	M5 - ENTORNOS DE DESARROLLO		
	UF2. Optimització de programari		
M05-M16-UF2-Pr4-Refactorizacion-de-codigo			
Apellidos: Jodar Ramos	Nombre: Martí	Fecha:	15/01/2023

Descripción y Objetivos


El objetivo principal de esta práctica es aprender a refactorizar nuestro código, aplicando las nuevas técnicas y habilidades adquiridas, como los parámetros por referencia.

Refactorización del Código del Videojuego

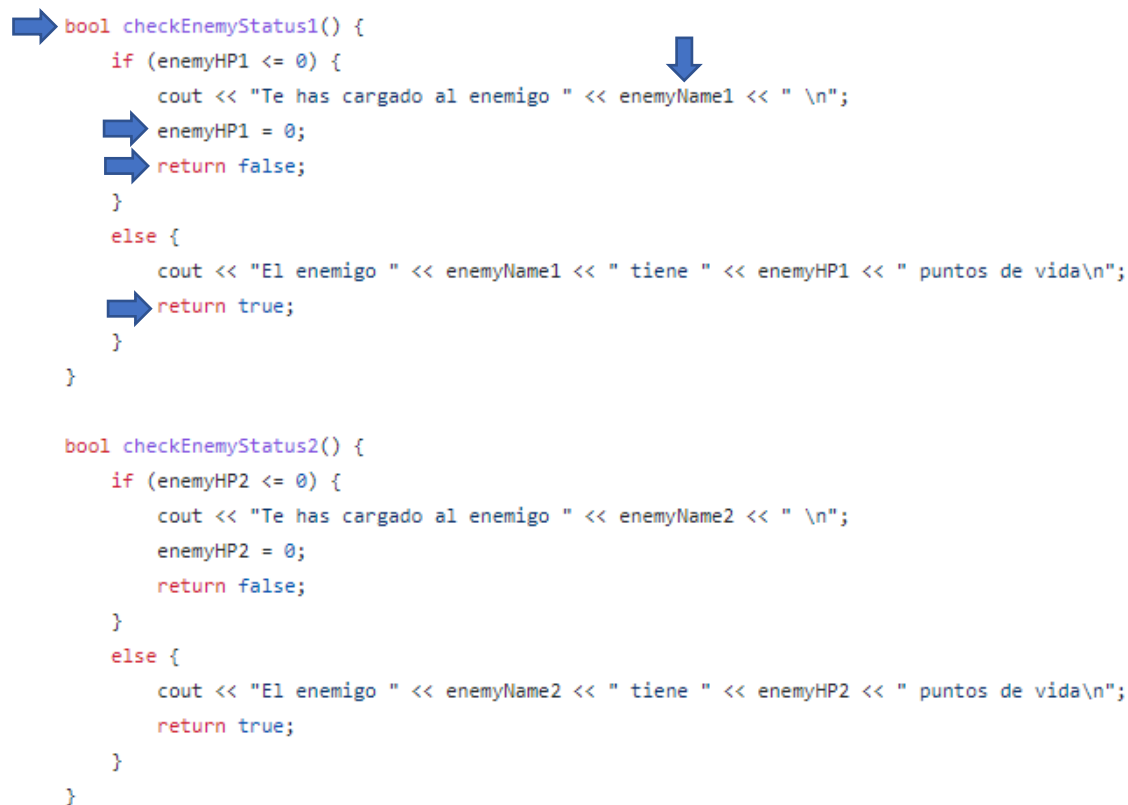
1. Usar las variables/parámetros por referencia para crear funciones globales donde creáis oportuno de vuestro código. **(5p)**
2. Modificar el código para mejorarlo, usando lo aprendido en M3 y M5/M16 desde el desarrollo de la PR3 **(3p)**
3. Subir a GIT el juego, teniendo en cuenta los parámetros del apartado de **Entrega de la Actividad PR4. (2p)**

Entrega de la Actividad Pr4:

- Al Moodle se subirá un fichero con únicamente una URL del proyecto de GIT
- La evaluación se completará con una **comprobación *in situ***.
- En el proyecto de GIT, dentro de una carpeta que se llame PR4, tendrá que haber:
 - El código del juego (Sin los archivos de DEBUG)
 - Un documento PDF explicando todos los cambios realizados a nivel de código/funciones
- En caso de que el proyecto GIT sea privado, ir a SETTINGS→COLLABORATORS y añadir al usuario @JAVIZAWA

