

## Contenido

Variables.....	2
main().....	3
enemiesAttack().....	4
checkHeroStatus() .....	4
Parámetros por referencia.....	5
atackChoose() .....	6

## Variables

En las variables se podrá apreciar dos simples cambios, el primero en los nombres de los atributos del enemigo1 debido a que usaba los parámetros anteriores para realizar los parámetros por referencia. El segundo cambio que se puede apreciar es la incorporación de una nueva variable llamada vida que la vamos a introducir en el apartado de enemiesAttack() que vamos a ver más tarde.

```
//Atributos del enemigo1
int enemylive = 1000;
int enemydamage;
string Enemyname = "Malo";
bool enemyIsAlive = true;
//sonidos
//Atributos del heroe
string heroname;
int herodamage;
int heroHP = 5000;
bool heroIsAlive = true;
int escoger;
//Atributos del enemigo2
int villanolive = 1000;
int villanodamage;
string Villanoname = "Malisimo";
bool villanoIsAlive = true;
//ataques especiales
int espada = 350;
int golpe = 250;
int magia = 500;
int usos = 2;
int ataque;
Marc Crespo
```

ANTES

```
//Atributos del enemigo1
bool enemy1IsAlive = true;
int enemy1HP = 1000;
int enemyDamage;
int attack;
string enemyname = "Malo";
//Atributos del heroe
string heroName;
int heroHP = 5000;
bool heroIsAlive = true;
int escoger = 0;
int heroDamage;
//Atributos del enemigo2
bool villanoIsAlive = true;
int villanoHP = 1000;
int villanoDamage;
int attack;
string enemyname2 = "Malisimo";
//ataques especiales
int espada = 350;
int golpe = 250;
int magia = 500;
int limitado = 2;
int ataque;
//variables pelea
int vida;
Marc Crespo
```

DESPUÉS

## main()

Después de refactorizar todo el código anterior la parte en la que se vera mas reflejado es en el main debido a que la cantidad de cosas que habían escritas se van a ver reducidas a palabras que les hemos puesto a nuestras funciones principales en su momento. Podemos observar que ha habido cambios en el nombre de variables, pero es debido a causa de que las podamos identificar más fácilmente.

```
int main()
{
    srand(time(NULL));
    gameStart();
    while ((enemyIsAlive || villanoIsAlive) && heroIsAlive) {
        chooseEnemy();
        chooseattack();
        enemydamage = rand() % 999 + 30;
        villanodamage = rand() % 999 + 30;
        heroIsAlive = enemysAttack();
    }
}
```

Marc Crespo

ANTES

```
int main() {
    srand(time(NULL));
    gameStart();
    while ((enemy1IsAlive || villanoIsAlive) && heroIsAlive) {
        //SELECCIONAMOS AL ENEMIGO QUE QUEREMOS ATACAR
        enemyChoose();
        //SELECCIONAMOS EL ATAQUE QUE QUEREMOS HACERLE AL ENEMIGO
        atackChoose();
        //NOS ATACAN LOS ENEMIGOS Y COMPROBAMOS SI AUN LES QUEDA VIDA RESTANTE
        enemiesAttack();
        //COMPROBAMOS SI AL HEROE AUN LE QUEDA VIDA RESTANTE
        heroIsAlive = checkHeroStatus();
    }
}
```

