UNIVERSIDAD DE COSTA RICA

Escuela de Ciencias de la Computación e Informática

Proyecto Programado: Segunda Entrega

Curso:	Programación I
Siglas:	CI-0112
Ciclo:	II-2024
Profesor:	Jonathan Esquivel Montoya
Fecha de Entrega:	Jueves 28 de Noviembre
Porcentaje de Evaluación:	15%

Creación de un Rogue-Like

Este proyecto se centra en el diseño e implementación de un juego roguelike en Java, explorando conceptos fundamentales de la programación orientada a objetos, estructuras de datos y algoritmos. El juego presenta la generación procedural de mazmorras utilizando matrices para representar habitaciones y un árbol binario para interconectarlas. El jugador navega por estas habitaciones, enfrentándose a enemigos en un sistema de batalla por turnos basado en agentes.

Estructuras:

Mazmorra: Representada por un conjunto de nodos doblemente enlazados. Nodo debe contener:

- Conexión a la habitación de arriba, abajo, izquierda y derecha
- Definición de la habitación: cuarto normal, cuarto de jefe, cuarto de salida
- Definición de si la habitación ya ha sido visitada o no
- Una habitación asignada

Los nodos de las mazmorras tendrán una habitación generada al pasar por el nodo una primera vez. Una vez generado el nodo, la habitación deberá guardar su estado, el cual solo se modificará si el jugador vuelve a entrar a esta de nuevo.

La mazmorra deberá poder imprimir un mapa actual del juego, con solo nodos vecinos:

```
N
|
[1]-[2]-[S]
|
[B]
```

Donde cada línea identifica un camino encontrado entre cuartos, el número indica el orden en el que se entró a ese cuarto normal. La S el cuarto con la salida, la B el cuarto con el jefe y N si no hay cuarto.

El nodo activo de una mazmorra deberá:

1. Tener una conexión con el nodo de donde fue generado.
2. Una vez que el jugador esté en el nodo, creará inicialmente la matriz y definirá aleatoriamente de 2 a 4 puertas:
    a. 100% de probabilidad de generar la puerta de conexión con el nodo anterior.
    b. 60% de probabilidad de crear una puerta nueva a un cuarto.
    c. 40% de probabilidad de crear una puerta a un segundo cuarto.
    d. 20% de probabilidad de crear una cuarta puerta conectada a otro cuarto.
3. Las puertas generadas en las probabilidades anteriores, crearán los nodos inmediatamente junto a sus conexiones, pero no creará el cuarto hasta que el jugador entre a alguna de estas habitaciones nuevas.

Las conexiones pueden ser a cuartos ya existentes, a pensar de que inicialmente un cuarto no conectó desde el lado contrario. Los cuartos estarán basados en una maya de nodos. Pueden verlo como una matriz, pero de nodos que se van generando si entra o no a un lugar:

```
[3]-[2]-[4]
 |   |   |
[S]-[5]-[1]-[6]-N
 |   |   |
[7]-[8]-[B]
```

Habitación: Representada por una matriz bidimensional de enteros. Cada celda puede contener:

- Paredes: el jugador no puede pasar por las paredes que estén definidas en la matriz.
    - El cuarto deberá poseer paredes alrededor de los bordes, y un vacío si tiene una puerta en esa entrada, excepto en las esquinas. Cada puerta estará en uno de los bordes que conecte con el cuarto correspondiente.
- Suelo: se representa con la casilla vacía/en blanco.
- Puerta: las puertas se representan con un espacio vacío en la pared.
    - Cuando el jugador se coloca en uno de esos espacios vacíos, puede moverse una vez más hacia el lado de la apertura para entrar al siguiente nodo.
    - El jugador entrará al siguiente nodo, justo en la apertura de la puerta correspondiente en ese otro nodo.

- Enemigo: Los enemigos serán mostrados con una E en la habitación, más adelante se hablará de cómo funcionan.
- Arma: el jugador podrá encontrar armas en las habitaciones que potencien su ataque.
- Item: los agentes que pasen encima de un ítem, adquieren el buff/debuff de este.

