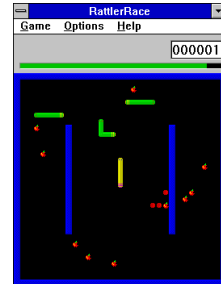


El juego de la serpiente (1ª parte)



La serpiente es un clásico de los videojuegos. Se lanzó a mediados de los años 70, pero alcanzó una popularidad masiva a principios de este siglo, ya que prácticamente todos los teléfonos *Nokia* venían con este juego preinstalado.

El juego consiste en mover una serpiente por un tablero recogiendo la mayor cantidad de manzanas posible, evitando que la serpiente choque con los límites del tablero o consigo misma. La dificultad del juego reside en que la serpiente aumenta de longitud cada vez que come una manzana, de modo que cada vez es más difícil evitar que la serpiente acabe chocando con su propio cuerpo.



Este ejercicio consiste en implementar la lógica de un juego de la serpiente multijugador. En el tablero puede haber varias serpientes, cada una de ellas controlada por un jugador. Por supuesto, las serpientes deben evitar chocar entre ellas, además de evitar chocar consigo mismas. Para facilitar las cosas, consideramos que el tablero es ilimitado, es decir, no tiene paredes. Además, cada serpiente tiene una puntuación que aumenta cada vez que come una manzana. Algunas manzanas otorgan más puntos que otras, pero también pueden provocar que la longitud de la serpiente que se las come aumente más que otras.

Se debe implementar un TAD JuegoSerpiente con las operaciones que se indican a continuación. Ninguna de ellas debe realizar operaciones de entrada/salida.

- `JuegoSerpiente()`

Constructor. Crea una nueva partida cuyo tablero no tiene serpientes ni manzanas.

- `void nueva_serpiente(const string &nombre, const Posicion &posicion)`

Añade una nueva serpiente al tablero con el nombre pasado como parámetro. La longitud de esta serpiente será 1 (es decir, la serpiente ocupará una única casilla), y estará situada en la posición indicada. Si ya existe una serpiente con el mismo nombre en el juego, se lanzará una excepción `domain_error` con el mensaje `Serpiente ya existente`. Si la posición en la que se quiere añadir la serpiente está ocupada por otra serpiente o por una manzana, se lanzará una excepción `domain_error` con el mensaje `Posicion ocupada`.

- `void nueva_manzana(const Posicion &posicion, int crecimiento, int puntuacion)`

Añade una nueva manzana al tablero en la posición indicada. La serpiente que se coma esa manzana aumentará inmediatamente su puntuación en la cantidad indicada en parámetro `puntuacion`, pero durante sus próximos n movimientos, donde n es el valor indicado en el parámetro `crecimiento`, aumentará de longitud. Si ya existe una serpiente u otra manzana en la casilla indicada por `posicion`, se lanzará una excepción `domain_error` con el mensaje `Posicion ocupada`.

- `int puntuacion(const string &nombre) const`

Devuelve la puntuación de la serpiente cuyo nombre es el pasado como parámetro. La serpiente ha de estar en la partida. Si no existe una serpiente con ese nombre en la partida, se lanzará una excepción `domain_error` con el mensaje `Serpiente no existente`.

- `Elemento que_hay(const Posicion &p) const`

Devuelve el tipo de elemento en la posición `p` del tablero. Si hay una serpiente o una manzana, debe devolverse el valor `Serpiente` o `Manzana`, respectivamente. Si no hay ninguna de las dos cosas, debe devolverse el valor `Nada`.

- `bool` avanzar(`const` string &nombre, `const` Direccion &dir)

Avanza la cabeza de la serpiente con el nombre indicado en la dirección dir pasada como parámetro. Esta última puede tener el valor Norte, Sur, Este u Oeste. La cabeza de la serpiente se moverá en esa dirección, y el cuerpo de la misma irá detrás de ella. Si la nueva posición de la cabeza está ocupada por otra serpiente, o por su propio cuerpo, la serpiente se eliminará de la partida y del tablero y la función devolverá true. En caso contrario, devolverá false. Si la nueva posición de la cabeza está ocupada por una manzana, la manzana desaparecerá del tablero y se sumará su puntuación a la de la serpiente. Además, en ese caso, la serpiente aumentará de longitud en sus próximas llamadas a avanzar (tantas como indique el valor de crecimiento de esa manzana). Si la partida no contiene una serpiente con el nombre indicado, se lanzará una excepción domain_error con el mensaje Serpiente no existente.