Marc Crespo

DESPUÉS

## enemiesAttack()

En este apartado después de ver el código pude observar que había cierta dificultad para poder leer bien el código. Entonces decidí que iba a dividir esta función en dos funciones posteriormente.

```
bool enemiesAttack() {
    enemydamage = rand() % 999 + 30;
    villanodamage = rand() % 999 + 30;
    if (enemyIsAlive && villanoIsAlive) {
        heroHP = heroHP - (enemydamage + villanodamage);
        if (heroHP <= 0) {
            heroHP = 0;
        }
        cout << "El enemigo y el villano te han hecho " << enemydamage + villanodamage << " puntos de ataque y te quedan " << heroHP << " puntos de vida.\n\n";
    }
    else if (enemyIsAlive == true && villanoIsAlive == false) {
        heroHP = heroHP - enemydamage;
        if (heroHP <= 0) {
            heroHP = 0;
        }
        cout << "El enemigo te ha hecho " << enemydamage << " puntos de ataque y te quedan " << heroHP << " puntos de vida.\n\n";
    }
    else if (enemyIsAlive == false && villanoIsAlive == true) {
        heroHP = heroHP - villanodamage;
        if (heroHP <= 0) {
            heroHP = 0;
        }
        cout << "El villano te ha hecho " << villanodamage << " puntos de ataque y te quedan " << heroHP << " puntos de vida.\n\n";
    }
    else if (enemyIsAlive == false && villanoIsAlive == false) {
        cout << "Has ganado el combate!!!\n";
    }
    if (heroHP <= 0) {
        return false;
        cout << "El heroe ha sido derrotado\n";
    }
}
```

ANTES

```
void enemiesAttack() {
    enemyDamage = rand() % 999 + 30;
    villanoDamage = rand() % 999 + 30;
    if (enemyIsAlive && villanoIsAlive) {
        vida = enemyDamage + villanoDamage;
        heroHP = heroHP - vida;
        if (heroHP <= 0) {
            heroHP = 0;
        }
        cout << "\nLos enemigos te han hecho " << vida << " te quedan " << heroHP << " puntos de vida\n\n";
    }
    else if (enemyIsAlive && villanoIsAlive == false) {
        heroHP = heroHP - enemyDamage;
        if (heroHP <= 0) {
            heroHP = 0;
        }
        cout << "\nEl " << enemyname << " te ha hecho " << enemyDamage << " te quedan " << heroHP << " puntos de vida\n\n";
    }
    else if (enemyIsAlive == false && villanoIsAlive) {
        heroHP = heroHP - villanoDamage;
        if (heroHP <= 0) {
            heroHP = 0;
        }
        cout << "\nEl " << enemyname2 << " te ha hecho " << villanoDamage << " te quedan " << heroHP << " puntos de vida\n\n";
    }
    else if (enemyIsAlive == false && villanoIsAlive == false) {
        cout << "Has Ganado el combate\n";
    }
}
```

DESPUÉS

## checkHeroStatus()

Este pequeño apartado forma parte de como decía antes una de las dos funciones que hemos creado nuevas partiendo de una, Creo que hacerlo en otra función nos permite observar con más claridad cómo funciona esta.

```
bool checkHeroStatus() {
    if (heroHP <= 0) {
        cout << "El heroe ha sido derrotado\n ";
        return false;
    }
    else {
        return true;
    }
}
```

Marc Crespo


## Parámetros por referencia

He creado dos funciones nuevas, una es la función de `attackChose()` que esta sirve para saber el ataque que va a realizar el héroe. La otra función que he creado es la función `statusCheck()` en esta compruebo si al enemigo le queda vida restante o si esta muerto.

```
void attackChose(int& enemyHP) {
    if (ataque == 1) {
        enemyHP = enemyHP - espada;
    }
    else if (ataque == 2) {
        enemyHP = enemyHP - golpe;
    }
    else if (ataque == 3) {
        if (limitado != 0) {
            enemyHP = enemyHP - magia;
            limitado -= 1;
        }
        else {
            cout << "no has podido atacar con el ataque especial de magia porque no te quedan usos restantes\n";
        }
    }
}

else
{
    "Has fallado, este ataque no existe";
}

bool statusCheck(bool& enemyIsAlive, int& enemyHP, string& enemynamex) {
    if (enemyHP <= 0) {
        enemyHP = 0;
        enemyIsAlive = false;
        cout << "Has derrotado al " << enemynamex << "\n";
    }
    else {
        return true;
    }
}
```