	M5 - ENTORNOS DE DESARROLLO		
	UF2. Optimització de programari		
M05-M16-UF2-Pr4-Refactorizacion-de-codigo			
Apellidos: Jodar Ramos	Nombre: Martí	Fecha:	15/01/2023

CODIG ORIGINAL:



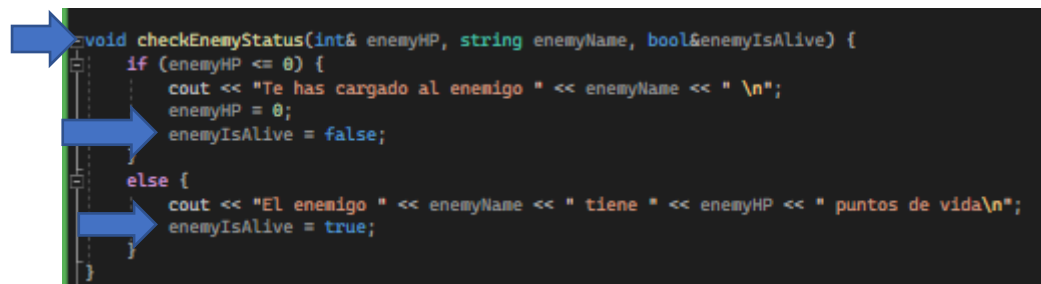
```
bool checkEnemyStatus1() {
    if (enemyHP1 <= 0) {
        cout << "Te has cargado al enemigo " << enemyName1 << " \n";
        enemyHP1 = 0;
        return false;
    }
    else {
        cout << "El enemigo " << enemyName1 << " tiene " << enemyHP1 << " puntos de vida\n";
        return true;
    }
}

bool checkEnemyStatus2() {
    if (enemyHP2 <= 0) {
        cout << "Te has cargado al enemigo " << enemyName2 << " \n";
        enemyHP2 = 0;
        return false;
    }
    else {
        cout << "El enemigo " << enemyName2 << " tiene " << enemyHP2 << " puntos de vida\n";
        return true;
    }
}
```

MODIFICACIÓ:

He eliminat els dos enemyStatus1 i 2 per enemyStatus. He declarat com a variable de referència de tipus int a enemyHP i de tipus bool a enemyIsAlive. També he declarat com variable d'entrada de tipus string a enemy name.

Degut a que he declarat la funció com un void i a més enemyIsAlive es un bool, igualet la variable a true o a false segons l'estat de vida de l'enemic.



```
void checkEnemyStatus(int& enemyHP, string enemyName, bool& enemyIsAlive) {
    if (enemyHP <= 0) {
        cout << "Te has cargado al enemigo " << enemyName << " \n";
        enemyHP = 0;
        enemyIsAlive = false;
    }
    else {
        cout << "El enemigo " << enemyName << " tiene " << enemyHP << " puntos de vida\n";
        enemyIsAlive = true;
    }
}
```

CODIG ORIGINAL:

```
if (enemyChoosed ==1){
    //ATACAMOS AL ENEMIGO 1
    enemyHP1 = heroAttackEnemy1(heroDamage);
    enemyIsAlive1 = checkEnemyStatus1();
}else{
    //ATACAMOS AL ENEMIGO 2
    enemyHP2 = heroAttackEnemy2(heroDamage);
    enemyIsAlive2 = checkEnemyStatus2();
}
```

MODIFICACIÓ:

Degut a que la funció checkEnemyStatus ha passat a ser un void, només escrivim el nom sense igualar-ho a res.

```
if (enemyChoosed == 1) {
    //ATACAMOS AL ENEMIGO 1
    enemyHP1 = heroAttackEnemy(heroDamage, enemyName1, enemyHP1);
    checkEnemyStatus(enemyHP1, enemyName1, enemyIsAlive1);
}
else {
    //ATACAMOS AL ENEMIGO 2
    enemyHP2 = heroAttackEnemy(heroDamage, enemyName2, enemyHP2);
    checkEnemyStatus(enemyHP2, enemyName2, enemyIsAlive2);
}
```

