

DOCUMENTACIÓN

PROYECTO DSA

2017 - OTOÑO

Miembros del grupo

- Albert Álvarez
- Jesús Ligeró
- Óscar Mampel
- Jaume Mateu

ÍNDICE

1. INTRODUCCIÓN	3
2. CLASES	3
3. BASE DE DATOS MySQL	4

1. INTRODUCCIÓN

Nuestro juego será una vista bidimensional tipo dungeon, basado en muchas dinámicas de Pokémon. Cada personaje tendrá una visión aérea del mismo, tendrá un inventario donde almacena objetos y una lista de monstruos aliados, que serán con los que luche contra otros monstruos salvajes que podrán ser capturados y añadidos a su lista de aliados. El mapa será parecido al de Pokémon, habrá caminos seguros y caminos donde puedan aparecer aleatoriamente enemigos y en el caso que se dé tope con un monstruo cambiará a "pantalla de combate". Ahí tenemos la divergencia con el juego Pokémon convencional ya que el modo de combate será hack and slash. Tendrás un monstruo aliado que verás de perfil, al igual que el enemigo, y podrás desplazarte hacia adelante, atrás, saltar y atacar (ataque normal y ataque cargado).

Los combates contra un monstruo salvaje serán contra la inteligencia del juego, mientras que los combates contra los entrenadores podrán ser contra la inteligencia del juego o jugadores online.

2. CLASES

El juego se diseñará con una estructura de clases heredadas según los 5 tipos de clases principales o padres (Personaje, Monstruo, Ataque, Objeto e Inventario).

La clase Personaje (padre) contiene los campos:

String	nombre
String	escenario
int	x, y
Inventario	inventario
Lista_Monstruos	lista_monstruos
bool	hombre/mujer

Esta clase la hereda Usuario con un e-mail (String) y password (password), también hereda Enemigo que contendrá una función para hacer ataques aleatorios (inteligencia aleatoria).

En segundo lugar la clase Monstruo que incluirá:

String	nombre
int	nivel
int	experiencia
int	vida
int	ataque
int	defensa
Lista_Atques	lista_ataques

En tercer lugar la clase Ataque dispondrá de un nombre (String) y un multiplicador (int).

La clase Objeto dispone de los atributos nombre (String) y descripción (String) que heredarán todos los distintos objetos que implementemos en el juego tales como pociones, potenciadores... con los atributos que necesiten para estar operativas.

Habrà una subclase, por cada tipo de objeto diferente ej: pokeballs pociones, potenciadores..

En último lugar disponemos la clase Inventario que es una lista de Objetos.

3. BASE DE DATOS MySQL

El juego tendrá 6 tablas de momento en mySQL las cuales corresponden a Usuario, Enemigo, Mapa, Celdas, Monstruo y Objetos.

Las tablas de Usuario y Enemigo aunque hereden de personaje se dividen en 2 tablas para poder facilitar más adelante el código y evitar tratar con nulls en las tablas. Estas tablas contendrán la información compartida que se encuentra en la clase Personaje mas la información específica correspondiente a cada una de las clases de Usuario y Enemigo. La tabla Monstruo contendrá los ataques de todos los monstruos que se puedan poseer, junto con sus stats para el combate. La tabla Mapa tendrá la información de donde se encuentran las puertas, el array de celdas de cada escenario y finalmente el nombre del mismo.

La tabla Celdas contendrá la información de cada celda que contendrá la información de que tipo son, de que mapa y a que coordenadas pertenecen.

si se puede andar por encima, actuar con ella (por ejemplo si hay una papelera lanzar cosas) y qué tipo de celda es si es hierba, hielo, tierra, muro, etc.

Finalmente la tabla Objetos contendrá todos los objetos existentes en el mundo con sus stats, tipo y a que usuario corresponde.

4. Mundo

El mundo es una matriz 2D de objetos celda donde cada celda especifica los campos: tipo de celda, la probabilidad con la que aparezcan monstruos monstruos, si los personajes pueden pasar por encima y finalmente la probabilidad de que aparezca un objeto.

Cada celda es una clase virtual con dos metodos vacios.Son accionEncima que se efectua cuando estas encima de una celda, y accionActivar que se activa cuando estas al lado encarado hacia ella y le das al botón “a”(Ej: cuando buscas en las papeleras de pokemon)

Aparte de las funciones misceláneas de arriba habrá otro método llamado accion en el que se hará un random en el que se decidira si se activan las funciones comenzarCombate y obtenerObjeto.

Los diferentes tipos de terreno y paredes serán clases heredadas de celda, por ejemplo tendremos la clase terrenoHierba, terrenoTierra, terrenoCamino, terrenoPared etz.

Un caso especial será la casilla de cambio de mapa, donde la celda correspondiente tiene el metodo accionEncima sobreescrito.

Celda
-String tipo
-float probObj
-float probComb
-bool andable
Personaje personajeEncimaCelda
accionEncima
accionActivar
accion
obtenerObjeto
comenzarCombate