UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

Programación II

Práctica Calificada 2 (de verdad)
Pregrado
2022-II
Profesor: Ian Paul Brossard
Lab 2.03

Indicaciones específicas:

- Esta evaluación contiene 9 páginas (incluyendo esta página) con 1 pregunta. El total de puntos es 20.
- El tiempo límite para la evaluación es 100 minutos.
- Cada pregunta deberá ser respondida en un archivo diferente. Se deberá usar los siguientes nombres de archivos:
  - p1.cpp
  - p2.cpp
  - p3.cpp
- Deberá subir sus archivos directamente a www.gradescope.com.
- Recuerde que Gradescope solo conserva el último envío que se realiza. Si va a modificar su entrega, debe volver a adjuntar todos los archivos de su práctica.

Competencias:

- Para los alumnos de la carrera de Ciencia de la Computación
  - Aplicar conocimientos de computación apropiados para la solución de problemas definidos y sus requerimientos en la disciplina del programa. (Evaluar)
  - Analizar problemas e identificar los requerimientos computacionales apropiados para su solución. (Usar)
  - Usar técnicas y herramientas actuales necesarias para la práctica de la computación. (Usar)
- Para los alumnos de las carreras de Ingeniería
  - Capacidad de aplicar conocimientos de matemáticas (nivel 3)
  - Capacidad de aplicar conocimientos de ingeniería (nivel 2)
  - Capacidad para diseñar un sistema, un componente o un proceso para satisfacer necesidades deseadas dentro de restricciones realistas (nivel 2)

Calificación:

Tabla de puntos (Sólo para uso del profesor)

Question	Points	Score
1	7	
2	6	
3	7	
Total:	20	

Preguntas

1. **(7 puntos) Matrices dinámicas:**

   Super Mario Bros. o Super Mario Brothers es un videojuego publicado en 1985 para la consola Nintendo Entertainment System. Cada nivel de dicho juego consiste en un mapa que tiene 16 casillas de altura y una cantidad variable de ancho. Algunos niveles son más largos que otros, pero todos tienen 16 casillas de altura.

   Los niveles están conformados por diferentes piezas. A estas piezas se les conoce como baldosas o azulejos. En este examen usaremos el término «azulejo». A continuación, se muestra el primer nivel de Super Mario Bros.

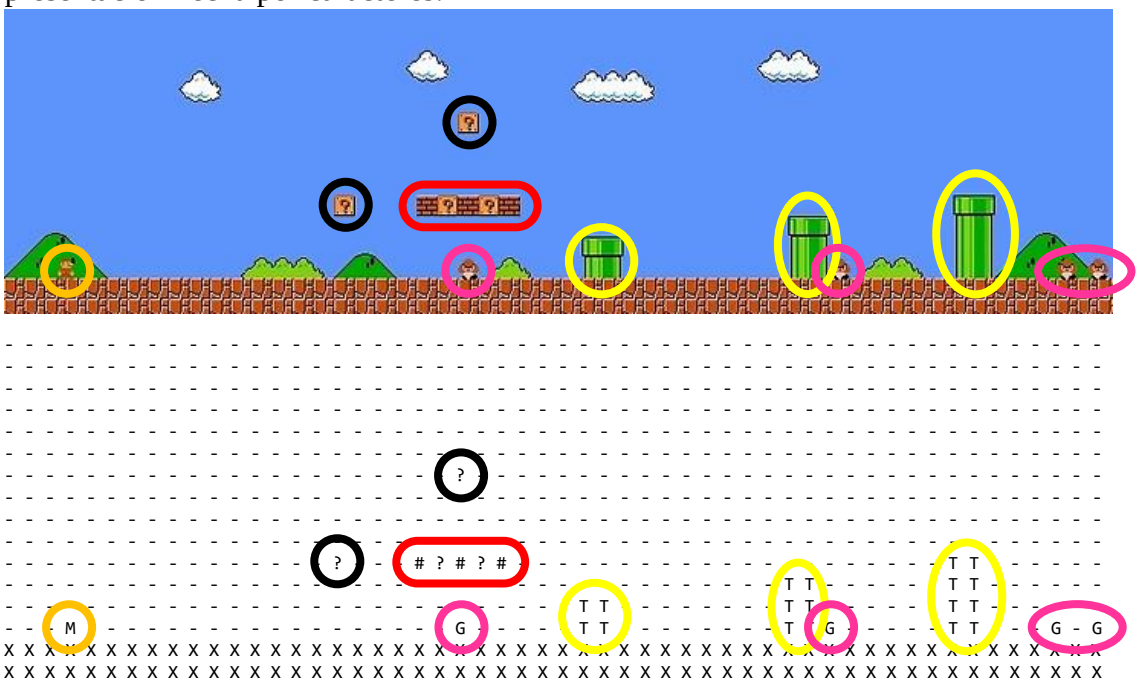
   Es posible representar un nivel de Super Mario Bros. por medio de caracteres alfanuméricos. Es decir, cada azulejo, se puede representar con un símbolo. Así, los azulejos del piso se pueden representar con el carácter X, los azulejos de los tubos se pueden representar con el carácter T, y así sucesivamente.

```
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - - ? - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - ? - - - # ? # ? # - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - T T - - - - - - T T - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - - - - - - - - T T - - - - - - - - T T - - - - - - T T - - - - - - - -
- - - M - - - - - - - - - - - - - - - - - G - - - - - - T T - - - - - - - - T T - G - - - - T T - - - - - G - G
X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X
X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X
```

