

## Descripción del Problema

Realizar una simulación para una version simplificada y modificada del juego de Serpientes y Escaleras [1]  
[2] con las siguientes características.

### 1. Características del tablero

- 1.1. El tablero debe contener 30 casillas (tiles)
- 1.2. El tablero debe contener 3 casillas de tipo serpiente indicadas con la letra S (snakes)
- 1.3. El tablero debe contener 3 casillas de tipo escalera indicadas con la letra L (ladders)
- 1.4. Las casillas restantes deberán identificarse como casillas normales con la letra N
- 1.5. Cada casilla serpiente deberá retroceder al jugador 3 casillas (penalty)
- 1.6. Cada casilla escalera deberá avanzar 3 casillas adicionales (reward)
- 1.7. Cada casilla se identifica con un número natural a partir del 1

### 2. Características de los jugadores

- 2.1. El juego debe soportar 2 jugadores
- 2.2. Cada jugador se identifica con un número natural a partir del 1
- 2.3. Al inicio del juego todos los jugadores se encuentran en la casilla 1
- 2.4. El jugador 1 comienza el primer turno del juego

### 3. Características del juego

- 3.1. El juego se compone de una sucesión de turnos
- 3.2. Cada turno se podrá ejecutar en línea de comandos alguna de las 2 posibles acciones siguientes:
  - 3.2.1. Introducir la letra "C" (continue) para continuar con el juego con el siguiente turno
  - 3.2.2. Introducir la letra "E" (end) para terminar el juego
  - 3.2.3. Solamente antes del primer turno se debe imprimir el siguiente menú de instrucciones:  
**Press C to continue next turn, or E to end the game:**
  - 3.2.4. Si la tecla introducida no es alguna de "C" o "E" se debe imprimir el siguiente mensaje:  
**Invalid option, please press C to continue next turn or E to end the game**

#### 3.3. Ejecución del turno

- 3.3.1. Durante cada turno se debe imprimir en consola la siguiente información correspondiente al turno.
- 3.3.2. El número del turno (el primer turno se indica con el número 1)
- 3.3.3. El número del jugador con el turno actual
- 3.3.4. El número de casilla correspondiente a la posición actual del jugador
- 3.3.5. El número obtenido al simular un dado convencional de 6 caras
- 3.3.6. El tipo de casilla a la que el jugador debería moverse después de tirar el dado
  - 3.3.6.1. N para casillas normales
  - 3.3.6.2. S para casillas serpiente (Snake)
  - 3.3.6.3. L para casillas escalera (Ladder)
- 3.3.7. La casilla final a la que el jugador debe moverse
  - 3.3.7.1. Si la casilla es Normal, entonces indicar la posición final directamente
  - 3.3.7.2. Si la casilla es Escalera, la posición final será la resultante del incremento en casillas correspondiente indicado por la recompensa
  - 3.3.7.3. Si la casilla es Serpiente, la posición final será la resultante del retroceso en casillas correspondiente, indicado por la penalización
  - 3.3.7.4. Si una casilla especial (Serpiente o Escalera) da como resultado que el jugador en turno caiga en otra casilla especial, esta segunda deberá ser ignorada, es decir, sólo se permite un salto o un retroceso por turno

3.3.8. El siguiente ejemplo representa el resultado del primer turno, para el jugador número 1, suponiendo que obtiene un 5 en el dado, y la casilla número 6 es una serpiente con penalización de 3 casillas: **1 1 1 5 S 3**

3.3.9. Cada elemento del turno debe ir separado del anterior por un espacio en blanco

#### 3.4. Fin del juego:

3.4.1. Una vez terminado el juego ya sea por victoria de un jugador, por instrucción de algún jugador o por el límite de turnos, se debe imprimir en una nueva línea el mensaje:

**-- GAME OVER --**

3.4.2. Si se ha alcanzado el máximo número de turnos se debe imprimir el siguiente mensaje:

**The maximum number of turns has been reached...**

3.4.3. Si al final del turno el jugador actual ha alcanzado o superado la casilla final, el juego termina. En este caso se debe imprimir el número de la última casilla del tablero, es decir, si el tablero tiene 30 casillas, no se puede imprimir que la casilla 31. Una vez impresa la información del turno, se debe imprimir el siguiente mensaje indicando qué jugador ganó el juego en lugar del símbolo "#":

**Player # is the winner!!!**

3.4.4. Si no se ha superado la casilla final y se presiona la tecla con la letra mayúscula "E" el juego termina y se debe imprimir el siguiente mensaje:

**Thanks for playing!!!**

## Especificación técnica

1. Escribir un método *main* que inicialice el juego
2. El método *main* debe encontrarse en un archivo denominado *snakes.cpp*
3. Escribir una clase denominada *MyGame* que contenga los atributos y métodos necesarios para representar el juego
4. Dentro de la clase *MyGame* crear un método llamado *start* de tipo *void* que sirva para empezar la ejecución del juego
5. En el método *main* crea una nueva instancia de tu juego y manda llamar el método *start*.
6. El siguiente snippet muestra la estructura mínima que debería tener tu archivo *snakes.cpp*

```
class MyGame
{
    public:
        void start()
        {
            // La lógica de tu juego va aquí
        }
};

int main()
{
    MyGame g;
    g.start();
    return 0;
}
```

7. Utiliza el dado que te hemos proveído dentro del archivo *dice.cpp*. Impórtalo en tu programa principal y utiliza el método *roll()* para simular el dado. Puedes hacer esto utilizando objetos de la siguiente manera:

```
#include "snakes.hpp"
...
...
...

Dice * dice = new Dice();
dice->roll();
```

```
#include "snakes.hpp"
...
...
...

Dice dice();
dice.roll();
```

8. Los mensajes impresos en la consola **NO** deben llevar espacios en blanco al final de cada línea, y debe haber un salto de línea entre cada mensaje
9. Los siguientes son ejemplos de información de cada turno
  - 1 1 1 4 N 5 -> Indica que en el turno 1, el jugador 1 está en la casilla 1, y que al tirar el dado le salió el número 4, lo cuál lo llevaría a una casilla Normal, que sería la número 5
  - 6 2 3 6 S 3 -> Indica que en el turno 6 el jugador 2 estaba en la casilla 3, y que saco un número 6 en el dado, lo cuál debió llevarlo a la casilla 9, misma que tiene una serpiente y por ello retrocedió 6 casillas, quedando nuevamente en la casilla 3
10. El programa debe regresar 0 al finalizar la ejecución

## Recomendaciones

1. Escribe tu código en inglés y sigue los nombres propuestos en la descripción del problema
2. Trata de validar las entradas y salidas, y considera los casos frontera
3. Para esta entrega puedes utilizar arreglos simples de caracteres para las casillas, y de enteros para los jugadores
4. Utiliza g++ para compilar tu programa con la siguiente instrucción desde línea de comandos:  
`>g++ snakes.cpp -o snakes.out`
6. Compila tu programa cada vez que hagas cambios. Si utilizas algún IDE, verifica que tu programa se pueda compilar desde la línea de comandos
7. Una vez compilado, prueba tu programa desde una línea de comandos con la siguiente instrucción:  
`>./snakes.out`
8. Utiliza archivos de entrada y de salida para agilizar las pruebas de tu programa. Esto lo puedes hacer ejecutando tu programa con la siguiente instrucción desde una línea de comandos:  
`>./snakes.out < myInput.txt > myOutput.txt`

## Entradas y salidas

El siguiente es un ejemplo de archivo de entrada que puedes utilizar para ejecutar una simulación de 5 turnos en tu programa:

```
C
C
C
C
C
E
```

El siguiente es un ejemplo de archivo de salida de la ejecución de una simulación completa:

```
Press C to continue next turn, or E to end the game:
1 1 1 3 N 4
2 2 1 5 N 6
3 1 4 5 N 9
4 2 6 1 N 7
5 1 9 1 N 10
6 2 7 2 N 9
...
...
16 2 22 3 N 25
17 1 21 2 N 23
18 2 25 4 N 29
19 1 23 6 N 29
20 2 29 2 N 30
-- GAME OVER --
Player 2 is the winner!!!
```

## Entregables

Deberás entregar un archivo zip que contenga un **README.txt** en el que describas las instrucciones para compilar y ejecutar tu programa, así como una breve discusión sobre los conceptos de programación orientada a objetos que hayas utilizado y la justificación para las decisiones de diseño tomadas. Así mismo deberás incluir todo el código fuente de tu programa en una carpeta llamada *src*. Tu archivo zip debe tener la siguiente estructura y nomenclatura, en donde A00000000 representa tu matrícula:

```
TC1030 - T1 - A00000000.zip
-- README.txt
-- src
---- snakes.cpp
---- dice.cpp
---- otros.cpp
---- ...
```

## Referencias

- [1] Wikipedia, «Serpientes y escaleras,» 27 July 2019. [En línea]. Available: [https://es.wikipedia.org/wiki/Serpientes\\_y\\_escaleras](https://es.wikipedia.org/wiki/Serpientes_y_escaleras).
- [2] F. Hoffmann, F. J. Augustyn Jr and M. J. Manning, Dictionary of toys and games in American popular culture, Routledge, 2013.