

INFORME DE CAMBIOS VIDEOJUEGO

Referencias

En cuanto a las referencias he añadido 3 referencias distintas en mi código.

La primera referencia es para mejorar y acortar código sobre el daño que se le realiza a cada enemigo anteriormente lo realizaba por separado y personalizado para cada enemigo de esta forma solo lo declaro una vez para todos adjunto la imagen del Código:

```
void enemyAttackDmg(int& damage, string& eName) {  
    damage = 1 + (rand() % 25);  
    heroHP = heroHP - damage;  
    Sleep(dwMilliseconds:500);  
    cout << "El enemigo " << eName << " te ha atacado con " << damage << " y te quedan: " << heroHP << " PX\n";  
}
```

La segunda referencia es para mejorar y acortar código sobre el ataque a cada enemigo antes tenía los ataques separados para cada enemigo y de esta forma resumo mi código bastante y solo lo tengo redactado una única vez para atacar a todos los enemigos y debido a que mi Código de ataque es bastante extenso ahora queda más limpio y sobre todo más entendible para todos adjunto la imagen del código:

```
bool ataqueEnemy(string& eName, string& eName2, int& damage, int& damage2, int& eHP, int& eHP2) {  
    if (eHP > 0) {  
        Sleep(dwMilliseconds:500);  
        cout << "\t" << "Le metes un golpe de " << heroDamage << " de DMG!\n";  
        eHP = eHP - heroDamage;  
    }  
    if (eHP <= 0) {  
        Sleep(dwMilliseconds:500);  
        cout << "Te has cargado al enemigo " << eName << "\n";  
        enemyIsAlive = false;  
        PlaySound(TEXT("muerte.wav"), NULL, SND_RESOURCE | SND_ASYNC);  
    }  
    else if (enemigo == 1) {  
        Sleep(dwMilliseconds:500);  
        cout << "El enemigo " << eName << " tiene: " << eHP << " PX\n";  
    }  
    else {  
        Sleep(dwMilliseconds:500);  
        cout << "El enemigo " << eName2 << " tiene: " << eHP2 << " PX\n";  
    }  
    if (enemyIsAlive) {  
        enemyAttackDmg(&damage, &eName);  
    }  
    if (enemyIsAlive2) {  
        enemyAttackDmg(&damage2, &eName2);  
        return false;  
    }  
}
```

Por ultimo tengo una tercera referencia dedicada a resumir mi Código sobre la elección correcta del enemigo en el caso de no seleccionar un enemigo valido me sigue preguntando el enemigo hasta seleccionar un enemigo valido, además en el caso de solo quedar un enemigo vivió solo te da la opción de seleccionar al enemigo vivo y de nuevo si seleccionas un enemigo no valido como el enemigo muerto no dejara seguir hasta seleccionar al único vivo, de esta forma ahora mi Código es mas simple y de nuevo mas entendible para el usuario que lo ve, adjunto una imagen del Código:

```
int OpcionCorrectaSimple(string& eName) {  
  
    int enemigoseleccionado = 0;  
    int correctChoice = 0;  
    while (correctChoice == 0) {  
        Sleep(dwMilliseconds:500);  
        cout << "Escoje al enemigo que deseas atacar " << eName << " con el numero [2]: \n";  
        cin >> enemigoseleccionado;  
        if (enemigoseleccionado == 2) {  
            correctChoice = 1;  
        }  
        else {  
            Sleep(dwMilliseconds:500);  
            cout << "Escoje una opcion valdia\n";  
        }  
    }  
    return enemigoseleccionado;  
}
```

Cambios en mi Código

Los cambios que he realizado en mi código sobre todo son en cuanto a la opción correcta seleccionada ya que antes si un enemigo moria seguía dejándome seleccionarlo y atacarlo a pesar de estar muerto perdiendo una ronda ahora gracias a preguntar si esta muerto o no un enemigo solo puedes seleccionarlo de forma correcta al que está vivo adjunto captura del código;

```
int opcioCorrecta() {  
  
    int enemigoseleccionado = 0;  
    int correctChoice = 0;  
    if (enemyIsAlive == true && enemyIsAlive2 == true) {  
        while (correctChoice == 0) {  
            Sleep(dwMilliseconds:500);  
            cout << "Escoje al enemigo que deseas atacar " << enemyName << " con el numero [1] " << enemyName2 << " con el numero [2]: \n";  
            cin >> enemigoseleccionado;  
            if (enemigoseleccionado == 1) {  
                correctChoice = 1;  
            }  
            else if (enemigoseleccionado == 2) {  
                correctChoice = 1;  
            }  
            else {  
                Sleep(dwMilliseconds:500);  
                cout << "Escoje una opcion valdia\n";  
            }  
        }  
    }  
    else if (enemyIsAlive == true && enemyIsAlive2 == false) {  
        OpcionCorrectaSimple(&eName: enemyName);  
    }  
    else if (enemyIsAlive == false && enemyIsAlive2 == true) {  
        OpcionCorrectaSimple(&eName: enemyName2);  
    }  
    return enemigoseleccionado;  
}
```

Otro cambio también realizado es el de poner tiempos de espera hasta que se ejecuten las líneas de código sobre todo en cuánto a los cout para tener una interacción mas tranquila con el código y poder leer con tiempo el código en la captura anterior se puede apreciar con el comando Sleep(500).