Consulta el apéndice al final de este enunciado para más información sobre los movimientos de las serpientes y los efectos de las manzanas.

Los tipos Posicion, Direccion y Elemento se definen del siguiente modo:

```
struct Posicion { int x, y; };
enum class Direccion { Norte, Sur, Este, Oeste };
enum class Elemento { Manzana, Serpiente, Nada };
```

Justifica el coste de cada una de las operaciones.

Entrada

La entrada consta de una serie de casos de prueba. Cada caso contiene una serie de líneas en las que se describen las operaciones que deben invocarse sobre una instancia del TAD. Cada línea consiste en el nombre de la operación seguido de sus argumentos. Suponemos que los nombres de serpientes están formados por caracteres alfanuméricos sin espacios. Las posiciones se representan mediante dos números X Y que representan las coordenadas del tablero. Las direcciones se representan mediante las letras N, S, E y O.

Cada caso de prueba finaliza con una línea con la palabra FIN.

Salida

Las operaciones nueva_serpiente, nueva_manzana no imprimen nada, salvo en caso de error. Con respecto al resto, debe imprimirse el resultado de la operación:

- puntuacion: debe imprimirse una línea con el texto `X tiene Y puntos`, donde X es el nombre de la serpiente e Y es su puntuación.
- que_hay: debe imprimirse una línea con la palabra `MANZANA`, `SERPIENTE` o `NADA`.
- avanzar: si devuelve false, no debe imprimirse nada. Si devuelve true, hay que imprimir `X muere`, donde X es el nombre de la serpiente que se ha intentado avanzar.

Si alguna de las operaciones produce una excepción, debe imprimirse una línea con el texto `ERROR:` seguido del mensaje de la excepción, y no debe imprimirse nada más para esa operación.

Al finalizar el caso de prueba, debe imprimirse una línea con tres guiones: ---.

Entrada de ejemplo

```
nueva_serpiente olivia 2 1
nueva_serpiente obdulia 1 3
nueva_serpiente ovidia 3 3
nueva_manzana 2 2 2 100
que_hay 2 2
avanzar olivia N
que_hay 2 2
que_hay 2 1
avanzar olivia N
que_hay 2 3
que_hay 2 2
avanzar olivia N
que_hay 2 2
avanzar olivia 0
que_hay 2 2
avanzar obdulia E
avanzar olivia S
avanzar ovidia 0
FIN
nueva_serpiente ofidia 0 0
avanzar ofidia 0
nueva_manzana 0 0 10 10
avanzar ofidia E
avanzar ofidia E
avanzar ofidia 0
avanzar ofidia 0
FIN
nueva_serpiente odila 0 0
nueva_manzana -1 0 3 10
puntuacion odila
avanzar odila 0
puntuacion odila
avanzar odila N
avanzar odila E
avanzar odila S
avanzar odila 0
FIN
nueva_serpiente odila 0 0
nueva_manzana -1 0 4 10
puntuacion odila
avanzar odila 0
puntuacion odila
avanzar odila N
avanzar odila E
avanzar odila S
avanzar odila 0
FIN
```

Salida de ejemplo

```
MANZANA
SERPIENTE
NADA
SERPIENTE
SERPIENTE
SERPIENTE
NADA
obdulia muere
---
ofidia muere
ERROR: Serpiente no existente
---
odila tiene 0 puntos
odila tiene 10 puntos
---
odila tiene 0 puntos
odila tiene 10 puntos
odila muere
---
```