Las habitaciones deben de ser generadas proceduralmente una única vez al entrar en estas y ser mostradas en pantalla con una representación acorde a lo que puede o no tener. El jugador debe poder interactuar con los elementos de la habitación al pasar por encima o al lado de estos (dependiendo de que sea).

Una habitación tiene una probabilidad inicial del 10% de ser la habitación del jefe y un 5% de ser la habitación de salida. Ambas probabilidades aumentarán en la creación de cada nueva habitación en un 7% y un 5% respectivamente hasta que genere la habitación.

Agentes: Esta es la clase base para el jugador y los enemigos. Deben poseer las siguientes propiedades:

- Salud: cantidad de vida actual que tiene el agente.
- Ataque: Ataque base del agente.
- Defensa: Defensa base del agente.
- Arma: el arma solo funciona para modificar el ataque base del agente.
- Armadura: funciona para modificar la defensa base del agente.
- Buffs/Debuffs: algunos ítemes que se encontraran en la habitación podrán dar buffs o debuffs al agente.

Enemigos: Los enemigos deberán tener las mismas acciones que el jugador y tratarán de atacar al entrar en su habitación:

- El enemigo se moverá como el jugador.
- Si el enemigo pasa por un item, lo agrega a su inventario
- Si el enemigo pasa por un debuff, lo afectará como al jugador.
- Los ítems los usará en combate únicamente, y los usará al tener 50% de la vida o menos (si son de vida), o si son ítems para buffear, al iniciar el combate.
- Al derrotar un enemigo, si posee ítems, dejará en el lugar donde estaba uno aleatorio entre los que poseía.
- Si no tenía ítems, pude soltar uno aleatorio con una probabilidad del 15%.
- El enemigo tendrá en cada turno, una probabilidad del 75% de moverse hacia el jugador.

Sistema de Batalla: El jugador y los enemigos se turnan para realizar acciones de atacar, defender o utilizar un ítem que posea. El jugador puede poseer más de un ítem, tendrá un vector de éstos y cuando decide utilizar uno, imprime la lista de ítems que tiene, junto con una breve descripción.

- Los agentes deben poder atacar en un sistema similar al de Jugador vs Monstruos. El jugador y enemigo pueden seleccionar una acción de Defender, para aumentar sus defensas y curarse, en vez de atacar. Además puede utilizar ítems, los cuales cuentan como una acción.
- El enemigo en combate siempre preferirá atacar, pero habrá una posibilidad del 25% de defenderse si posee 50% o menos de vida.
- Tanto los buffs como debuffs de los ítems aplican al ataque/defensa base de los agentes.
- **Cálculo de daño:** Se basa en las propiedades de ataque y defensa de los agentes, y en las características del arma y armadura equipadas.
    - (Daño base + extra de arma) - armadura del enemigo.
- **Buffs/Debuffs:** Pueden afectar temporalmente las propiedades de los agentes.

Armas y Armaduras: Las armas y armaduras podrán caer de los enemigos una única vez:

- Arma básica: aumenta el daño en un 20%. Probabilidad de salir luego de encuentro: 30%.
- Arma secreta: esta ataca porcentualmente, reduciendo la vida en un 50% de la vida actual por cada ataque, agregado al ataque base del agente (Sí, cada vez pegará menos que la anterior). Probabilidad de salir luego de encuentro: 15%.
- Arma legendaria: el ataque base se multiplica por 2. Probabilidad de salir luego de encuentro: 5%.
- Armadura básica: reduce el daño en un 20%. Se pierde un 2% con cada golpe recibido. Probabilidad de salir luego de encuentro: 15%.
- Armadura secreta: funciona igual que la anterior, pero reduce un 30% y luego de una batalla recupera todos los puntos de armadura. Probabilidad de salir luego del encuentro: 10%.
- Armadura legendaria: reduce todo ataque en 50%. Probabilidad de salir luego de encuentro: 5%.
- Cualquier armadura saldrá solo si no ha salido anteriormente o una de mayor nivel.