Además, en los niveles de Super Mario Bros. hay personajes. El personaje principal es Mario, quien empieza el juego desde la posición (14, 4), es decir, en todos los niveles, Mario siempre empieza en la cuarta columna (contadas de izquierda a derecha), y en la fila 14 (contadas desde arriba hacia abajo).

En nuestra representación con caracteres alfanuméricos, la ubicación de Mario se representa con el carácter M. Siempre ubicado en la posición (14,4).

Otros personajes que aparecen en el juego son los enemigos de Mario. Existen diferentes tipos de enemigos. En este examen solo usaremos a un tipo de enemigo llamado Goomba Troopa. Sin embargo, este tipo de enemigo aparecerá en diferentes lugares del mapa. A continuación se muestra un comparativo entre el mapa original, y la representación hecha por caracteres:



Finalmente, el tercer y último recurso visual es el fondo. El fondo del mapa tiene nubes, arbustos, y montañas. Estos recursos no son parte del juego, y solo son decoración que se ve en el fondo del escenario. Por lo que no es necesario representarlos con ningún carácter.

Se le pide:
Elaborar un programa en lenguaje C++ en el que se solicite una cantidad variable de azulejos a usar a lo ancho de un nuevo nivel que será creado por usted.

Después de leer dicho ancho, se deberá crear un arreglo de 2 dimensiones, que permita almacenar todos los azulejos del nuevo nivel que será creado por usted. Su nuevo mapa tendrá 16 azulejos de altura y una cantidad variable de azulejos a lo ancho.

Los azulejos deberán ser creados al azar. Usted puede usar la técnica para generar números aleatorios de su preferencia. Sin embargo, esta asignación al azar debe respetar algunas reglas:

Reglas para el piso:

- Los azulejos de las dos filas de abajo representan el piso. El 90% de ellos debe ser X u el 10% de ellos debe ser _.
- Los azulejos de las dos filas de abajo representan el piso. La fila que está más abajo (fila 16) debe ser una copia idéntica de la penúltima fila (fila 15).

Estas dos reglar permiten crear el piso correctamente, es decir, con pequeños precipicios por los que el jugador podría caer.. Tal y como se muestra en la siguiente imagen:

Reglas para los tubos:

- Solo pueden haber tubos de altura 2, 3 y 4. Es decir, solo pueden haber azulejos T en las filas 11, 12, 13 y 14.
- Los azulejos de los tubos deben coincidir a lo largo de todo el tubo. Es decir, si alguna columna tiene un azulejo de tubo en la fila 13, debe tener también un azulejo de tubo en la fila 14. Si alguna columna tiene un azulejo de tubo en la fila 12, debe tener también un azulejo de tubo en las filas 13 y 14. Si alguna columna tiene un azulejo de tubo en la fila 11, debe tener también un azulejo de tubo en las filas 12, 13 y 14.
- No se preocupe por el ancho del tubo. En el videojuego, los tubos siempre tienen dos columnas de ancho, pero no es necesario que usted haga esto.

Reglas generales:

- Desde las filas 1 hasta la 14, se deben generar azulejos al azar. Estos azulejos estarán representados con los caracteres _, ?, # y T y deben estar distribuidos de la siguiente forma:
  - El 2% de los azulejos de una fila debe ser ?.
  - El 4% de los azulejos de una fila debe ser #.
  - El 2% de los azulejos de la fila 11, 12 o 13 deben ser T. Además, considerar que los azulejos de un tubo deben respetar las reglas de los tubos.
  - En la fila 14, el 1% de los azulejos debe tener el carácter G.
  - El resto de columnas debe ser llenado con el carácter _.

Imprimir su nivel en pantalla.