CODIG ORIGINAL:

```
int chooseEnemy(){
64 int enemyChoose = 0;
65 while (enemyChoose != 1 && enemyChoose != 2){
66     cout << "A que enemigo quieres atacar? \n";
67     if(enemyIsAlive1){
68         cout << "1.- " << enemyName1 << "\n";
69     }
70     if(enemyIsAlive2) {
71         cout << "2.- " << enemyName2 << "\n";
72     }
73     cin >> enemyChoose;
74     if (enemyChoose <=0 || enemyChoose >2){
75         cout << "La opcion elegida no es correcta. Por favor, elige una opcion valida. \n";
76     }
77     if(!enemyIsAlive1 && enemyChoose==1){
78         cout << "El enemigo " << enemyName1 << " ya esta muerto. Por favor, elige un enemigo que todavia este vivo." "
79         \n";
80         enemyChoose =0;
81     }
82     if(!enemyIsAlive2 && enemyChoose==2){
83         cout << "El enemigo " << enemyName2 << " ya esta muerto. Por favor, elige un enemigo que todavia este vivo." "
84         \n";
85         enemyChoose =0;
86     }
87     return enemyChoose;
}
```

MODIFICACIÓ:

He canviat la funció choseEnemy de tipus int a void (eliminant també l'int de la variable enemyChose = 0 de la segona linea). També he declarat la variable per referència de tipus int a enemyChose.

Ja que la funció és un void no ens fa falta el return.

```
49 void chooseEnemy( int& enemyChose) {
50     enemyChose = 0;
51     while (enemyChose != 1 && enemyChose != 2) {
52         cout << "A que enemigo quieres atacar? \n";
53         if (enemyIsAlive1) {
54             cout << "1.- " << enemyName1 << "\n";
55         }
56         if (enemyIsAlive2) {
57             cout << "2.- " << enemyName2 << "\n";
58         }
59         cin >> enemyChose;
60         if (enemyChose <= 0 || enemyChose > 2) {
61             cout << "La opcion elegida no es correcta. Por favor, elige una opcion valida. \n";
62         }
63         if (!enemyIsAlive1 && enemyChose == 1) {
64             cout << "El enemigo " << enemyName1 << " ya esta muerto. Por favor, elige un enemigo que todavia este vivo." << "\n";
65             enemyChose = 0;
66         }
67         if (!enemyIsAlive2 && enemyChose == 2) {
68             cout << "El enemigo " << enemyName2 << " ya esta muerto. Por favor, elige un enemigo que todavia este vivo." << "\n";
69             enemyChose = 0;
70         }
71     }
72 }
```

CODIG ORIGINAL:

```
while(heroIsAlive && (enemyIsAlive1 || enemyIsAlive2)){
    //ELEGIMOS ENEMIGO AL QUE ATACAR
    enemyChosed = chooseEnemy();
```

MODIFICACIÓ:

Degut a que la funció choseEnemy ha passat a ser un void, només escrivim el nom sense igualar-ho a res.

```
while (heroIsAlive && (enemyIsAlive1 || enemyIsAlive2)) {
    //ELEGIMOS ENEMIGO AL QUE ATACAR
    chooseEnemy(enemyChosed);
```

CODIG ORIGINAL:

```
int heroAttackEnemy1(int damage) {  
    cout << "Atacas al enemigo " << enemyName1 << " y le haces un danyo de " << damage << " \n";  
    return enemyHP1 - damage;  
}  
  
int heroAttackEnemy2(int damage) {  
    cout << "Atacas al enemigo " << enemyName2 << " y le haces un danyo de " << damage << " \n";  
    return enemyHP2 - damage;  
}
```

MODIFICACIÓ:

He eliminat una funció i també he eliminat totes les variables 1 i 2 per només una sense numero (exemple: enemyName1→enemyName). A la nova variable la declarem com a heroAttackEnemy i he declarat com a variable d'entrada el damage i enemyName, de tipus int i string respectivament. També he declarat com a variable per referència de tipus int a enemyHP.

```
int heroAttackEnemy(int damage, string enemyName, int& enemyHP) {  
    cout << "Atacas al enemigo " << enemyName << " y le haces un danyo de " << damage << " \n";  
    return enemyHP - damage;  
}
```

CODIG ORIGINAL:

```
if (enemyChoosed ==1){  
    //ATACAMOS AL ENEMIGO 1  
    enemyHP1 = heroAttackEnemy1(heroDamage);  
    enemyIsAlive1 = checkEnemyStatus1();  
}  
else{  
    //ATACAMOS AL ENEMIGO 2  
    enemyHP2 = heroAttackEnemy2(heroDamage);  
    enemyIsAlive2 = checkEnemyStatus2();  
}
```

