

Explorando la mina

Nuestro minero ha llegado a la mina y va a entrar en ella. Seguiremos sus movimientos consultando nuestro plano del merodeador. El minero irá avanzando por la mina y el plano contendrá en cada momento su posición.

El minero puede moverse a la izquierda o derecha del plano manteniendo el nivel en la mina, puede subir un nivel en la mina o bajar al nivel inmediatamente inferior. Si se encuentra en una galería ya excavada (**LIBRE**) o en una galería con tierra (**TIERRA**) avanzará por ella en el sentido y dirección que se indique. Si se encuentra con un muro (**MURO**) no puede atravesarlo, por lo que permanecerá donde está chocándose con el muro hasta que cambie de sentido o dirección. Si se encuentra una piedra (**PIEDRA**) no puede atravesarla, pero puede desplazarla si detrás de la piedra hay una galería excavada (**LIBRE**).

En este caso al avanzar hacia la piedra esta se desplaza al sitio libre y el minero puede ocupar la posición de la piedra. Si detrás de la piedra la galería no está excavada entonces el minero no puede pasar y permanecerá chocándose hasta que cambie de sentido o dirección. Si se encuentra con un diamante (**GEMA**) lo recoge y avanza a la posición del diamante. Por último si se encuentra con la salida (**SALIDA**) saldrá al exterior y se acabará su expedición. El minero también puede decidir que abandona la exploración sentándose en el suelo hasta que le vengán a buscar.

Tu tarea es seguir los pasos del minero.



Requisitos de implementación Este problema está asociado a la práctica que se debe realizar en la asignatura. Debe resolverse después del problema PR01. Debes seguir las siguientes indicaciones.

Crea un módulo `juego`, con dos ficheros: `juego.h` y `juego.cpp`.

En el fichero `juego.h` se definirán las siguientes estructuras de datos y prototipos.

Un tipo enumerado `tTecla` para definir el movimiento del minero. El minero puede avanzar en horizontal hacia la izquierda (**IZDA**) o hacia la derecha (**DCHA**). Puede desplazarse en vertical hacia arriba (**ARRIBA**) o hacia abajo (**ABAJO**). Utilizaremos el valor **SALIR** para indicar que el minero abandona la exploración.

Un tipo enumerado `tEstado` que define el estado en que se encuentra el minero. Este puede ser **EXPLORANDO** si el minero continua buscando diamantes por la mina, **EXITO** si el minero consigue llegar a la salida, **ABANDONO** si el minero decide sentarse a esperar que le rescaten y **FRACASO** si el minero muere durante la exploración de la mina.

Un `struct tJuego`, como:

- Una mina (`tMina`)
- El número de gemas que ha recogido el minero (`int`)
- El número de movimientos que ha realizado el minero (`int`)
- El estado en que se encuentra el minero.
- La opción seleccionada para dibujar el juego (`int`)
- La opción seleccionada sobre introducir los movimientos por teclado o por fichero. (`int`)

Y los prototipos de las funciones implementadas.

En el fichero `juego.cpp`, debes declarar el vector de direcciones `tdirs4`, para moverte en las cuatro direcciones del tablero. Las direcciones deben coincidir con el orden en que se han declarado los valores del tipo enumerado. Si el primer valor del tipo enumerado es **ARRIBA**, entonces el primer valor del vector debe ser `(-1,0)` etc..

Debes implementar las siguientes funciones:

1. `std::istream& operator<< (std::istream & movimientos, tTecla& tecla);`, lee del flujo de entrada `movimientos` un movimiento y devuelve el valor del tipo enumerado correspondiente.

