



Πανεπιστήμιο Πελοποννήσου

Τμήμα Ψηφιακών Συστημάτων

ΣΧΟΛΗ ΟΙΚΟΝΟΜΙΑΣ & ΤΕΧΝΟΛΟΓΙΑΣ

ΤΜΗΜΑ ΨΗΦΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

**Εργασία εξαμήνου στην C**

**ΚΑΡΑΒΙΑ ΜΑΡΙΑ ΣΠΥΡΙΔΟΥΛΑ**

**ΑΜ:2026202100049**

**Εξάμηνο: 5**

ΙΑΝΟΥΑΡΙΟΣ , 2024

```

#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <sys/types.h>
#include <windows.h>
#include <unistd.h>
#include <math.h>
int hp = 0;
//φτιάχνω το μενού και τις επιλογές του
int menu(){
    int choise;
    printf("\n###----- Select from the below options -----
--###\n");
    printf("\nContinue battles with the same stats : Press (1)\n");
    printf("\nSmall Rest : Continue battles with the same heal 10/100 hp and lose 10/100 XP :
Press (2)\n");
    printf("\nBig Rest : Continue battles with the same heal 20/100 hp and lose 20/100 XP :
Press (3)\n");
    scanf("%d",&choise);
    return choise;
}
//υπολογισμός hp του ήρωα όταν χάνει
void update_hero_stats(int points_lost) {
    hp -= points_lost;
    if (hp < 0) {
        hp = 0; //το hp δεν πηγαίνει κάτω του μηδενός
    }
}
//εκτυπώνω τα στατιστικά του ήρωα
void print_stats_player(int level,int hp,int damage,int XP){
    printf("\nHero stats : level %d , Hit Points : %d , Damage : %d , XP : %d ",
level, hp, damage, XP);
}
//παράγω το τέρας
int * generate_monster(int diff ){
    srand(time(0)); //δημιουργία ψευδοτυχαίων τιμών
    static int monster_stats[2]; //στατιστικός πίνακας για αποθήκευση monster_hp και
monster_damage
    int monster_hp = (rand() % (diff *2- diff + 1)) + diff; //υπολογισμός του monster_hp
ανάλογα με την δυσκολία του τέρατος
    int monster_damage = (rand() % (diff/2 - diff/4 + 1)) + diff/4; //υπολογισμός του
monster_damage ανάλογα με την δυσκολία του τέρατος
    printf("\nMonster stats : Hit Points : %d , Damage : %d ,", monster_hp, monster_damage);
    //αποθήκευση στοιχείων του τέρατος που δημιουργήθηκαν στον στατικό πίνακα
    monster_stats[0]=monster_hp;
    monster_stats[1]=monster_damage;
    return monster_stats;
}
//πόντοι εμπειρίας μετά την νίκη του τέρατος
int gainXP(int monster_hp,int monster_damage){
    return monster_hp+monster_damage;
}

```

```

}
//διατήρηση του παλιού hp στο new_hp
int holdHp(int old_hp){
    int new_hp ;
    return new_hp = old_hp;
}
//μάχη
int fight(int level,int hp,int damage,int monster_hp,int monster_damage,int num_battles){
    printf("\n###----- ( Battle: %d ) -----###\n",num_battles); //εκτυπώνεται ο αριθμός της μάχης
    int round =1;
    while((hp>0 && monster_hp>0)){//ο ήρωας και το τέρας μάχονται εφόσον ισχύει η
    συνθήκη μέσα στην while
        printf("\n----- round: %d ----- ",round);
        monster_hp -=damage ;
        update_hero_stats(monster_damage);
        printf("\nMonster hp : %d",monster_hp); //το νέο monster_hp σε κάθε round

        if (monster_hp <= 0){
            break;
        }
        hp -= monster_damage;
        printf("\nHero hp : %d",hp); //το νέο hero hp σε κάθε round

        if(hp<=0){
            break;
        }
        round++; //τα rounds συνεχίζονται
    }
    //πότε σταματάει η μάχη (πότε χάνει ο ήρωας και πότε το τέρας)
    if (hp<=0){
        printf("\nHero die");
        return 0; //ο ήρωας χάνει
    }
    else{
        printf("\nMonster die");
        return 1; //ο ήρωας νικάει
    }
}

int main(){
    //τα στατιστικά με τα οποία ξεκινά το παιχνίδι
    int level = 1;
    hp = 100;
    int new_hp = 100;
    int damage =10;
    int diff = 20;
    int XP=0;
    int *monster_stats;
    print_stats_player(level,hp,damage,XP);
    monster_stats = generate_monster(diff);