MODIFICACIÓ:

Posem el nom de la nova funció (heroAttackEnemy) i amb les variables declarades anteriorment indicant amb un numero l'enemic a la qual afecta.

```

if (enemyChosed == 1) {
    //ATACAMOS AL ENEMIGO 1
    enemyHP1 = heroAttackEnemy(heroDamage, enemyName1, enemyHP1);
    checkEnemyStatus(enemyHP1, enemyName1, enemyIsAlive1);
}
else {
    //ATACAMOS AL ENEMIGO 2
    enemyHP2 = heroAttackEnemy(heroDamage, enemyName2, enemyHP2);
    checkEnemyStatus(enemyHP2, enemyName2, enemyIsAlive2);
}

```

CODIG ORIGINAL:

```

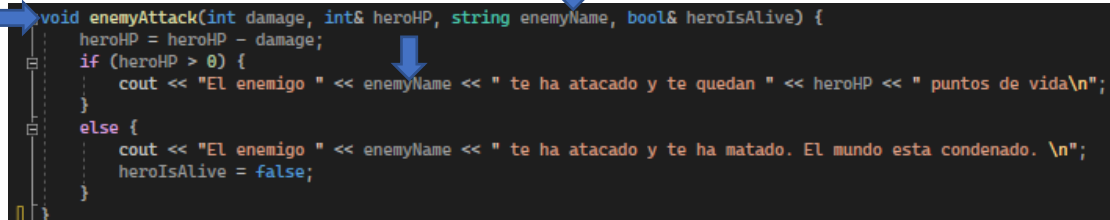
void enemyAttack1(int damage) {
    heroHP = heroHP - damage;
    if(heroHP > 0){
        cout << "El enemigo " << enemyName1 << " te ha atacado y te quedan " << heroHP << " puntos de vida\n";
    }else{
        cout << "El enemigo " << enemyName1 << " te ha atacado y te ha matado. El mundo esta condenado. \n";
        heroIsAlive = false;
    }
}

void enemyAttack2(int damage) {
    heroHP = heroHP - damage;
    if(heroHP > 0){
        cout << "El enemigo " << enemyName2 << " te ha atacado y te quedan " << heroHP << " puntos de vida\n";
    }else{
        cout << "El enemigo " << enemyName2 << " te ha atacado y te ha matado. El mundo esta condenado. \n";
        heroIsAlive = false;
    }
}

```

MODIFICACIÓ:

He eliminat una funció i també he eliminat totes les variables 1 i 2 per només una sense numero (exemple: enemyName1 → enemyName). A la nova variable la declarem com a enemyAttack i he declarat com a variable d'entrada el damage i enemyName, de tipus int i string respectivament. També he declarat com a variable per referència a heroHP i heroIsAlive, de tipus int i bool.



```
void enemyAttack(int damage, int& heroHP, string enemyName, bool& heroIsAlive) {
    heroHP = heroHP - damage;
    if (heroHP > 0) {
        cout << "El enemigo " << enemyName << " te ha atacado y te quedan " << heroHP << " puntos de vida\n";
    }
    else {
        cout << "El enemigo " << enemyName << " te ha atacado y te ha matado. El mundo esta condenado. \n";
        heroIsAlive = false;
    }
}
```

The image shows a C++ code snippet for the `enemyAttack` function. Blue arrows point to the function signature and the `enemyName` parameter, indicating the changes made.

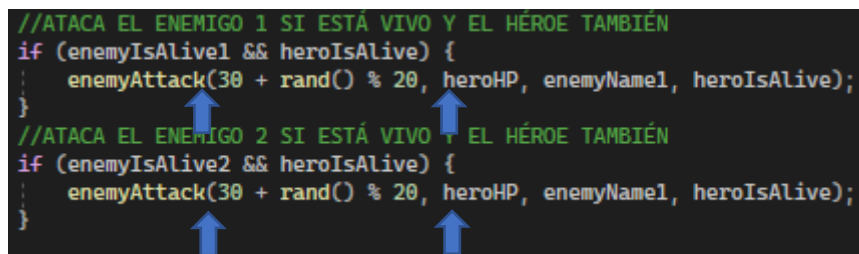
CODIG ORIGINAL:

```
//ATACA EL ENEMIGO 1 SI ESTÁ VIVO Y EL HÉROE TAMBIÉN
if (enemyIsAlive1 && heroIsAlive) {
    enemyAttack1(30 + rand()%20);
}
//ATACA EL ENEMIGO 2 SI ESTÁ VIVO Y EL HÉROE TAMBIÉN
if (enemyIsAlive2 && heroIsAlive) {
    enemyAttack2(30 + rand()%20);
}
```

The image shows the original code with two separate attack functions, `enemyAttack1` and `enemyAttack2`. Blue arrows point to the function calls, indicating the changes made.