## atackChoose()

Gracias a las dos funciones anteriores he podido reducir la función atackChoose() reduciendo a más de la mitad del espacio.

```
void chooseAttack() {
    cout << "\nQue ataque quieres hacer?\n";
    "Espada [1]\n" "Golpe [2]\n" "Magia[3] Te quedan " << usos << " ataques restantes\n";
    cin >> ataque;
    if (enemyIsAlive) {
        if (escoger == 1) {
            if (ataque == 1) {
                enemyLive = enemyLive - espada;
            }
            else if (ataque == 2) {
                enemyLive = enemyLive - golpe;
            }
            else if (ataque == 3) {
                if (usos != 0) {
                    enemyLive = enemyLive - magia;
                    usos -= 1;
                }
                else {
                    cout << "No te quedan mas ataques de magia restantes\n";
                }
            }
            else {
                cout << "Acabas de perder tu oportunidad de atacar\n";
            }
        }
        if (enemyLive <= 0) {
            enemyLive = 0;
            enemyIsAlive = false;
            cout << "Has derrotado al enemigo\n";
        }
        cout << "\nAl enemigo malo le quedan " << enemyLive << " de vida y al villano malísimo le quedan " << villanoLive << " de vida.\n\n";
    }
    else {
        cout << "\nYa has matado al enemigo, no puedes seguir atacando\n";
    }
    if (villanoIsAlive) {
        if (escoger == 2) {
            if (ataque == 1) {
                villanoLive = villanoLive - espada;
            }
            else if (ataque == 2) {
                villanoLive = villanoLive - golpe;
            }
            else if (ataque == 3 && usos != 0) {
                if (ataque == 3) {
                    villanoLive = villanoLive - magia;
                    usos -= 1;
                }
                else {
                    cout << "No te quedan mas ataques de magia restantes\n";
                }
            }
            else {
                cout << "Acabas de perder tu oportunidad de atacar\n";
            }
        }
        if (villanoLive <= 0) {
            villanoLive = 0;
            villanoIsAlive = false;
            cout << "Has derrotado al villano\n";
        }
        cout << "\nAl enemigo malo le quedan " << enemyLive << " de vida y al villano malísimo le quedan " << villanoLive << " de vida.\n\n";
    }
    else {
        cout << "\nYa has matado al villano, no puedes seguir atacando\n";
    }
}
```

ANTES

```
void atackChoose() {
    //ESCOGEMOS EL ATAQUE QUE QUEREMOS REALIZAR
    cout << "Que ataque quieres realizar:\n" "Espada[1]\n" "Golpe[2]\n" "Magia[3] Quedan " << limitado << " ataques restantes\n";
    cin >> ataque;
    //ATACAMOS AL ENEMIGO 1 Y TAMBIÉN COMPROBAMOS SI ESTA VIVO O ESTA MUERTO
    if (enemyIsAlive) {
        if (escoger == 1) {
            attackChose(enemy1HP);
            enemyIsAlive = statusCheck(enemyIsAlive, enemy1HP, enemynome);
        }
    }
    else {
        cout << "\nEl " << enemynome << " esta muerto no puedes seguir atacandole, has perdido el turno\n";
    }
    //ATACAMOS AL ENEMIGO 2 Y TAMBIÉN COMPROBAMOS SI ESTA VIVO O ESTA MUERTO
    if (villanoIsAlive) {
        if (escoger == 2) {
            attackChose(villanoHP);
            villanoIsAlive = statusCheck(villanoIsAlive, villanoHP, enemynome2);
        }
    }
    else {
        cout << "\nEl " << enemynome2 << " esta muerto no puedes seguir atacandole, has perdido el turno\n";
    }
    cout << "\nAl " << enemynome << " le quedan " << enemy1HP << " puntos de vida\n";
    cout << "\nAl " << enemynome2 << " le quedan " << villanoHP << " puntos de vida\n";
}
```

DESPUÉS