



Examen #1, II-2020

Miércoles 14 de octubre del 2020

Prof. Maureen Murillo

Indicaciones generales

Relacionadas con la entrega:

1. Cuenta con 3 horas para desarrollar la solución del examen.
2. Una vez transcurridas las 3 horas tendrá 30 minutos adicionales para subir un archivo .jar (cuyo nombre de archivo sea el nombre y el apellido suyo) a mediación virtual. El enlace de mediación virtual se cerrará a las 2:30pm en punto. Note que estos 30 minutos adicionales se otorgan para solventar cualquier problema técnico para la entrega del archivo. Queda a su criterio si utiliza parte de este tiempo para seguir programando, quedando bajo su responsabilidad cualquier problema que se le presente para subir el archivo (falla eléctrica, falla de internet, problemas con mediación, etc).
3. El proyecto entregado en mediación virtual debe compilar, tal como está establecido en la carta al estudiante. De no cumplir con esta característica se considerará como no entregado y tendrá un 0 en el examen.
4. El desarrollo completo del examen (desde el inicio hasta el fin) deberá realizarse utilizando el repositorio personal de Github creado para este curso, en la carpeta de exámenes. Este repositorio deberá estar compartido únicamente con el(la) asistente y la profesora. Deberá contener los commit-push que se le indiquen durante el examen, no importa si en ese momento no compila. Si el examen no lo desarrolla asociado con este repositorio y con los commit-push que la profesora indique durante el examen, se considerará no entregado el examen aunque haya subido el examen en mediación virtual. El objetivo de esta forma de trabajo es validar la autenticidad de su examen.

Relacionadas con el desarrollo:

1. Según los lineamientos de la Universidad de Costa Rica en cuanto a evaluaciones virtuales sincrónicas, para tener derecho a realizar el

examen el estudiante deberá permanecer todo el tiempo en la sesión de Zoom y deberá tener encendida la cámara en donde se vea claramente en todo momento su rostro. De ser necesario se le solicitará que encienda el micrófono. De tener algún inconveniente con estos requerimientos debe conversarlo previamente con la docente.

2. Descargue el enunciado de mediación virtual y trabaje en BlueJ. Recuerde que debe trabajar conectado con el repositorio de Github indicado anteriormente.
3. El examen es individual. Es prohibido interactuar con cualquier otra persona que no sea la profesora.
4. Puede utilizar cualquier material y código propio escrito por usted (prácticas, tareas, libros, apuntes) y material de Internet debidamente referenciado. Si no cumple con estas características se considerará fraude.
5. En el comentario principal de cada clase, como autor indique su número de carné y su nombre.
6. Cada cierto tiempo la profesora indicará que deben hacer un commit-push en el repositorio, no importa si en ese momento no compila el examen, para lo cual debe estar atento con el audio encendido en todo momento. Es obligatorio hacer estos commit-push, de lo contrario la entrega del examen se considerará inválida y tendrá un cero como nota.

Relacionadas con un imprevisto:

1. Cualquier imprevisto técnico durante el examen deberá comunicarlo inmediatamente a la profesora por medio de un mensaje privado en Telegram.
2. Deberá estar atento(a) al chat grupal y al personal de Telegram, en caso de ser necesario que la profesora dé indicaciones.
3. Si durante el período de entrega del examen, entre las 2:00pm y las 2:30pm, mediación virtual estuviera “caído”, envíe el archivo .jar al correo institucional de la profesora con copia al asistente.
4. Si posee una computadora portátil, es una buena alternativa que la utilice para realizar el examen, asegurándose de que la batería está cargada al 100%, para que en caso de una falla eléctrica pueda continuar el examen. Si realiza el examen en una computadora de escritorio que depende de energía eléctrica y sucede un fallo en el suministro, debe continuar su examen en papel. En este caso se calificará la versión subida hasta ese

momento a Github, así como la documentación que aporte en papel. Recuerde reportarlo a la profesora tan pronto suceda el imprevisto.

5. Si normalmente utiliza el internet de la casa con señal wifi y se va la electricidad, pero posee internet en el celular infórmelo inmediatamente a la profesora por un mensaje privado de Telegram. En estos casos, de ser posible conecte la computadora a la señal del celular para realizar al menos los commit-push y para subir el examen en mediación.

Enunciado

Usted debe programar el juego llamado **MaDi** que se describe a continuación. Se posee un tablero de $N \times N$ (el tamaño N lo define el usuario y debe ser un número mayor o igual a 2), en donde en cada casilla hay una instrucción. La primera (la $[0][0]$) y la última casilla (la $[N-1][N-1]$) no tienen instrucción. Las posibles instrucciones son:

1. No se mueva
2. Avance 4 lugares
3. Brinque a la próxima fila
4. Retroceda 2 lugares
5. ¡Explotó! Fin del juego.

Para facilitar la visualización de la matriz al jugador se desplegará el número de instrucción en el tablero y debajo de éste la lista de instrucciones con su numeración. Por ejemplo:

-	5	3	2
3	4	3	1
1	3	2	5
2	1	3	-

Instrucciones:

1. No se mueva
2. Avance 4 lugares
3. Brinque a la próxima fila
4. Retroceda 2 lugares
5. ¡Bomba! Explotó.

La instrucción que va en cada casilla del tablero se asignará aleatoriamente cada vez que se inicial el juego.

