

## Manual Técnico Juego Peleas

### Class Juego Pelas

-metodo main

llamar al motor de juego, creando una instancia de motor

### Class MotorDeJuego

-menuPrincipal

\*\*\*\*Bienvenido a Juego Peleas\*\*\*\*

Elije una una accion...

[1] Iniciar Partida

[2] Tienda

[3] Generar tablero

[4] Inventario

[5] Salir

caso1:

llamar a metodo iniciarPartida

caso2:

llamar a metodo comprarArticulos

caso3:

llama a metodo generarTablero

caso4:

llama a metodo pintarInventario

caso5:

matar el programa

-iniciarPartida

llamar a metodo iniciarInventarioJugador-IA(para que no sea nulo)

llamar a metodo comprarPersonajesJugables(2 veces)

llamar a metodo seleccionEnemigos(para obtener a los personajes enemigos)

llamar a metodo pintarInventarioUsuario-IA(para mostrar al usuario los personajes de la partida)

llamar a metodo elegirPersonajePrincipal(el que estara en el mapa y podra usar el usuario)

llamar a metodo seleccionarEnemigos(para que se asignen al tablero)

llamar a partida(para iniciar la partida)

-seleccionarPersonajePrincipal

pedir un numero de indice, para que el seleccionado sea el personaje principal

caso1:

el personaje en la posicion 1 sera el principal

caso2:

el personaje en la posicion 2 sera el principal

caso3:

el personaje en la posicion 3 sera el principal

caso4:

el personaje en la posicion 4 sera el principal

caso5:

el personaje en la posicion 5 sera el principal

-seleccionarEnemigos

recorrer el inventario de enemigos

asignar la posicion i en una var tipo enemigos a cada uno de los 5 enemigos

-comprarJugables

mostrar a los 5 personajes disponibles en la tienda

verificar que se tenga el oro

comprar personaje en la posicion indicada por el usuario

asignarlo en el inventario del usuario, a un espacio disponible

-seleccionEnemigos

elige a los enemigos en funcion de la dificultad de juego indicada por el usuario

dificultad=1

2 enemigos

dificultad=2

3 enemigos

dificultad=1

5 enemigos

para ingresarlos al tablero de juego

Class Partida

-iniciarPartida

definir al personaje que iniciara el turno

mostrar la eleccion del tablero a jugar

verificar que los personajes del usuario y de la IA esten vivos

mientras los personajes esten vivos

ejecutar el turno correspondiente a usuario-jugador

si es turno del usuario

preguntar que accion desea realizar

1 mover al principal

2 atacar a un enemigo

sino

realizar las acciones automaticamente de la IA  
con numeros aleatorios

si los personajes no viven

finalizar la partida y mostrar al ganador

volver al menuPrincipal

-definirPersonajeInicial

con un numero random y la posibilidad 50/50 que el usuario inicie el turno

-mostrarOpcionesMovimientoUsuario-IA

le muestra las 4 opciones de movimiento al jugador y a la IA

1 norte

2 sur

3 este

4 oeste

-mostrarOpcionesAtaqueUsuario-IA

le muestra las 4 opciones de ataque al jugador y a la IA

1 norte

2 sur

3 este

4 oeste

-mostrarEleccionMapa

le muestra al usuario los 3 posibles tableros para jugar

1 mapa 8\*8

2 mapa 10\*10

3 mapa personalizado (n\*m)

m y n son ingresados por el usuario

-ganoJugador

si los personajes de la IA estan muertos

indicar al usuario que gano

caso contrario

indicar al enemigo que perdio

-recorrerInventarioUsuario-IA

verifica en todas las posicion de los inventarios si hay personajes con vida

Class Usuario extends Jugador

-iniciarInventarioUsuario

recorre el inventario del usuario y lo llena de personajes vacios

-pintaInventarioUsuario

pinta todos los objetos que hay en el inventario del usuario

-comprarJugables

personajes verifica que el usuario tenga el oro suficiente para que pueda comprar a los

si oro>=200

true

sino

false

Class IA extends Jugador

-iniciarInventarioIA

recorre el inventario de la IA y lo llena de personajes vacios

-pintaInventarioIA

pinta todos los objetos que hay en el inventario de la IA

-llenarInventarioIA

recibe la dificultad

llenar inventario de enemigos

Class Tablero

-iniciarTablero

recibir la eleccionMapa

caso1:

tablero8\*8

caso2:

tablero10\*10

caso3:

pedir al usuario que ingrese los valores n y m

tablerom\*n

recorrer las dimensiones de la matriz desde m-n=0, hasta m y m indicados por el usuario, y llenar las posiciones con casillaVacia

llamar a metodo generaraCasillas

llamar a metodo generarCasillas

-generarCasillas

recorrer el tablero, donde hayan casillasVacías poner un tipo de casilla generado aleatoriamente, tomando en cuenta los porcentajes de generacion