MODIFICACIÓ:

Canviem el nom de la funció al nou `enemyAttack` i declarant les variables amb un numero segons l'enemic.



```
//ATACA EL ENEMIGO 1 SI ESTÁ VIVO Y EL HÉROE TAMBIÉN
if (enemyIsAlive1 && heroIsAlive) {
    enemyAttack(30 + rand() % 20, heroHP, enemyName1, heroIsAlive);
}
//ATACA EL ENEMIGO 2 SI ESTÁ VIVO Y EL HÉROE TAMBIÉN
if (enemyIsAlive2 && heroIsAlive) {
    enemyAttack(30 + rand() % 20, heroHP, enemyName1, heroIsAlive);
}
```

The image shows the modified code where both `enemyAttack1` and `enemyAttack2` are replaced with a single `enemyAttack` function call. Blue arrows point to the function calls, indicating the changes made.

MAIN ORIGINAL:

```
int main()
{
    gameStart();
    while(heroIsAlive && (enemyIsAlive1 || enemyIsAlive2)){
        //ELEGIMOS ENEMIGO AL QUE ATACAR
        enemyChooosed = chooseEnemy();
        //ELEGIMOS ATAQUE Y CAPTURAMOS EL VALOR DEL DANYO QUE HAREMOS
        heroDamage = chooseAttack();

        if (enemyChooosed ==1){
            //ATACAMOS AL ENEMIGO 1
            enemyHP1 = heroAttackEnemy1(heroDamage);
            enemyIsAlive1 = checkEnemyStatus1();
        }else{
            //ATACAMOS AL ENEMIGO 2
            enemyHP2 = heroAttackEnemy2(heroDamage);
            enemyIsAlive2 = checkEnemyStatus2();
        }
        //ATACA EL ENEMIGO 1 SI ESTÁ VIVO Y EL HÉROE TAMBIÉN
        if (enemyIsAlive1 && heroIsAlive) {
            enemyAttack1(30 + rand()%20);
        }
        //ATACA EL ENEMIGO 2 SI ESTÁ VIVO Y EL HÉROE TAMBIÉN
        if (enemyIsAlive2 && heroIsAlive) {
            enemyAttack2(30 + rand()%20);
        }
    }
}
```

MAIN MODIFICAT:

```
int main()
{
    gameStart();
    while (heroIsAlive && (enemyIsAlive1 || enemyIsAlive2)) {
        //ELEGIMOS ENEMIGO AL QUE ATACAR
        chooseEnemy(enemyChooosed);
        //ELEGIMOS ATAQUE Y CAPTURAMOS EL VALOR DEL DANYO QUE HAREMOS
        heroDamage = chooseAttack();

        if (enemyChooosed == 1) {
            //ATACAMOS AL ENEMIGO 1
            enemyHP1 = heroAttackEnemy(heroDamage, enemyName1, enemyHP1);
            checkEnemyStatus(enemyHP1, enemyName1, enemyIsAlive1);
        }
        else {
            //ATACAMOS AL ENEMIGO 2
            enemyHP2 = heroAttackEnemy(heroDamage, enemyName2, enemyHP2);
            checkEnemyStatus(enemyHP2, enemyName2, enemyIsAlive2);
        }

        //ATACA EL ENEMIGO 1 SI ESTÁ VIVO Y EL HÉROE TAMBIÉN
        if (enemyIsAlive1 && heroIsAlive) {
            enemyAttack(30 + rand() % 20, heroHP, enemyName1, heroIsAlive);
        }
        //ATACA EL ENEMIGO 2 SI ESTÁ VIVO Y EL HÉROE TAMBIÉN
        if (enemyIsAlive2 && heroIsAlive) {
            enemyAttack(30 + rand() % 20, heroHP, enemyName1, heroIsAlive);
        }
    }
}
```