El jugador inicia con su ficha en la primera posición (fila:0, columna:0) y su objetivo es avanzar por el tablero hasta llegar a la última posición (esquina de abajo-derecha). En cada turno el jugador lanza un dado que le dirá cuántos espacios avanzar. Se avanza por el tablero de izquierda a derecha y de arriba hacia abajo de la siguiente manera:

En cada turno el jugador lanza un dado que le indica cuántas posiciones del tablero avanzar. Luego de avanzar deberá ejecutar lo que indique la instrucción de la casilla en donde cayó. Por cada turno, solamente ejecuta una instrucción del tablero. El jugador posee una cantidad máxima de lanzamientos del dado, la cual se le preguntará al jugador al iniciar el juego.

El juego se pierde si se le acaban los lanzamientos del dado sin haber llegado a la meta o si producto de lanzar el dado avanza a una casilla con una bomba. El juego se gana si llega a la meta (no es necesario que sea por cuenta exacta) antes de que se le acaben los lanzamientos del dado.

(20%) **Clases y atributos.** Usted debe programar una clase Instrucción, la cual está compuesta por un identificador (un número entre 1 y 5) y por un texto que la describe. También debe programar la clase Tablero, la cual será una matriz (de corchetes `[][]`, no puede usar ninguna clase predefinida de Java) de objetos de tipo Instrucción. Además, deberá incluir el resto de clases necesarias según el modelo de clases visto en el curso.

(15%) **Constructores.** El constructor de la clase Instrucción se encargará de definir aleatoriamente la información de la instrucción según las 5 posibles descritas anteriormente. El constructor de la clase Tablero creará la matriz cuadrada según el tamaño que el jugador haya definido y la llenará de instrucciones.

(5%) **toString.** Programe el método toString en la clase Tablero que devuelva (NO debe mostrar nada) una hilera con el contenido de la matriz en una disposición de filas por columnas, seguida de la lista de instrucciones con su numeración. Por ejemplo, si la matriz contiene las instrucciones del ejemplo indicado anteriormente, el método debería devolver una hilera con lo siguiente:

-	5	3	2
3	4	3	1
1	3	2	5
2	1	3	-

Instrucciones:

1. No se mueva
2. Avance 4 lugares
3. Brinque a la próxima fila
4. Retroceda 2 lugares
5. ¡Explotó! Fin del juego

(35%) **Turnos.** El programa debe desarrollar el juego y avanzar la ficha apropiadamente. Deberá mostrar en una sola ventana el resumen de lo que sucede en cada lanzamiento del dado, a saber:

la matriz con sus instrucciones

el valor que el dado sacó

la posición temporal adonde lo llevó el dado

la posición definitiva luego de ejecutar la instrucción de la casilla anterior

Lanzamientos restantes

Si una instrucción solicita realizar un movimiento inválido (por ejemplo, retroceder 2 lugares estando en la primera casilla, o brincar a la próxima fila cuando se está en la última fila), la instrucción simplemente no se ejecuta.

Al iniciar cada juego, se le mostrará al usuario el tablero con las instrucciones y la cantidad de lanzamientos disponible, similar al siguiente:

-	5	3	2
3	4	3	1
1	3	2	5
2	1	3	-

Instrucciones:

1. No se mueva
2. Avance 4 lugares
3. Brinque a la próxima fila
4. Retroceda 2 lugares
5. ¡Explotó! Fin del juego

Cantidad de lanzamientos disponibles: 10

Luego, se le irá mostrando una ventana por cada lanzamiento del dado, similar a la siguiente:

-	5	3	2
3	4	3	1
1	3	2	5
2	1	3	-

Instrucciones:

1. No se mueva
2. Avance 4 lugares
3. Brinque a la próxima fila
4. Retroceda 2 lugares
5. ¡Explotó! Fin del juego

Dado sacó: 2

Posición temporal al avanzar por el valor del dado (fila, columna): 0,2

Posición definitiva al ejecutar la instrucción de la casilla: 1,2

Cantidad de lanzamientos disponibles: 9

(15%) **Fin del juego.** El juego termina por alguno de los siguientes tres motivos: el jugador alcanza la meta, se le acaban los lanzamientos de dado o explota. Deberá informársele al jugador este resultado.

(10%) **Repetición del juego.** Su programa debe mostrar un menú que permita al usuario jugar cuantas veces quiera. Recuerde que cada vez que reinicia el juego, deberá preguntársele al jugador por la cantidad de lanzamientos, el tamaño de la matriz y volverla a llenar aleatoriamente de instrucciones.

Lineamientos generales de evaluación

- a) Cumplimiento de buenas prácticas de programación (convenciones).
- b) Correcto diseño y programación según el paradigma orientado a objetos.
- c) Uso correcto de las estructuras de datos.
- d) Uso correcto de estructuras de control.
- e) Correctitud de algoritmos y funcionamiento del programa.
- f) Buen diseño de la interfaz de usuario: mensajes claros, facilidad de uso, programación defensiva (que no se caiga por errores de entrada del jugador).