Proyecto 2

Raúl Nuño Valdés, 317204580

18 de enero de 2021

Problema

Se te pide hacer un juego basado en texto en el que el usuario puede escribir las acciones a realizar cada que se le dé la opción. En específico habrá el comando .ªyuda"que mostrara los posibles comandos que se puede usar para indentificar una oración, pero se podrán usar algunos sinónimos de estos, si los hay. (ayuda==comando o ir==dirigirse==caminar, etc.)

Además, debe tene una interfaz gráfica, aunque no tiene que ser muy sofisticada.

De igual forma, el personaje debe de poder tener objetos en su inventario, sin embargo tendra un espacio redicido para cada tipo de objetos que vaya a conseguir, y estos servirán para que este ejecute ciertas acciones que no podría si no los tuviese; se cure la vida, incrementar su daño, abrir cerraduras o simplemente leerlos; en caso específico para los objetos que le curen, podrán ser combinados para incrementar o reducir sus efectos curativos y en caso de las llaves, habrá frgamentos de estas y los podrá combinar para formar una llave completa.

También habrá un enemigo el cual debera poder atacar al jugador o buscarle para luego atacarle cada cierto tiempo establecido (treinta segundos) que comenzará a correr cuando se inicie el juego; el juego puede pausarse, y ya que sería injusto que el enemigo ataque en esos momentos, el tiempo también deberá hacerelo. No se pausara mientras el jugador este leyendo. También va a atacar al jugador si este entra a un cuarto donde ya estaba el enemigo y el jugador no ssale por la puerta por la que entró o si el enemigo entra al

cuarto del jugador y este intenta pasar por la dirección por la que el enemigo ingresó.

Este juego se ambientará en una casa, por lo que se debe distinguir en que ubicación de esta se está.

El juego termina si el jugador muere o el enemigo es atacado con una espada oxidada (herramienta encontrable en el juego).

Características

Se planea que el juego cuente con una pantalla de **inicio**, una de **en juego** y otra de **en pausa**.

En inicio debe dar opciónes de salir del juego o iniciar nueva partida. En juego debe de poder pausarse, poder escribir la acción y es en este en el que se contará la historia. En pausa debe de dar las opciones de volver al menú de inicio o vover a la pantalla de en juego, si se va al menú de inicio, se reinicia el progreso hecho.

El **jugador** y el **enemigo** tendrán los atributos de ubicacion, ubicacion anterior, vida, vida maxima y ataque. Para el caso del jugador también deberá tener un inventario para guardar los **objetos** que vaya obteniendo y poder manipularlos; también puede abrir muebles que son objetos que se encontraran en los diferentes cuartos, para depués tomar lo que tengan. Ambos podrán curarse, atacar o moverse a una nueva ubicaión.

El inventario del jugador será el siguente: 10 notas, 6 consumibles, 6 llaves y 2 herramientas. No podra tomar más de un tipo si se llena el inventario específico. Algunos objetos que puedan ser combinados crearán uno nuevo para guardarlo y desaparecen los objetos usados: esto libera un espacio. También hay algunos que pueden ser consumidos como los consumibles al ser comidos o las llaves al ser usadas para oabrir un mueble, liberando un espacio en sus respectivos inventarios.

Los objetos serán clasificados en distintas categorias, como **llaves**, **herramientas**, **consumubles** y **notas**.

Las llaves servirán para abrir muebles que no se puedan abrir sin una; las herramientas son equipables para incrementar el daño del jugador, los consumibles son para recuperar vida perdida por el jugador y las notas son objetos

que contienen un escrito que podrá ser leído.

Para recibir las acciones posibles, se debe contar, con un diccionario de **comandos** (verbos en infinitivo o palabras clave). Estos se complementaran con una cadena que que indicará a qué cosa se le efectuará la acción y, de ser necesario, con qué objeto se realiza; en qué dirección; aunque algunos comandos no requerirán de ningún complemento. Se debe tratar de que el complemento sea lo más natural posible, como si se tratase de una oracion imperativa. debe de ser un poco flexible, de forma que si tuvo un error como poner o no acentos combinar mayúsculas con minúsculas no deben impedir que se «entienda».

