Final project

Project description

In the beginning, the player needs to set the size of the map, then set village, monsters, player, gun and trap location. The player need to combat monster in this game. Each monster has a different value of strength and the player can only beat the monster which has less value of strength than the player. If the player defeats the monster, the player's strength can be incremented by 1. However, if the player can't beat the monster, the player's blood will be decremented by 1 to 4 according to the number of monsters. The player will have 10 blood and 2 strength at the beginning. The player can choose to go to the village before selecting which monster to combat. In the village, the player can do three actions. The first action is to buy a magical sword to increase the number of strengths by 1. The player can only buy one sword during the same visit to the village. If the player buys n swords after n iterations without fighting any monster, the player's strength will be increased by n. The second action is to go to the bar and have a drink, which will increase the value of the blood by 1. The player can go to the bar multiple times during the same visit to the village until the value of blood reaches 10. The third action is to exit the village. After exiting the village the player will be asked to choose which monster to fight. If the player move to the location of gun, player can increase player_attack. If the player move to trap, the player will minus his/her player_heakth.

Playing method

Using direction key to control the player's location. Follow the instruction on the screen. You can leave the game if you choose 5 to exit, or you will stop the game until your player_health less than 0.

Function description

- start_game() : Initial map, village, monster, player, gun, trap
- create_map() : Create a map for the game by using 2D array and asking user to input the size of row and column of the array

- print_map() : The program use **v** to label the location of the village, **m** to label the location of monsters, **p** to label the location of player, **g** to label the location of gun, **t** to label the location of trap
- setup_village() : set village location
- setup_monster() : set monster location
- setup_player() : set player location
- setup_gun() : set gun location
- setup_trap() : set trap location
- encounter_village() : player can do three things here, 1. buy a magical sword 2. go to the bar 3. leave the village
- encounter_monster() : player have to choose a monster(1~4)
- battle_result() : set random seed for checking if the monster dodge successfully and judge whether player or monster win
- go_up()/go_down()/go_right()/go_left() : check every moves legal, judge if the player encounter v, m, g, t
- check_boundary() : Show a warning message "**the location is outside the map**" when the location of the village or monsters is outside the range of the array.
- check_availability() : Check whether the location is already occupied by the village or other monsters. If the location is already taken by the village or other monsters, the program needs to show a warning message "**the location is occupied**"
- travel_map() : move the player up/down/right/left

Variable description

- char** map : declare a double pointer of a map
- int map_row, map_col : map_row*map_col size of map
- int player_row, player_col /village_row, village_col/monster_row, monster_col/gun_row, gun_col/trap_row, trap_col : the location of player/village/monster/gun/trap
- int choose : create a new map or start travel the map
- int monster : decide which monster to fight
- int player_health : initial as 10
- int player_attack : initial as 2
- int seed : dodge probability
- int go : decide whether go to the village or not
- int sword : judge whether the play already buy a sword

