desordena todo el mazo de cartas.
- Se desordena todo el mazo de laberintos.
- Se reparte a cada jugador 2 cartas.
- Se colocan los jugadores en el tablero inicial.

2.4 Desarrollo de la Partida

La partida dura solo una ronda con 15 turnos. Cada turno se desarrolla de la siguiente manera:

1. El jugador elige una de las cartas de su mano por consola para poder realizar una acción o movimiento, el movimiento se puede realizar en la dirección escogida una cantidad ilimitada de veces si no esta chocando contra un muro, jugador, escalera o limite del laberinto, el resto de los jugadores deberán elegir la mejor carta a jugar dependiendo de su posición.
2. En el momento que algún jugador se encuentre en una casilla "B", debe realizar la acción de buscar para añadir el siguiente laberinto, usando su turno en el proceso.
3. Lo mismo sucede en el caso de que un jugador obtenga un tesoro o suba escaleras.
4. Para obtener las bonificaciones de las casillas especiales basta con que el jugador(usted) camine por sobre ellas.
5. El turno termina cuando los 4 jugadores hayan realizado una acción.

2.5 Fin de la Partida

La partida puede finalizar de dos formas:

1. Los jugadores logran escapar con los 4 tesoros.
2. Los jugadores se quedan sin turnos.

3 Jugadores y procesos

El juego se compone de 4 jugadores (siendo usted uno de ellos). Cada jugador debe ser un proceso distinto, por lo que en total su programa estará ejecutando 5 procesos (un padre y 4 hijos). La acción que tendrá cada jugador, será escoger la acción o movimiento a realizar. El orden sera el jugador seguido por los 3 bots, el proceso padre debe indicar la accion realizada a los demas, actualizar el mapa ,notificar la victoria/derrota. Recuerde que para finalizar el programa, primero deben finalizar los procesos hijos, y luego el proceso padre.

3.1 Visualización de la Consola

Queda a su creatividad diseñar el formato de visualización del juego. Como mínimo se requiere que se muestren los siguientes elementos:

- Mano del Jugador: se deben poder saber las cartas del jugador principal en consola y debe permitir escoger la carta a jugar en su turno. El resto de los jugadores deben ser automáticos bajo las reglas indicadas anteriormente.
- Turno Actual: Debe estar indicado por pantalla el jugador que esté jugando. Un pequeño mensaje bastará.
- Debe mostrar la posición inicial y la posición final de cada jugador cuando comiencen y terminen su turno.
- Debe mostrar cuando el juego termine, en cualquiera de los 2 casos.

4 Presentación Aleatoria

Para cada tarea, se seleccionarán grupos al azar para presentar su tarea frente a ayudantes y eventualmente profesor, recibiendo una ponderación del 80% y 20% entre tarea y presentación respectivamente. Si su grupo presentó en una tarea, no volverá a salir nuevamente. Se comunicará días antes que grupos presentarán.

5 README

Debe contener como mínimo:

- Nombre, Rol y Paralelo de los integrantes.
- Especificación de los nombres de los archivos.
- Instrucciones generales de compilación y uso.

6 Consideraciones Generales

- Se deberá trabajar de a pares. Se deberá entregar en Github a mas tardar el día 30 de Septiembre de 2023 a las 23:59 horas. Se descontarán 10 puntos por cada hora o fracción de atraso. Las copias serán evaluadas con nota 0 en el promedio de las tareas.
- La tarea debe ser hecha en el lenguaje C. Se asume que usted sabe programar en este lenguaje, ha tenido vivencias con el, o que aprende con rapidez.
- Pueden crear todas las funciones auxiliares que deseen, siempre y cuando estén debidamente comentadas.
- Las tareas serán ejecutadas en **Linux**, cualquier tarea que no se pueda ejecutar en dicho sistema operativo, partirá de nota máxima 60. formato **TAREA2_ROL1_ROL2**.
- Las preguntas deben ser hechas por Aula o por Discord. De esta forma los demás grupos pueden beneficiarse en base a la pregunta.
- Si no se entrega README o MAKE, o si su programa no funciona, la nota es 0 hasta la recorrección.

- Se descontarán 50 puntos por:
  - Mala implementación del Makefile.
  - No respetar el formato de entrega.