Universidad Tecnológica Centroamericana
UNITEC

Laboratorio de Programación I

Laboratorio #6

Carlos Espinal

Onasis Reyes

25 de mayo de 2024

Desarrollo de la práctica

Para el desarrollo satisfactorio del laboratorio, siga al pie de la letra cada instrucción que a continuación se le presenta.

Cree un nuevo proyecto Java con el nombre **Lab#P1_PrimerNombrePrimerApellido** (**Lab6P1_CarlosEspinal**). Recuerde hacer su código robusto, tabulado y estar autodocumentado, la falta de alguna de éstas, incluyendo el nombre, será reflejado en su nota. **También debe crear un repositorio en GitHub con el mismo formato de nombre.**

Debe realizar 8 pushes:

1. 01:30 pm
2. 02:00 pm
3. 02:30 pm
4. 3:00 pm
5. 3:30 pm
6. 4:00 pm
7. 4:30 pm
8. 5:00 pm

Objetivos del laboratorio

- Evaluar los temas vistos hasta la semana actual
- Manejo de métodos
- Matrices

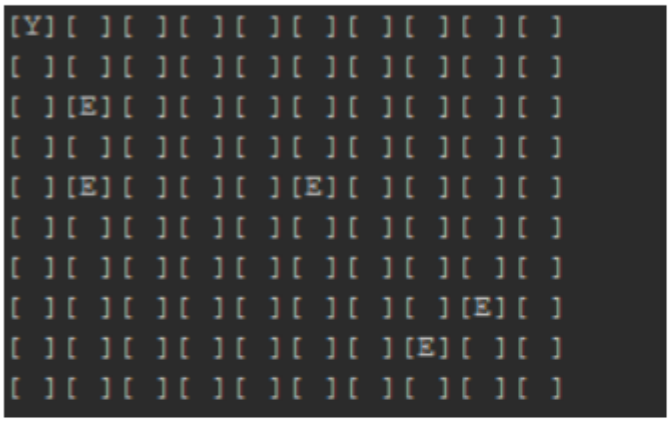
Consideraciones

- Su programa debe tener un menú con una opción correspondiente a cada ejercicio. La opción 0 deberá ser para salir del programa, por lo tanto, la ejecución terminará.
- El uso de **drag and drop no es permitido** a la hora de hacer los commits.
- Si en cualquier momento del laboratorio usted tiene una duda, hágasela saber a los instructores. Puede utilizar las herramientas de Zoom, como levantar la mano, escribir en el chat y realizar la pregunta que se tiene.
- Uso de variables con nombres significativos.
- **La copia de código será penalizada con la máxima penalización en la nota del laboratorio y será remitido al Comité de Ética.**
- Seguir instrucciones de cada ejercicio. Haga las validaciones necesarias.
- **Si usted en su estudio personal descubre métodos que no se vieron en clases, pregunte a su instructor si los puede utilizar, de lo contrario, se verá reflejado en su nota.**
- **NO SE ACEPTARÁN LABORATORIOS ENVIADOS AL CORREO DEL INSTRUCTOR, a menos que se haya hablado con el mismo previamente.**

Ejercicio Práctico #1 – Water Emblem: 2 Houses

Después de la muerte de la emperatriz del Imperio Adrestian, Edelgard, el malvado Obsidian Noctus ha tomado el liderazgo de este imperio. La región de Faerghus ha sido victima de las malas decisiones del nuevo emperador, se están preparando para iniciar una guerra. Como el programador de alto calibre que es usted, el líder de la región Faerghus, Dimitri Alexandre Blaiddyd, le ha pedido a usted que desarolle un simulador de peleas para poder desarrollar sus tácticas contra el Imperio corrompido.

Tras elegir la opción del simulador de tácticas en el menú principal, se deberá imprimir una matriz de caracteres 10x10 la cual es el tablero del simulador:



El carácter 'Y' representa a la persona siendo controlada. Los caracteres 'E' representan a los enemigos. Se deberá poder mover a la persona 'Y' cerca de los enemigos 'E' para poder atacarlos y respectivamente matarlos.