Version history

Global variables version

```c
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
#include<time.h>

int map_row, map_col, village_row, village_col, monster_row, monster_col, player_row, player_col, gun_row, gun_col, trap_row, trap_col;
int monster; // choose a monster to fight
int player_health =10;
int player_attack = 2;
char map[1000][1000];

int main(){
    printf("Welcome, Adventurer!\n");
    init_game();
    return 0;
}
void create_map(){
    printf("Input the number of row and column for the map: \n");
    scanf("%d%d", &map_row, &map_col);
    //set all the elements to be '.'
    for(int i=0; i<map_row; i++){
        for(int j=0; j<map_col ; j++)
            map[i][j] = '.';
    }
    print_map();
}
void print_map(){
    printf("=== MAP ===\n");
    for(int j=0; j<map_row; j++){
        for(int k=0; k<map_col; k++){
            printf("%c ", map[j][k]);
        }
    printf("\n");
    }
    printf("=== MAP ===\n");
}
```

```c
void setup_village(){
    while(1){
        printf("Input the row and column for the village location: \n");
        scanf("%d%d", &village_row, &village_col);
        if(check_boundary(village_row, village_col) != 0 && check_availability(village_row, village_col) != 0){
            map[village_row][village_col] = 'v';
            break;
        }
    }
    print_map(map_row, map_col);
}
void encounter_villiage(){
    int go; // decide whether go to the village or not
    int choose; // what to do in the village
    int sword=1;
    while(1){
        printf("Do you want to go to our village first (1:Yes; 0: No)?\n");//Ask if the player wants to go to the village.
        scanf("%d", &go);
        if(go == 1){
            printf("What do you want to do in our village (1:buy a magical sword; 2: go to the bar; 3:leave the village)?\n");
            scanf("%d", &choose);
            if(choose == 1){
                if(sword == 1){
                    player_attack++;
                    printf("Nice sword! Now you have %d strength!\n", player_attack);
                    sword = 0;
                }
                else if(sword == 0)
                    printf("You already brought a sword.\n");
            }
            else if(choose == 2){
                if(player_health<10){
                    player_health++;
                    printf("Nice beer! Now you have %d player_health!\n", player_health);
                }
                else if(player_health>=10)
                    printf("Your player_health is full. You don't need a beer.\n");
            }
```

```
            else if(choose == 3){
                printf("You are welcome to come back anytime!\n");
                break;
            }
        }
        else
            break;
    }
}
void setup_monster(){
    int monster = (map_row/10) + (map_row%10);
    printf("Input the row and column for the monster location: \n");
    for(int i=0; i<monster; i++){
        printf("Input the row and column for monster %d: \n", i);
        scanf("%d%d", &monster_row, &monster_col);
        if(check_boundary(monster_row, monster_col) != 0 && check_availability(monster_row, monster_col) != 0)
            map[monster_row][monster_col] = 'm';
        else
            i--;
    }
    print_map();
}
void encounter_monster(){
    printf("Please choose which monster (1 to 4) you want to fight: \n");
    scanf("%d", &monster);
    if(monster<1 || monster>4)
        printf("No such monster, please enter 1 to 4 to choose the monster.\n");
    else{
        printf("You are fighting monster %d!\n", monster);
        battle_result();
    }
    if(player_health<=0){
    printf("You Dead. Bye!\n");
    exit(0);
    }
}
void battle_result(){
    srand(time(0));
```

```
    int seed;
    while(1){
        seed = rand()%101;
        if(seed>50){
            printf("dodge probability is : %d\n", seed);
            printf("monster dodge successfully\n");
        }
        else{
            printf("dodge probability is : %d\n", seed);
            if(player_attack>monster){
                player_attack++;
                printf("You Win! \n");
                break;
            }
            else{
                player_health -= monster;
                printf("You Lose!\n");
                break;
            }
        }
        printf("Please choose which monster (1 to 4) you want to fight: \n");
        scanf("%d", &monster);
    }
}
void setup_player(){
    while(1){
        printf("Input the row and column for player: \n");
        scanf("%d%d", &player_row, &player_col);
        if(check_boundary(player_row, player_col) != 0 && check_availability(player_row, player_col) != 0){
            map[player_row][player_col] = 'p';
            break;
        }
    }
    print_map();
}
void go_up(){
    player_row--;
    if(check_boundary(player_row, player_col) != 0){
```

```
        if(map[player_row][player_col] == '.'){
            map[player_row][player_col] = 'p';