```

```

int choise;
int num_battles = 1;
    //βρόχος για να συνεχίζονται οι μάχες
while(fight(level, hp, damage,* (monster_stats),*(monster_stats+1),num_battles) !=0){
    XP+=gainXP(* (monster_stats) ,*(monster_stats+1)); //αυξάνονται τα χρ με βάση τα
monster_stats
        //αν ο ήρωας αναίβει level
if(XP>= 50*row(2,level)){//πόντοι εμπειρίας για να αναίβει level ο ήρωας
    sleep(4); //αναμονή 4 sec
    level++; //ανεβαίνει level
    damage +=5; //υπολογισμός damage αφού level
    hp =new_hp; //νέο hp ήρωα
    hp+=20; //λαμβάνει +20 hp για κάθε level που ανεβαίνει
    new_hp = hp;
    printf("\n Congratulation you level up \n Boost your Hp and Damage \n");
    print_stats_player(level, hp, damage, XP); //νέα hero stats
}
//αν ο ήρωας δεν αναίβει level (επιλογές)
else{
    choise = menu();
    //συνεχίζει με την επόμενη μάχη
    if(choise==1){
        }
        //ξεκουράζεται λίγο
    else if(choise ==2){
        hp=hp+10/100 * hp;
        XP=XP-10/100*XP;
        }
        //ξεκουράζεται πολύ
    else if (choise == 3){
        hp=hp+20/100 * hp;
        XP=XP-20/100*XP;
        }
        else{
            printf("Wrong input try again from option (1) ,(2) ,(3)");
            choise = menu();
        }
    }

    diff+=4; //νίκησε ο ήρωας άρα η δυσκολία αυξάνεται κατά 4
    printf("\n %d \n", diff);
    monster_stats = generate_monster(diff); //υπολογισμός δυσκολίας του τέρατος

    print_stats_player(level, hp, damage, XP); //τυπωνόνται τα στατιστικά του ήρωα
    num_battles ++; //οι μάχες συνεχίζονται
}
return 0;
}

```

```

Hero stats : level 1 , Hit Points : 100 , Damage : 10 , XP : 0
Monster stats : Hit Points : 40 , Damage : 6 ,
###----- ( Battle: 1 ) -----###

----- round: 1 -----
Monster hp : 30
Hero hp : 94

----- round: 2 -----
Monster hp : 20
Hero hp : 88

----- round: 3 -----
Monster hp : 10
Hero hp : 82

----- round: 4 -----
Monster hp : 0
Monster die
###----- Select from the below options -----###

Continue battles with the same stats : Press (1)

Small Rest : Continue battles with the same heal 10/100 hp and lose 10/100 XP : Press (2)

Big Rest : Continue battles with the same heal 20/100 hp and lose 20/100 XP : Press (3)
2

24

Monster stats : Hit Points : 42 , Damage : 9 ,
Hero stats : level 1 , Hit Points : 76 , Damage : 10 , XP : 46
###----- ( Battle: 2 ) -----###

----- round: 1 -----
Monster hp : 32
Hero hp : 67

----- round: 2 -----
Monster hp : 22
Hero hp : 58

----- round: 3 -----
Monster hp : 12
Hero hp : 49

----- round: 4 -----
Monster hp : 2
Hero hp : 40

----- round: 5 -----
Monster hp : -8
Monster die
###----- Select from the below options -----###

Continue battles with the same stats : Press (1)

Small Rest : Continue battles with the same heal 10/100 hp and lose 10/100 XP : Press (2)

Big Rest : Continue battles with the same heal 20/100 hp and lose 20/100 XP : Press (3)
1

28

Monster stats : Hit Points : 44 , Damage : 13 ,
Hero stats : level 1 , Hit Points : 31 , Damage : 10 , XP : 97
###----- ( Battle: 3 ) -----###

----- round: 1 -----
Monster hp : 34
Hero hp : 18

----- round: 2 -----
Monster hp : 24
Hero hp : 5

----- round: 3 -----
Monster hp : 14
Hero hp : -8
Hero die
-----

```

- Το πιο δύσκολο σημείο στην συγγραφή του κώδικα ήταν η παραγωγή του τέρατος με βάση τη δυσκολία, δηλαδή το παρακάτω τμήμα του κώδικα:

```
//παράγω το τέρας
int * generate_monster(int diff ){
    srand(time(0)); //δημιουργία ψευδοτυχαίων τιμών
    static int monster_stats[2]; //στατιστικός πίνακας για αποθήκευση monster_hp και
    monster_damage
    int monster_hp = (rand() % (diff *2- diff + 1)) + diff; //υπολογισμός του monster_hp
    ανάλογα με την δυσκολία του τέρατος
    int monster_damage = (rand() % (diff/2 - diff/4 + 1)) + diff/4; //υπολογισμός του
    monster_damage ανάλογα με την δυσκολία του τέρατος
    printf("\nMonster stats : Hit Points : %d , Damage : %d , " ,
    monster_hp,monster_damage);
    //αποθήκευση στοιχείων του τέρατος που δημιουργήθηκαν στον στατικό πίνακα
    monster_stats[0]=monster_hp;
    monster_stats[1]=monster_damage;
    return monster_stats;
}
```