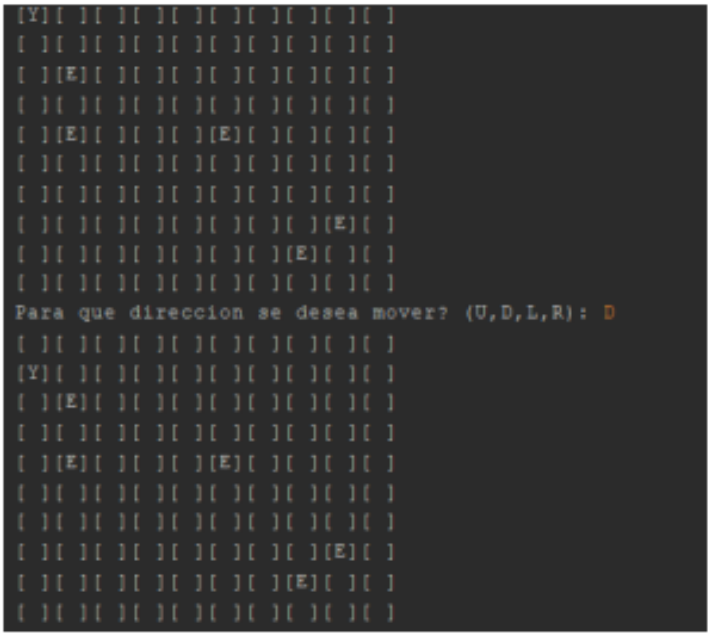
El programa consiste de los siguientes métodos:

- llenarTablero: recibe un tablero 10x10 y se encarga de llenar la matriz con 5 enemigos 'E'. Estos estarán en posiciones aleatorias del tablero, estas coordenadas son determinadas por 2 numeros generados con la librería Random. Los enemigos no podrán aparecer en la posición 0,0 de la matriz ya que es donde el personaje representado por el carácter 'Y' aparece. Tras haber establecido las posiciones de los caracteres 'Y' y 'E', se deberá rellenar el resto de la matriz con caracteres ' ', representativos de que se podrá mover a esa celda.
- hayEnemigos: recibe el tablero 10x10 de caracteres, recorre toda la matriz. Cada vez que encuentre un carácter 'E', sumara a un contador. Si el valor del contador es mayor a 0 retorna true, de lo contrario, retornara falso. Este método es el que permite que la simulación continue hasta que no haya enemigos.
- MovimientoJugador: recibe el tablero 10x10 de caracteres, recorre toda la matriz hasta encontrar el carácter 'Y', después, guardara las coordenadas en donde se encuentra ese

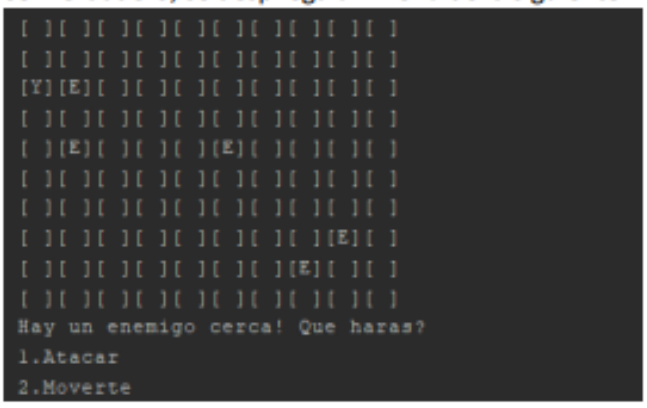
carácter. Tras obtener las coordenadas, le pedirá al usuario a que dirección quiere moverse:

- U: arriba [i-1][j]
- D: abajo [i+1][j]
- L: izquierda [i][j-1]
- R: derecha [i][j+1]

