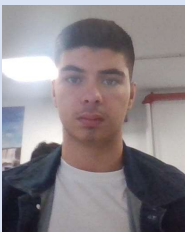
	M3 PROGRAMACION BASICA	
	UF1-Programacion estructurada	
Estructuras Condicionales III		
Apellidos: Herraiz Foz	Nombre: Angel	Fecha: 02/01/2023

```

39 > void gamestarter() { ...
61
62 > int wizardaahero() { ...
66
67 > int goblinaahero() { ...
71
72 > int heroaawizard() { ...
76
77 > int heroaagoblin() { ...
81
82 > void gamestarter2() { ...
117
118 > void attackselecting() { ...
178
179 > int enemysselecting() { ...
202
203 > boolean goblinfightback() { ...
222
223 > boolean heroattackgoblin() { ...
262
263 > boolean heroattackwizard() { ...
326
327 > boolean wizardturn() { ...
346
347 > boolean herofightback() { ...
365
366 > boolean goblinturn() { ...
382
383 > void endgame() { ...
398
399 > int main() { ...

```

ANTES DE LA OPTIMIZACIÓN

Angel Herraiz

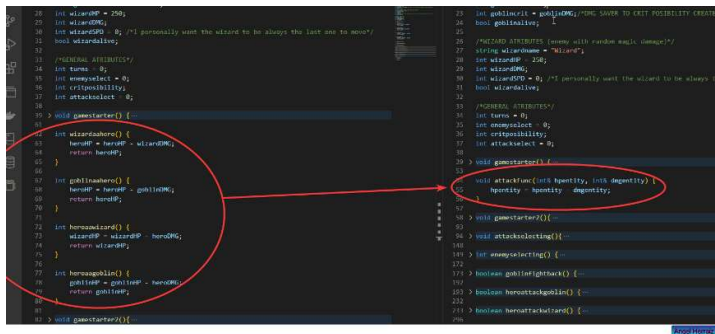
```

> void gamestarter() { ...
> void attackfunc(int& hentity, int& dmgentity) { ...
> void gamestarter2() { ...
> void attackselecting() { ...
> int enemysselecting() { ...
> boolean goblinfightback() { ...
> boolean heroattackgoblin() { ...
> boolean heroattackwizard() { ...
> boolean wizardturn() { ...
> boolean herofightback() { ...
> boolean goblinturn() { ...
> void attacksys() { ...
> void endgame() { ...
> int main() { ...

```

DESPUÉS DE LA OPTIMIZACIÓN

X



```

28 int wizardHP = 250;
29 int wizardDMG;
30 int wizardSPD = 0; /*I personally want the wizard to be always the last one to move*/
31 bool wizardAlive;
32
33 /*GENERAL ATTRIBUTES*/
34 int turns = 0;
35 int enemysselect = 0;
36 int enemyskill;
37 int attackselect = 0;
38
39 > void gamestarter() { ...
40
41 > int wizardaahero() { ...
42
43 > int goblinaahero() { ...
44
45 > int heroaawizard() { ...
46
47 > int heroaagoblin() { ...
48
49 > void gamestarter2() { ...
50
51 > void attackselecting() { ...
52
53 > int enemysselecting() { ...
54
55 > boolean goblinfightback() { ...
56
57 > boolean heroattackgoblin() { ...
58
59 > boolean heroattackwizard() { ...
60
61 > boolean wizardturn() { ...
62
63 > boolean herofightback() { ...
64
65 > boolean goblinturn() { ...
66
67 > void endgame() { ...
68
69 > int main() { ...

```

El cambio más grande visible consiste en que previamente, existían cuatro funciones de ataque, una para cuando el mago ataca al héroe, otra para cuando el goblin ataca al héroe, y otras dos cuando el héroe ataca a alguno de los enemigos. Después de la revisión, reduje el código a

una sola función que recibe dos parámetros, siendo uno la vida de la entidad que va a recibir un ataque, y el otro siendo el daño de ese ataque.El

otro cambio más importante, fue en la función main. Esta consistía en la ejecución de diversas funciones que constaban de los turnos de los enemigos y los ataques de las distintas entidades. La optimización en esta parte del código consistió en englobar todos estos turnos dentro de una función a la que he llamado attacksys().

```

int main() {
    gamestarter();

    /*RPG TURN SYSTEM CREATING*/
    while ((goblinHP >= 1) || (wizardHP >= 1) && (heroHP >= 1)) {
        gamestarter2();

        attackselecting();

        /*ENEMY SELECTOR*/
        enemysselecting();

        attacksys();

        endgame();
    }
}

```

Función Main después de la optimización

```

int main() {
    gamestarter();

    /*RPG TURN SYSTEM CREATING*/
    while ((goblinHP >= 1) || (wizardHP >= 1) && (heroHP >= 1)) {
        gamestarter2();

        attackselecting();

        /*ENEMY SELECTOR*/
        enemysselecting();

        /*FIGHT SYSTEM ON GOBLIN*/

        /*IF HERO IS FASTER THAN THE GOBLIN*/
        if (heroSPD >= goblinSPD) {
            heroSPD = 0;
            goblinSPD = 100;

            /*IF YOU SELECT TO ATTACK GOBLIN*/
            heroattackgoblin();

            /*IF YOU SELECT TO ATTACK WIZARD*/
            heroattackwizard();

            /*GOBLIN'S TURN*/
            goblinturn();

            /*WIZARD'S TURN*/
            wizardturn();

            /*IF HERO IS SLOWER THAN THE GOBLIN*/
            else if (heroSPD <= goblinSPD) {
                heroSPD = 100;
                goblinSPD = 0;

                /*THE GOBLIN ATTACKS*/
                goblinturn();

                /*IF U SELECTED GOBLIN*/
                heroattackgoblin();

                /*IF U SELECTED WIZARD*/
                heroattackwizard();

                /*WIZARD'S TURN*/
                wizardturn();
            }

            /*RESET DAMAGE FOR GOBLIN IF THIS TURN HATE ATTACKED CRITICALLY
            if (turns > 1) {
                goblinDMG = goblincrit;
            }
            endgame();
        }
    }
}

```

Función Main previa a la optimización