Apéndice: reglas de movimiento de las serpientes

1. La serpiente ocupa una serie de posiciones contiguas en el tablero. Cada casilla se identifica con unas coordenadas (X, Y) . En este sentido, la dirección *norte* se corresponde con la dirección del eje Y y la dirección *este* con la del eje X .
2. Cuando la cabeza de la serpiente se mueve en una dirección, el resto del cuerpo sigue a la cabeza, como se muestra en la Figura 1 (ver página siguiente). En esta figura, dada una posición inicial, se muestra cómo se movería la serpiente si avanzase hacia el norte, sur y el este. Observa que, en esa posición, la serpiente no puede avanzar hacia el oeste, ya que chocaría con su propio cuerpo y moriría.
3. Cuando una serpiente tiene longitud 1, puede moverse en cualquiera de las cuatro direcciones sin chocar consigo misma.
4. En general, la cola también se desplaza siguiendo el resto del cuerpo, pero si la serpiente se encuentra creciendo por el efecto de una manzana, la cola se mantiene en la misma posición, provocando que la serpiente aumente su longitud en una unidad (ver Figura 2).
5. Si la cabeza de una serpiente se mueve justo a la casilla en la que está su cola (ver Figura 3), en principio no chocará consigo misma, ya que la cola también se desplaza simultáneamente. Sin embargo, si la serpiente está creciendo debido a los efectos de una manzana, sí chocará consigo misma, ya que la cola se mantendría donde está.
6. Los efectos de la manzana son acumulativos. Si una serpiente se come una manzana cuando aún está sufriendo los efectos de otra, la última manzana comida no empezará a hacer efecto hasta que termine el efecto de la manzana comida anteriormente.

Créditos

Autor: Manuel Montenegro

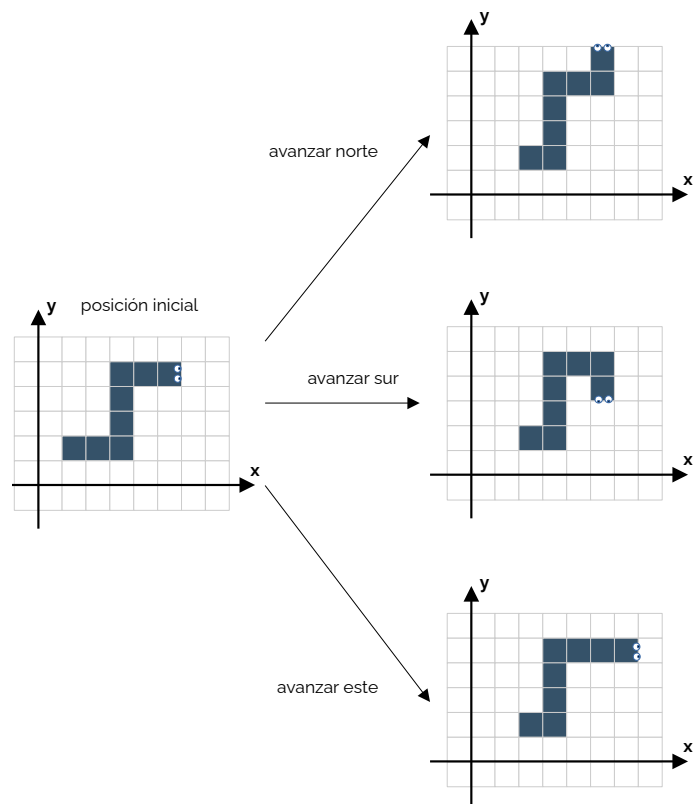


Figura 1: Movimientos de la serpiente cuando no está creciendo

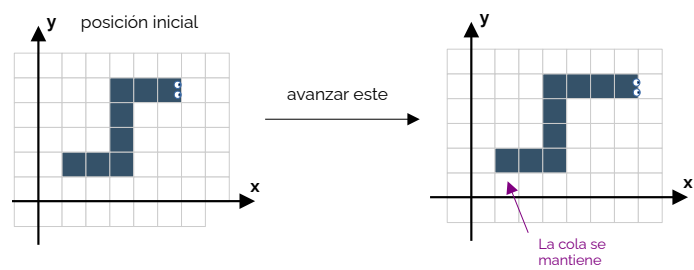


Figura 2: Movimientos de la serpiente cuando crece

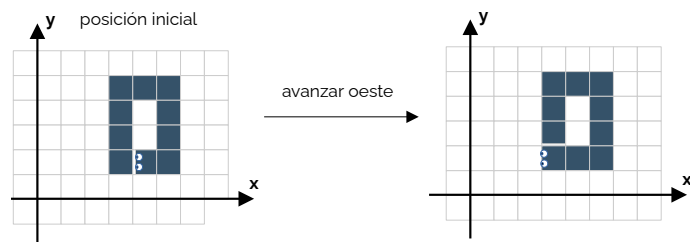


Figura 3: Movimientos de la serpiente cuando se mueve hacia su cola