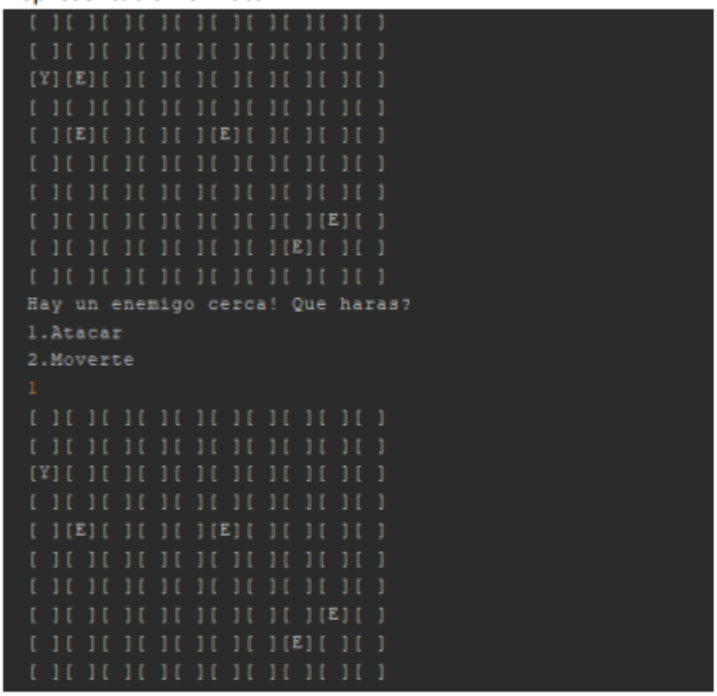
Dependiendo de donde haya elegido el usuario moverse, se reemplazara el carácter ' ' de la casilla destino por un carácter 'Y' y la casilla donde estaba el carácter 'Y' original será reemplazada por un carácter ' '.**IMPORTANTE:** validar que la persona no pueda salirse de los limites del tablero. Foto representación del movimiento:



- enemigosCerca: recibe el tablero de caracteres 10x10 y como el método anterior, obtendrá primero las coordenadas del carácter 'Y'. Retornara un true si se encuentra un carácter 'E' arriba, abajo, izquierda o derecha del carácter 'Y', retornara falso si no se cumple ninguna de las condiciones anteriores. Es el método que se encarga de validar cuando un usuario puede elegir atacar/moverse en vez de solo poder moverse. En caso de ser verdadero, se despliega un menú de la siguiente manera:



- ataqueJugador: recibe el tablero de caracteres de 10x10 y obtendrá la posición del carácter 'Y'. Si el carácter 'Y' tiene un carácter 'E' a la derecha, izquierda, arriba o abajo, lo modificara a ser un carácter ' ' representando que el jugador a asesinado al enemigo. Representación en foto:



- imprimirTablero: recibe el tablero 10x10 de caracteres, lo recorre celda por celda. Si se encuentra en una celda un carácter 'Y' lo imprimirá como [Y], si hay un carácter 'E' lo imprimirá como [E] y si hay un carácter ' ' lo imprimirá como [ ].

IMPORTANTE: se deberá validar en los métodos que revisan la posición del jugador de manera en que, si alguna de sus coordenadas contiene un 0, solo se podrán hacer los movimientos que involucren sumarle a la coordenada. EJEMPLO: [i+1][j] , [j][i+1]

Ejercicio Práctico #2 – Descifra la combinación

Una compañía de seguridad le ha pedido a usted, valiente, capaz y experimentado programador que desactive una bomba. Para desactivarla se debe ingresar al programa un char hasta que o se acaben los intentos y se adivine la combinación o la bomba explote. Se debe crear un método void llamado **combinacion()** el cual no recibe nada, y es aquí donde se desarrollará el ejercicio.

Se debe mostrar al usuario un menú con 3 dificultades:

1. Facil
2. Media
3. Dificil

Una vez seleccionada la dificultad, se debe **generar una matriz de chars aleatoria llamada combinación** que solo incluya **LETRAS MAYÚSCULAS O MINÚSCULAS** la cual no se debe mostrar en pantalla **nunca mientras se está adivinando la combinación**. El tamaño de la matriz y la cantidad de intentos dependerá de la dificultad escogida y se denota de la siguiente manera:

1. Para la dificultad fácil la combinación tendrá:
    - n=3 y un m = n+1
    - intentos hasta explote la bomba 36

2. Para la dificultad media la combinación tendrá:
    - n=3 y un m = n+1
    - intentos hasta explote la bomba 32

3. Para la dificultad difícil la combinación tendrá:
    - n=3 y un m = n+1
    - intentos hasta explote la bomba 28

Se debe **generar una segunda matriz de chars llamada progreso** con el mismo tamaño pero que solo contenga guiones. Por ejemplo, si la dificultad es fácil, la segunda matriz será de tamaño n * m:

```
- - - -
- - - -
- - - -
```

.

Si se adivina una letra, se reemplazará el '-' donde se encuentre esa letra por un '+'. Por ejemplo, si yo ingreso la letra 'a', se vería algo así:

```
- - - +
- - + -
- - - -
```

.

Nótese que por cada turno se deben marcar todas las posiciones donde se encontró la letra. Una letra mayúscula es distinta a una minúscula, aunque sean la misma letra.