-generarEnemigos

recibir la dificultad

en funcion de la dificultad poner a los distintos enemigos en el tablero

-moverJugadorY

tomar la posicion del jugador

si se mueve para arriba

si la casilla siguiente es una planicie

modificar la posicion Y restandole el valor del movimiento del

personaje

si se mueve para abajo

si la casilla siguiente es una planicie

modificar la posicion Y sumandole el valor del movimiento del

personaje

-moverJugadorX

tomar la posicion del jugador

si se mueve para la derecha

si la casilla siguiente es una planicie

modificar la posicion X sumandole el valor del movimiento del

personaje

si se mueve para la izquierda

si la casilla siguiente es una planicie

modificar la posicion X restandole el valor del movimiento del

personaje

-ataqueJugadorY

si el ataque es para arriba

si la casilla siguiente es un enemigo

realizar el ataque dirigido al enemigo

modificar los valores de puntosVida del enemigo

si el ataque es para abajo

si la casilla siguiente es un enemigo

realizar el ataque dirigido al enemigo

modificar los valores de puntosVida del enemigo

-ataqueJugadorX

si el ataque es para la derecha

si la casilla siguiente es un enemigo

realizar el ataque dirigido al enemigo

modificar los valores de puntosVida del enemigo

si el ataque es para la izquierda

si la casilla siguiente es un enemigo

realizar el ataque dirigido al enemigo

modificar los valores de puntosVida del enemigo

-moverEnemigosY

tomar la posicion del enemigo1

tomar la posicion del enemigo2

tomar la posicion del enemigo3

tomar la posicion del enemigo4



enemigo

enemigo

```
-moverEnemiesX
```

tomar la posicion del enemigo1

tomar la posicion del enemigo2

tomar la posicion del enemigo3

tomar la posicion del enemigo4

tomar la posicion del enemigo5

tomar al enemigo que se este moviendo y guardarlo en enemigo

si se mueve para la derecha

enemigo

modificar la posicion Y restandole el valor del movimiento del

si se mueve para la izquierda

si la casilla siguiente es una planicie  
modificar la posicion Y sumandole el valor del movimiento del  
enemigo

devolver al enemigo tomado, para actualizar sus posiciones

-ataqueEnemigosY

si el ataque es para la arriba

si la casilla siguiente es un jugador

realizar el ataque dirigido al jugador

modificar los valores de puntosVida del jugador

si el ataque es para la abajo

si la casilla siguiente es un jugador

realizar el ataque dirigido al jugador

modificar los valores de puntosVida del jugador

-ataqueEnemigosX

si el ataque es para la derecha

si la casilla siguiente es un jugador

realizar el ataque dirigido al jugador

modificar los valores de puntosVida del jugador

si el ataque es para la izquierda

si la casilla siguiente es un jugador

realizar el ataque dirigido al jugador

modificar los valores de puntosVida del jugador

-eliminarJugadoresEnemigos

verificar los puntos de vida de los enemigos

si los puntosDeVida >=0

eliminar al personajes de los enemigos del tablero

-pintarTablero

llamar a metodo mostrarEstadisticas

llamar a metodo eliminarJugadores Enemigos

imprimir el tablero de juego en la consola

-mostrarEstadisticas

obtener los puntos de vida de jugador-enemigos

obtener las posiciones de jugador-enemigos

Class Personajes

-estaVivo

si los puntosDeVida >0

retorna verdadero

-calcularDano

calcular probabilidad de ataque 1-100

si probabilidad ataque<=75

guardar el danoHecho, para mandarlo a el metodo recibir dano

sino

el danoHecho=0

-recibirDano

recibe danoHecho

si danoHecho>0

restar danoHecho de puntosVida del personaje que se este atacando

sino

el dano hecho es 0

-ataque

del personajeAtacado llama al metodo recibir ataque