2. `void leerMovimiento(tJuego & juego, tTecla& tecla, std::istream & movimientos);` lee un movimiento del fichero con la sobrecarga del operador. Si el minero selecciona el movimiento SALIR pondrá el estado del minero en situación de ABANDONO.
3. `void realizarMovimiento (tJuego & juego, tTecla mov);` se encarga de realizar el movimiento indicado en el parámetro `mov` modificando el plano de la mina y la posición del minero en la mina. Para obtener la casilla de destino se ayudará del vector `tdirs4`, de forma que no tiene que diferenciar el movimiento realizado.
4. `void dibujar(tJuego const& juego);` manda dibujar el plano de la mina a la escala que tiene guardada en las opciones del juego.
5. `void jugar(tJuego& juego, std::istream& entrada, std::istream & movimientos);` comienza cargando los datos de la mina, a continuación va leyendo los movimientos y para cada uno modifica la mina de acuerdo con el movimiento. Los movimientos no los almacena, los trata según los va leyendo. La función debe recibir los dos flujos de entrada, porque en fases posteriores del desarrollo de la práctica será necesario tenerlos diferenciados.

Debes añadir una función en el módulo mina, `bool dentroPlano (tMina const& mina, int x, int y)` que compruebe si una posición está dentro del plano.

La función `resuelveCaso` lee los parámetros del juego: escala del dibujo y entrada de movimientos por teclado; y llama a la función `jugar`. Al llamar a la función se utilizará la entrada por consola (`std::cin`) para el fichero de la mina (`entrada`) y para el fichero de movimientos (`movimientos`). Por último muestra si el minero ha salido de la mina, cuantos movimientos ha realizado y cuántas gemas ha recolectado, seguido del plano de la mina cuando el minero ha acabado de excavarla.

Entrada

La entrada comienza con el número de minas que queremos explorar. La descripción de cada exploración comienza con una línea en que se dan los parámetros del juego: escala a la que se mostrará el plano, 1 si se muestra a escala 1:1 y 2 si se muestra a escala 1:3; a continuación un 2 para indicar que la entrada será por fichero.

En la línea siguiente se muestran las dimensiones del plano, el primer valor será el número de filas n y el segundo el número de columnas m . A continuación se muestran n líneas cada una con m valores indicando el elemento que se encuentra en ese punto de la mina. Los caracteres que describen la mina en el fichero de entrada son: el minero (J), tierra (T), piedra (P), gema (G), salida (S). Los túneles ya cavados se muestran con caracteres blancos.

La última línea del caso indica los movimientos realizados por el minero. Si el minero sube de nivel se indica con una A, si el minero baja de nivel Z, el minero se mueve a la derecha M, el minero se mueve a la izquierda N. El final de los movimientos se indica con el carácter S, que indica que el minero abandona la exploración, se sienta en el suelo y ya no se moverá mas.

Salida

Para cada caso se muestra una línea con el número de movimientos totales realizados. En la línea siguiente se muestra el total de gemas recolectadas y en las líneas siguientes se dibujará el plano de la mina cuando se han acabado de realizar todos los movimientos. Para ello se utilizarán los mismos caracteres que en la práctica anterior.

Si el minero ha conseguido llegar a la salida de la mina, se escribirá en la línea posterior al plano **Mina recorrida**, en caso de que abandone y se siente escribirá **Pide rescate**. Mira el formato en el ejemplo de salida.

Entrada de ejemplo

```
3
1 2
5 6
MMMMMM
MJTMSM
MTTMTM
MTGTTM
MMMMMM
ZZMMMAA
1 2
5 6
TTTTTT
TJ--SM
GP--TT
MMTMGM
TTTTTT
NZMMZZMMAAA
1 2
4 3
JTT
TPT
MMS
MMM
ZMMMMS
```

Salida de ejemplo

```
Movimientos: 7
Gemas: 1
XXXXXX
X-.XMX
X-.X-X
X----X
XXXXXX
Mina recorrida
Movimientos: 11
Gemas: 2
.....
----MX
---@-.
XX-X-X
..---.
Mina recorrida
Movimientos: 5
Gemas: 0
-..
M@.
XXS
XXX
Pide rescate
```

Autor: Isabel Pita.