Los criterios en la rúbrica (y el puntaje respectivo) se condicionan a que la solución presentada corresponda al problema planteado. El 100\% corresponde al puntaje indicado en cada punto.

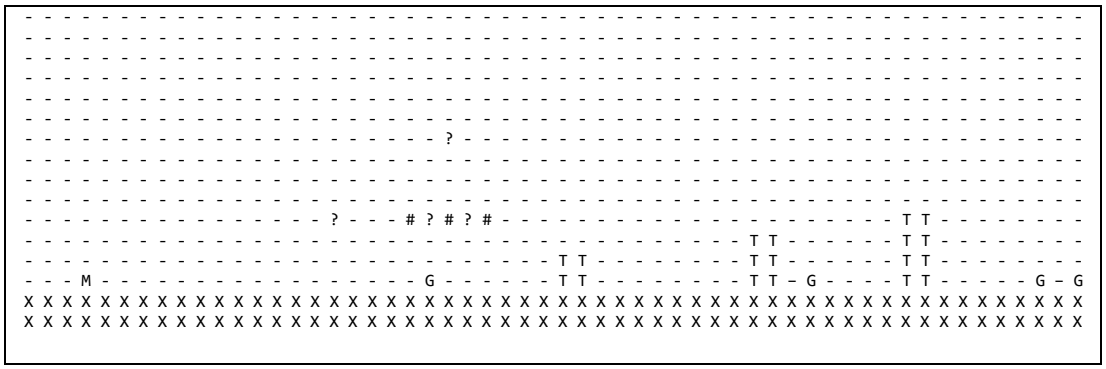
Criterio	Excelente	Adecuado	Mínimo	Insuficiente
Ejecución	El diseño del algoritmo es ordenado y claro, siguiendo buenas prácticas en programación. La ejecución es correcta (100%).	El diseño del algoritmo es ordenado y claro. La ejecución es correcta (70%).	El diseño tiene algunas deficiencias pero la ejecución es correcta (30%).	El diseño es deficiente y la ejecución no es correcta (0%).
Sintaxis	No existen errores sintácticos o de compilación (100%).	Existen algunos errores sintácticos de menor relevancia, que no afectan el resultado (50%).	Existen errores sintácticos en la forma de ejecución, que no afectan el resultado (30%).	El código tiene errores de sintáxis que afectan el resultado (10%).
Optimización	El código es óptimo y eficiente. De buen performance e interacción con el usuario (100%).	El código es de buen performance durante la ejecución (70%).	El código no está optimizado pero la ejecución no es deficiente (30%).	El código no está optimizado y la ejecución es deficiente (0%).

2. (6 puntos) Vectores:

Super Mario Bros. o Super Mario Brothers es un videojuego publicado en 1985 para la consola Nintendo Entertainment System. Cada nivel de dicho juego consiste en un mapa que tiene 16 casillas de altura y una cantidad variable de ancho. Algunos niveles son más largos que otros, pero todos tienen 16 casillas de altura.

Los niveles están conformados por diferentes piezas. A estas piezas se les conoce como baldosas o azulejos. En este examen usaremos el término «azulejo». A continuación, se muestra el primer nivel de Super Mario Bros.

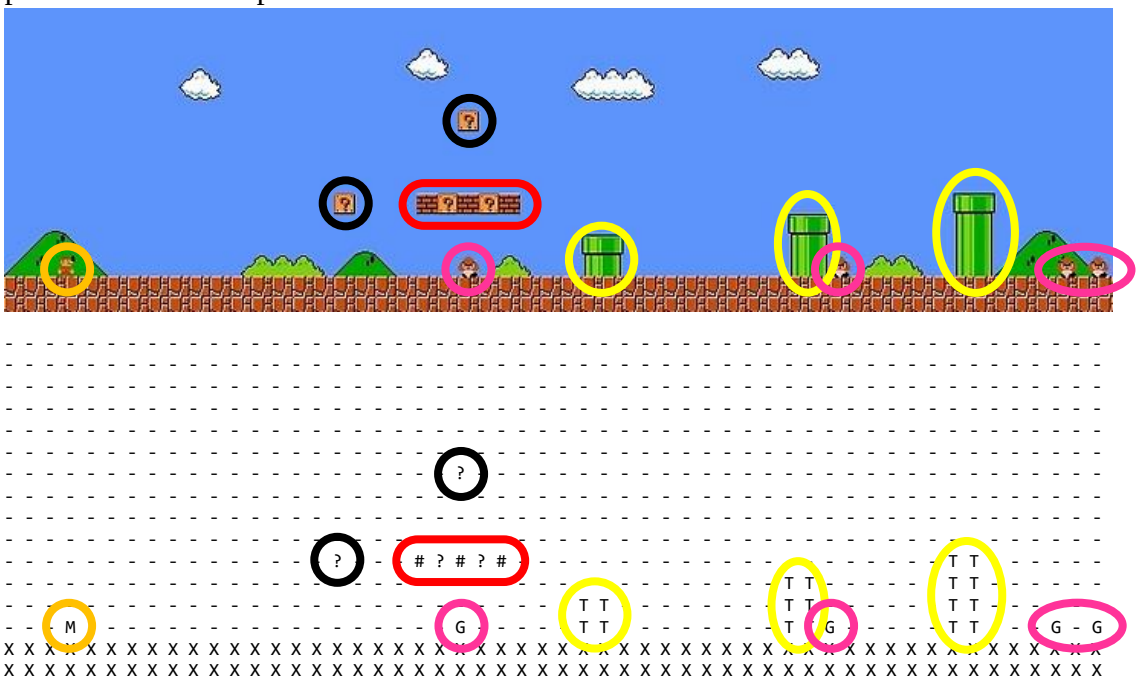
Es posible representar un nivel de Super Mario Bros. por medio de caracteres alfanuméricos. Es decir, cada azulejo, se puede representar con un símbolo. Así, los azulejos del piso se pueden representar con el carácter X, los azulejos de los tubos se pueden representar con el carácter T, y así sucesivamente.



