Universidad Nacional Autónoma de México
Semestre 2026-1
Fundamentos de programación
Juego bricks breaker
Profesora: Adara Mercado Martínez

Descripción General del Problema

El objetivo de esta actividad es que el estudiante implemente una versión simplificada del juego *Brick Breaker* utilizando únicamente salida por consola mediante la función `printf`. El jugador controla una plataforma horizontal que se mueve a la izquierda y derecha para rebotar una pelota, la cual debe romper los bloques ubicados en la parte superior del área de juego.

El área de juego debe representarse usando caracteres ASCII. La pelota debe moverse automáticamente dentro del entorno, rebotando en paredes, plataforma y bloques. El juego termina cuando la pelota cae fuera del área de juego o cuando todos los bloques han sido eliminados.

El programa debe ejecutarse completamente en consola, sin usar librerías gráficas ni de posicionamiento especial más allá de las permitidas por la plataforma de programación.

Requerimientos del Programa

- Implementar un menú de inicio con al menos tres opciones: Iniciar juego, Instrucciones y Salir.
- Representar con caracteres ASCII la plataforma, la pelota, las paredes y los bloques.
- Implementar movimiento automático de la pelota.
- Implementar rebote de la pelota en:
  - paredes laterales,
  - pared superior,
  - superficie de la plataforma,
  - bloques.
- Finalizar el juego adecuadamente con un mensaje cuando el jugador gana o pierde.
- Utilizar colores en la consola cuando sea posible (por ejemplo, ANSI escape codes).
- Incluir una función de salida limpia del programa.

Análisis del Problema

Para implementar este juego en consola, se consideran los siguientes elementos fundamentales:

- Un espacio rectangular que simula la pantalla.
- Una matriz o estructura que almacene la posición de los bloques.
- Variables para la posición de la plataforma y la pelota.
- Variables para la dirección de la pelota en sus dos ejes.
- Detección de colisiones entre pelota y diferentes elementos.
- Un ciclo principal del juego que:
  1. borre la pantalla,

2. imprima el estado actualizado,
3. procese la entrada del jugador,
4. actualice las posiciones,
5. detecte colisiones,
6. verifique condiciones de fin.

Diseño de la Solución

El diseño debe incluir:

- Diagrama de flujo general del ciclo del juego.
- Diseño de estructuras de datos (arreglos y variables).
- Descripción del algoritmo del movimiento de la pelota.
- Descripción del algoritmo de rebote y colisión.
- Diseño del menú de inicio.

Pseudocódigo del Rebote de la Pelota

A continuación se incluye un pseudocódigo en estilo PSeInt para el manejo del rebote de la pelota:

```
Proceso RebotePelota
Si pelota.x <= 0 O pelota.x >= anchoPantalla Entonces
dx <- dx * -1
FinSi


Si pelota.y <= 0 Entonces
dy <- dy * -1
FinSi


Si pelota.y == posicionPlataforma.y Y
pelota.x >= posicionPlataforma.x Y
pelota.x <= posicionPlataforma.x + largoPlataforma Entonces
dy <- dy * -1
FinSi


Para cada bloque en bloques
Si bloque.activo Y pelota.x == bloque.x Y pelota.y == bloque.y Entonces
bloque.activo <- Falso
dy <- dy * -1
FinSi
FinPara
FinProceso
```

Rúbrica de Evaluación

La evaluación se realizará considerando los siguientes criterios:

1. Descripción del problema (20%)

- Explica claramente qué es el juego.
- Detalla el objetivo del jugador.
- Describe los elementos principales.
- Identifica las restricciones del entorno (solo consola, ASCII, `printf`).

2. Análisis del problema (20%)

- Identifica las variables necesarias.
- Identifica estructuras de datos.
- Reconoce los eventos principales del juego.

3. Diseño (20%)

- Incluye diagramas o explicaciones claras.
- Explica el algoritmo del movimiento y rebote.
- Explica la estructura del menú.

4. Implementación (40%)

- El programa compila y ejecuta correctamente.
- El juego es funcional y jugable.
- Existe un menú de entrada.
- Existe un mensaje de salida.
- Usa colores correctamente.
- Permite interacción fluida del usuario.