Ejemplos: $\underbrace{\text{Comandos}}_{\text{Comando}}$, $\underbrace{\text{Abrir}}_{\text{Comando}}$ $\underbrace{\text{mueble}}_{\text{Qué cosa}}$, $\underbrace{\text{Abrir}}_{\text{Comando}}$ $\underbrace{\text{cajonera de roble}}_{\text{Qué cosa}}$ $\underbrace{\text{con}}_{\text{Resto}}$ $\underbrace{\text{llave roja}}_{\text{Con qué cosa}}$,

$\underbrace{\text{Ir}}_{\text{Comando}}$ $\underbrace{\text{hacia el}}_{\text{Resto}}$ $\underbrace{\text{norte}}_{\text{Dirección}}$

Patrones a usar y porqué

Se va a usar el patrón `MVC` para las distintas pantallas (inicio, en juego y en pausa) y la interacción con sus diferentes botones que darán información al juego.

El patrón `Template` para las distintas acciones puesto que, aunque cada unar hace distintas ejecucuiones que involucran al jugador y al enemigo, solo al jugador o a ninguno, todas van a analizar la cadena posterior al comando y luego van a realizar las acciones correspondientes.

Para el reloj y el juego usarán `Observador` puesto que el enemigo debe moverse cada cierto tiempo. El juego observará al reloj y este le notificará que debe hacer que el enemigo actue.

El patrón `Contructor` para poder combinar los objetos curativos y llaves ya que tendrán una receta y van a partir de un objeto ya combinado con uno sin combinar o dos objetos sin combinar. Puesto que ambos son items, pero no son lo mismo, un contructor será para las llaves y otro para los consumibles. El director de los contructores será usado en el jugador.

El patrón `Iterador` se usará puesto que, aunque se usan las mismas estructuras (ArrayList). Recordemos que se pide un solo inventario para el jugador,

así que se trata de in ArrayList con cuatro ArrayList de un objeto específico. Con Iterator podremos recorrer este gran inventario de una sola vez o recorrer un inventario en específico.

Relación de las clases con los patrones

Las clases en la carperta `/mvc` son las partes importantes para el patrón MVC. Las clases Ventana serán la vista, las clase juego, lógica, lógica enemigo y todas las clases en las carpeta `juego` serán el modelo y la clase gestor ventana será el controlador.

Las clases juego y reloj, como ya se dijo, se relacionan como observador y observable respectivamete.

Todas las clases en la carperta `/juego/comandosValidos` usan el patrón Template, siendo la clase accion defecto la clase con el «esqueleto» para las demás.

La clase director en la carpeta `/juego/ente/jugador` es el director para las clases en la carpeta `/juego/item/contructor` donde la clase contructor es una interfaz para implementar en las otras dos calses.

En la carpeta `/juego/inventario` se encontrará la clase inventario que permite crear un lugar donde guardar objetos de tipo T que extiendan a Item con un límite que tendrá su iterador, mientras que la clase inventario general solo tendrá cuatro inventarios de los respectivos items con sus respectivos límites. Esta también tendrá su iterador el cual recorre los cuatro inventarios como uno solo. También puede obetener los iteradores de un solo inventario de ser necesario.

Como ejecutar

Use `javac -d bin -cp src src/principal/Principal.java` para compilar y `java -cp bin principal/Principal` para ejecutar.

Notas

Se trabajó en Widows donde sí ejecuta; se trató de evitar que hubiese problemas al compilar y ejecutar en Linux dado que lee ciertos recursos para iniciar los cuartos, items, comandos y muebles, pero no hubo forma de compobar su correcto funcionamiento.

Se listarán algunos posibles comandos con sus respectivas cadenas, en caso de ser necesarias, aunque se trato de abarcar la mayor cantidad de posibiladades al escribir

1. ayuda
2. inventario
3. inventario de {uno de los cuatro tipos de objetos en plural}
4. vida
5. regresar al cuarto anterior
6. ir al {un punto cardinal}
7. comer {nombre completo de un consumible}
8. leer {nombre completo de una nota}
9. observar cuarto
10. observar {nombre de un mueble}
11. equipar {nombre completo de una herramienta}
12. pelear
13. agarrar {nombre completo de un item en un mueble}
14. abrir {nombre de un mueble}
15. abrir {nombre de un mueble} con {nombre de una llave en el inventario}
16. buscar salidas
17. combinar {nombre completo de un item} con {nombre completo de un item}

Notese que los que indican nombre completo debe ser escrito tal cual, muentras que los que dicen nombre, se buscará solo el item que tenga el nomnre mas similar.