Además, en los niveles de Super Mario Bros. hay personajes. El personaje principal es Mario, quien empieza el juego desde la posición (14, 4), es decir, en todos los niveles, Mario siempre empieza en la cuarta columna (contadas de izquierda a derecha), y en la fila 14 (contadas desde arriba hacia abajo).

En nuestra representación con caracteres alfanuméricos, la ubicación de Mario se representa con el carácter M. Siempre ubicado en la posición (14,4).

Otros personajes que aparecen en el juego son los enemigos de Mario. Existen diferentes tipos de enemigos. En este examen solo usaremos a un tipo de enemigo llamado Goomba Troopa. Sin embargo, este tipo de enemigo aparecerá en diferentes lugares del mapa. A continuación se muestra un comparativo entre el mapa original, y la representación hecha por caracteres:



Finalmente, el tercer y último recurso visual es el fondo. El fondo del mapa tiene nubes, arbustos, y montañas. Estos recursos no son parte del juego, y solo son decoración que se ve en el fondo del escenario. Por lo que no es necesario representarlos con ningún carácter.

Se le pide:
Elaborar un programa en lenguaje C++ en el que se solicite una cantidad variable de azulejos a usar a lo ancho de un nuevo nivel que será creado por usted.

Después de leer dicho ancho, se deberá crear un vector de vectores que represente las 2 dimensiones de su nivel, que permita almacenar todos los azulejos del nuevo nivel que será creado por usted. Su nuevo mapa tendrá 16 azulejos de altura y una cantidad variable de azulejos a lo ancho.

Los azulejos deberán ser creados al azar. Usted puede usar la técnica para generar números aleatorios de su preferencia. Sin embargo, esta asignación al azar debe respetar algunas reglas:

Reglas para el piso:

- Los azulejos de las dos filas de abajo representan el piso. El 90% de ellos debe ser X u el 10% de ellos debe ser _.
- Los azulejos de las dos filas de abajo representan el piso. La fila que está más abajo (fila 16) debe ser una copia idéntica de la penúltima fila (fila 15).

Estas dos reglar permiten crear el piso correctamente, es decir, con pequeños precipicios por los que el jugador podría caer.. Tal y como se muestra en la siguiente imagen:

Reglas para los tubos:

- Solo pueden haber tubos de altura 2, 3 y 4. Es decir, solo pueden haber azulejos T en las filas 11, 12, 13 y 14.
- Los azulejos de los tubos deben coincidir a lo largo de todo el tubo. Es decir, si alguna columna tiene un azulejo de tubo en la fila 13, debe tener también un azulejo de tubo en la fila 14. Si alguna columna tiene un azulejo de tubo en la fila 12, debe tener también un azulejo de tubo en las filas 13 y 14. Si alguna columna tiene un azulejo de tubo en la fila 11, debe tener también un azulejo de tubo en las filas 12, 13 y 14.
- No se preocupe por el ancho del tubo. En el videojuego, los tubos siempre tienen dos columnas de ancho, pero no es necesario que usted haga esto.