Mientras la cantidad de intentos sea mayor que 0 o que la combinación siga siendo desconocida, se le pedirá al usuario que ingrese un char cualquiera el cual se estará revisando si existe o no en la combinación. **El mensaje exacto** con el cual se le indicará al usuario su progreso, cuantos intentos le quedan y como pedirá al usuario que ingrese el char está en el ejemplo.

También debe crear estos métodos:

1. Un método que genere la combinación teniendo de parámetro el tamaño de la matriz y que retorne una matriz aleatoria. **PISTA: Utilizar método Random de java**
2. Un método que tenga de parámetros la combinación, el progreso y la letra ingresada y debe retornar la **matriz progreso** actualizada.
3. Es decir, si mi combinación es

```
H b C d
E f G h
I h K l
```

mí progreso inicialmente será

```
- - - -
- - - -
- - - -
```

y si yo ingreso la letra o, mí progreso será

```
- - - -
- - - +
- + - -
```

.

Deben utilizar variables significativas.

De nuevo, RECUERDE QUE INGRESAR UNA LETRA MAYÚSCULA ES DISTINTO A UNA MINÚSCULA.

Ejemplo para ganar:

```
--- Lab 6 ---
0. Salir
1. Water Emblem
2. Descifra la combinacion
Selecciona -> [0, 1, 2]: 2

---Dificultades---
1. Facil
2. Media
3. Dificil
Selecciona -> [1, 2, 3]: 1

- - - -
- - - -
- - - -
Tiene 0 letras descifradas, le quedan 36 intentos
Ingrese letra: h

- - - -
- - - +
- + - -
Tiene 2 letras descifradas, le quedan 35 intentos
Ingrese letra: H

+ - - -
- - - +
- + - -
Tiene 3 letras descifradas, le quedan 34 intentos
Ingrese letra: b
+ + - -
- - - +
- + - -
Tiene 4 letras descifradas, le quedan 33 intentos
Ingrese letra: b

+ + - -
- - - +
- + - -
Tiene 4 letras descifradas, le quedan 32 intentos
Ingrese letra: C

+ + + -
- - - +
- + - -
Tiene 5 letras descifradas, le quedan 31 intentos
Ingrese letra: d

+ + + +
- - - +
- + - -
Tiene 6 letras descifradas, le quedan 30 intentos
Ingrese letra: E

+ + + +
+ - - +
- + - -
Tiene 7 letras descifradas, le quedan 29 intentos
Ingrese letra: f
```

```
+ + + +
+ + - +
- + - -
Tiene 8 letras descifradas, le quedan 28 intentos
Ingrese letra: G

+ + + +
+ + + +
- + - -
Tiene 9 letras descifradas, le quedan 27 intentos
Ingrese letra: I

+ + + +
+ + + +
+ + - -
Tiene 10 letras descifradas, le quedan 26 intentos
Ingrese letra: K

+ + + +
+ + + +
+ + + -
Tiene 11 letras descifradas, le quedan 25 intentos
Ingrese letra: l

Has descifrado la combinacion

H b C d
E f G h
I h K l

--- Lab 6 ---
0. Salir
1. Water Emblem
2. Descifra la combinacion
Selecciona -> [0, 1, 2]:
```

Ejemplo para perder:

```
- - - -
- - - -
- - - -
Tiene 0 letras descifradas, le quedan 3 intentos
Ingrese letra: q

- - - -
- - - -
- - - -
Tiene 0 letras descifradas, le quedan 2 intentos
Ingrese letra: q

- - - -
- - - -
- - - -
Tiene 0 letras descifradas, le quedan 1 intentos
Ingrese letra: q

La bomba a explotado, la combinacion era:

H b C d
E f G h
I h K l

--- Lab 6 ---
0. Salir
1. Water Emblem
2. Descifra la combinacion
Selecciona -> [0, 1, 2]:
```

Ponderación

Elemento	Puntaje
Ejercicio #1	3.875
Ejercicio #2	3.875
Instrucciones	1
Total	**8.75**

Comentarios adicionales

El uso de variables significativas será considerado en cada ejercicio. Si desea utilizar algo que usted aprendió en su estudio individual, **consulte con un instructor** antes de usarlo. Recuerde invitar a los instructores como colaboradores, crear su repositorio en privado y subir su laboratorio a Canvas (link del repositorio) dentro del tiempo establecido. **El no seguir las instrucciones tal y como se especifican, se verá reflejado en su nota.**