Lo primero que necesitamos es el UML del juego, que será muy básico.

Diagrama

El contenido generado por IA puede ser incorrecto.

Ahora, la estructura del proyecto:

Captura de pantalla de un celular

El contenido generado por IA puede ser incorrecto.

Y, por último, el código de cada archivo:

Character.h:

#pragma once

#include <iostream>

#include <string>

using namespace std;

class Character {

protected:

string name;

int hp;

int damage;

public:

Character(string name, int hp, int damage);

virtual int attack(); // Método virtual para polimorfismo

void takeDamage(int dmg);

string getStatus();

bool isAlive();

};

Boss.h:

#include "character.h"

#include <iostream>

using namespace std;

class Boss : public Character {

private:

int danoUlti;

public:

Boss(string name, int hp, int damage, int danoUlti);

int finalAttack();

};

Boss.cpp:

#include "boss.h"

Boss::Boss(string name, int hp, int damage, int danoUlti)

: Character(name, hp, damage), danoUlti(danoUlti) {

}

int Boss::finalAttack() {

return danoUlti;

}

Character.cpp:

#include "character.h"

Character::Character(string name, int hp, int damage) : name(name), hp(hp), damage(damage) {}

int Character::attack() {

return damage;

}

void Character::takeDamage(int dmg) {

hp -= dmg;

if (hp < 0) hp = 0;

}

string Character::getStatus() {

return "Status:\n Name: " + name + "\n HP: " + to\_string(hp) + "\n Damage: " + to\_string(damage);

}

bool Character::isAlive() {

return hp > 0;

}

GameAppOOv2.cpp:

#include <iostream>

#include "Character.h"

#include "Boss.h"

using namespace std;

int main() {

Character hero("Elpepe", 100, 10);

Character enemy("Sauron", 80, 8);

Boss boss("Pedro Sanchez", 150, 15, 25);

cout << hero.getStatus() << endl;

cout << enemy.getStatus() << endl;

while (hero.isAlive() && enemy.isAlive()) {

enemy.takeDamage(hero.attack());

if (enemy.isAlive()) {

hero.takeDamage(enemy.attack());

}

cout << hero.getStatus() << endl;

cout << enemy.getStatus() << endl;

}

if (hero.isAlive()) {

cout << "\n¡El jugador ha vencido! Ahora se enfrenta al jefe final.\n";

while (hero.isAlive() && boss.isAlive()) {

boss.takeDamage(hero.attack());

if (boss.isAlive()) {

hero.takeDamage(boss.finalAttack());

}

cout << hero.getStatus() << endl;

cout << boss.getStatus() << endl;

}

if (hero.isAlive()) {

cout << "\n¡Felicidades! Has vencido al jefe final.\n";

}

else {

cout << "\nEl jefe final te ha derrotado. Fin del juego.\n";

}

}

else {

cout << "\nHas sido derrotado. Fin del juego.\n";

}

return 0;

}

Por último, un ejemplo del código:

Texto

El contenido generado por IA puede ser incorrecto.