Reglas generales:

- Desde las filas 1 hasta la 14, se deben generar azulejos al azar. Estos azulejos estarán representados con los caracteres _, ?, # y T y deben estar distribuidos de la siguiente forma:

- El 2% de los azulejos de una fila debe ser ?.
- El 4% de los azulejos de una fila debe ser #.
- El 2% de los azulejos de la fila 11, 12 o 13 deben ser T. Además, considerar que los azulejos de un tubo deben respetar las reglas de los tubos.
- En la fila 14, el 1% de los azulejos debe tener el carácter G.
- El resto de columnas debe ser llenado con el carácter _.

Imprimir su nivel en pantalla.

Los criterios en la rúbrica (y el puntaje respectivo) se condicionan a que la solución presentada corresponda al problema planteado. El 100\% corresponde al puntaje indicado en cada punto.

Criterio	Excelente	Adecuado	Mínimo	Insuficiente
Ejecución	El diseño del algoritmo es ordenado y claro, siguiendo buenas prácticas en programación. La ejecución es correcta (100%).	El diseño del algoritmo es ordenado y claro. La ejecución es correcta (70%).	El diseño tiene algunas deficiencias pero la ejecución es correcta (30%).	El diseño es deficiente y la ejecución no es correcta (0%).
Sintaxis	No existen errores sintácticos o de compilación (100%).	Existen algunos errores sintácticos de menor relevancia, que no afectan el resultado (50%).	Existen errores sintácticos en la forma de ejecución, que no afectan el resultado (30%).	El código tiene errores de sintáxis que afectan el resultado (10%).
Optimización	El código es óptimo y eficiente. De buen performance e interacción con el usuario (100%).	El código es de buen performance durante la ejecución (70%).	El código no está optimizado pero la ejecución no es deficiente (30%).	El código no está optimizado y la ejecución es deficiente (0%).

3. (7 puntos) Objetos:

Crear un programa en lenguaje C++ que represente el juego de las sillas. Este es un juego en el que hay los jugadores dan vueltas alrededor de sillas mientras suena la música. Hay 1 silla menos que la cantidad de jugadores. De manera aleatoria, la música se detiene. Si eso ocurre, cada jugador se sienta en una de las sillas y el jugador que se queda parado es eliminado.

Usted deberá implementar un programa en lenguaje C++ que contenga una clase CJugador y solicitar por teclado el número de jugadores que van a jugar. Luego de eso, usted deberá crear un conjunto de jugadores (puede usar arreglos, vectores, listas u objetos dinámicos).

La clase CJugador deberá tener como mínimo los siguientes 2 atributos:

- Un string con el nombre del jugador
- Un int que indique en qué silla está el jugador (la silla 0 representará al jugador que no tiene una silla asignada, el jugador en la silla 0 será eliminado en cada ronda).

Su programa deberá tener una estructura iterativa infinita que termine cuando solo quede 1 jugador en el juego. En cada iteración, se decidirá al azar si la música sigue sonando o no.

La probabilidad de que la música pare es del 10% en cada iteración.

- Si en una iteración la música no para, su programa deberá imprimir el mensaje «La música sigue sonando» y reacomodar las sillas asignadas a cada jugador. Es decir, la silla 0 pasa al jugador número 2, la silla 1 pasa al jugador número 3, y así, hasta que la última silla pase al jugador número 1.
- Si en una iteración la música para, se deberá imprimir el mensaje «La música se detuvo», eliminar al jugador que tiene la silla 0, y quitar una silla del juego (usted puede quitar la primera o la última silla, según usted desee, y asignarle el 0 al jugador que tenía dicha silla).

Los criterios en la rúbrica (y el puntaje respectivo) se condicionan a que la solución presentada corresponda al problema planteado. El 100\% corresponde al puntaje indicado en cada punto.

Criterio	Excelente	Adecuado	Mínimo	Insuficiente
Ejecución	El diseño del algoritmo es ordenado y claro, siguiendo buenas prácticas en programación. La ejecución es correcta (100%).	El diseño del algoritmo es ordenado y claro. La ejecución es correcta (70%).	El diseño tiene algunas deficiencias pero la ejecución es correcta (30%).	El diseño es deficiente y la ejecución no es correcta (0%).
Sintaxis	No existen errores sintácticos o de compilación (100%).	Existen algunos errores sintácticos de menor relevancia, que no afectan el resultado (50%).	Existen errores sintácticos en la forma de ejecución, que no afectan el resultado (30%).	El código tiene errores de sintáxis que afectan el resultado (10%).
Optimización	El código es óptimo y eficiente. De buen performance e interacción con el usuario (100%).	El código es de buen performance durante la ejecución (70%).	El código no está optimizado pero la ejecución no es deficiente (30%).	El código no está optimizado y la ejecución es deficiente (0%).