Buffs y Debuffs: Ítems que afectan positiva o negativamente al jugador o enemigo:

- Buff de aumento de ataque: aumenta el ataque base entre un 10% a un 20%. Acumulativo.
- Buff de aumento de defensa: aumenta la defensa en un 15%. Acumulativo.
- Buff de sangre: Por cada golpe, sube un 20% del daño final realizado a la vida en la batalla actual.

- Debuff de defensa: reduce la defensa en un 15%.
- Debuff de envenenar: por cada turno en batalla, pierde de 1, incrementando a 10 puntos de vida. Luego de eso sigue perdiendo 10 puntos por turno. El envenenamiento dura 20 turnos, o hasta tomar un buff de sangre.
- Debuff de ataque: reduce el ataque en un 20%.

Generación Procedural de habitaciones: El tamaño y la forma de la habitación se definen aleatoriamente, en un mínimo de 8x8 a un máximo de 16x16. Las habitaciones siempre tienen paredes alrededor del perímetro, y algunas generadas dentro de la habitación.

- Las paredes de los bordes se definen anteriormente.
- Además, se deben colocar aleatoriamente en el mapa de 2 a 5:
    - Las paredes tienen de longitud 2 a 5 respectivamente.
    - No pueden cruzarse, son lo primero a colocar en la habitación.
- Hay una probabilidad de 75% de que haya 1 enemigo, 25% que hayan 2 enemigos, 10% de que hayan 3 enemigos y 2% de que hayan 4.
- Hay una probabilidad de un 17% de que haya un ítem aleatorio en la habitación.
- Hay una probabilidad de 2% de que haya una armadura o arma, siguiendo las probabilidades de estas. (Si se da el 2%, se calcula de menor a mayor si sale una u otra).
- Los ítems y otros elementos, es más común que los suelte un enemigo al ser derrotado.
- Recordar seguir las reglas de generación de nodos.
- Una habitación solo es generada al entrar en el nodo, no antes.

El Jefe:

Será un enemigo en un cuarto vacío, solo existirán las paredes de los bordes. El jefe se defenderá con una probabilidad del 40% y atacará con una probabilidad del 60%. Cuando la vida empieza a bajar, las probabilidades se invierten. Cuando tenga el 60% de la vida, seguirá igual, pero al ir bajando, se invierten de manera que al tener 40% de la vida, se defenderá con un 60% y atacará con un 40%.

Flujo del Juego:

Al derrotar al jefe, este dará la llave de salida de la mazmorra, la cual permitirá abrir la puerta de la habitación de salida del juego.

El juego termina al llegar a una puerta de completar el nivel, o no tener puntos de vida.

Visualización:

La habitación se mostrará al jugador en la consola utilizando caracteres para representar los diferentes elementos (tentativo, pueden seleccionar otros caracteres):

- #: Pared
- : Suelo
- # #: Puerta (recuerden que es un símbolo diferente por puerta)
- E: Enemigo
- A: Arma
- R: Armadura
- I: Item
- @: Jugador

Teniendo por ejemplo una matriz pequeña representando una habitación de la mazmorra:

```
#### ####  Ítems……..[A][B][ - ]
# A        #  Vida:.....100
#       E  #  Armadura…20%
#@
#    +     #  Ataque…….40
### #####  Efectos……[Envenenado]
```

Tomen en cuenta que para cada movimiento, deben mostrar la matriz actualizada. Así como cuando inicia una batalla, no deben mostrar la matriz hasta que esta ha sido finalizada. También que el jugador puede optar por mostrar el mapa cuando esté en una habitación.

Entregables

Los proyectos deben ser subidos a Mediación Virtual. Cada equipo debe subir ahí los documentos de la entrega en la fecha indicada. Las clases deben de estar documentadas internamente y la principal con los nombres de los integrantes.

Rúbrica

Aspectos de evaluación	Porcentaje
Puntualidad y formalidad de la entrega	5%
Creación de clases	10%
Desarrollo de las clases y acciones	60%
Ciclo de juego completo	20%
Documentación	5%