            map[player_row+1][player_col] = '.';
        }
        else if(map[player_row][player_col] == 'v'){
            encounter_villiage();
            player_row++;
        }
        else if(map[player_row][player_col] == 'm'){
            encounter_monster();
            player_row++;
        }
        else if(map[player_row][player_col] == 'g'){
            player_attack++;
            printf("You have %d player_health and %d player_attack.\n", player_health, player_attack);
            player_row++;
        }
        else if(map[player_row][player_col] == 't'){
            player_health--;
            printf("You have %d player_health and %d player_attack.\n", player_health, player_attack);
            player_row++;
        }
        print_map();
    }
    else{
        printf("You can't go there!\n");
        player_row++;
    }
}
void go_down(){
    player_row++;
    if(check_boundary(player_row, player_col) != 0){
        if(map[player_row][player_col] == '.'){
            map[player_row][player_col] = 'p';
            map[player_row-1][player_col] = '.';
        }
        else if(map[player_row][player_col] == 'v'){
            encounter_villiage();
```

```
            player_row--;
        }
        else if(map[player_row][player_col] == 'm'){
            encounter_monster();
            player_row--;
        }
        else if(map[player_row][player_col] == 'g'){
            player_attack++;
            printf("You have %d player_health and %d player_attack.\n", player_health, player_attack);
            player_row--;
        }
        else if(map[player_row][player_col] == 't'){
            player_health--;
            printf("You have %d player_health and %d player_attack.\n", player_health, player_attack);
            player_row--;
        }
        print_map();
    }
    else{
        printf("You can't go there!\n");
        player_row--;
    }
}
void go_right(){
    player_col++;
    if(check_boundary(player_row, player_col) != 0){
        if(map[player_row][player_col] == '.'){
            map[player_row][player_col] = 'p';
            map[player_row][player_col-1] = '.';
        }
        else if(map[player_row][player_col] == 'v'){
            encounter_villiage();
            player_col--;
        }
        else if(map[player_row][player_col] == 'm'){
            encounter_monster();
            player_col--;
        }
```

```
        else if(map[player_row][player_col] == 'g'){
            player_attack++;
            printf("You have %d player_health and %d player_attack.\n", player_health, player_attack);
            player_col--;
        }
        else if(map[player_row][player_col] == 't'){
            player_health--;
            printf("You have %d player_health and %d player_attack.\n", player_health, player_attack);
            player_col--;
        }
        print_map();
    }
    else{
        printf("You can't go there!\n");
        player_col--;
    }
}
void go_left(){
    player_col--;
    if(check_boundary(player_row, player_col) != 0){
        if(map[player_row][player_col] == '.'){
            map[player_row][player_col] = 'p';
            map[player_row][player_col+1] = '.';
        }
        else if(map[player_row][player_col] == 'v'){
            encounter_villiage();
            player_col++;
        }
        else if(map[player_row][player_col] == 'm'){
            encounter_monster();
            player_col++;
        }
        else if(map[player_row][player_col] == 'g'){
            player_attack++;
            printf("You have %d player_health and %d player_attack.\n", player_health, player_attack);
            player_col++;
        }
```

```
        else if(map[player_row][player_col] == 't'){
            player_health--;
            printf("You have %d player_health and %d player_attack.\n", player_health, player_attack);
            player_col++;
        }
        print_map();
    }
    else{
        printf("You can't go there!\n");
        player_col++;
    }
}
void setup_gun(){
    while(1){
        printf("Input the row and column for gun: \n");
        scanf("%d%d", &gun_row, &gun_col);
        if(check_boundary(gun_row, gun_col) != 0 && check_availability(gun_row, gun_col) != 0){
            map[gun_row][gun_col] = 'g';
            break;
        }
    }
    print_map();
}
void setup_trap(){
    while(1){
        printf("Input the row and column for trap: \n");
        scanf("%d%d", &trap_row, &trap_col);
        if(check_boundary(trap_row, trap_col) != 0 && check_availability(trap_row, trap_col) != 0){
            map[trap_row][trap_col] = 't';
            break;
        }
    }
}
int check_boundary(int row, int col){
    if(row<0 || row>map_row-1 || col<0 || col>map_col-1){
        printf("the location is outside the map\n");
        return 0;
    }
```

```
    else if(row<map_row && col<map_col)
        return 1;
}
int check_availability(int row, int col){
    if(map[row][col] == '.')
        return 1;
    else{
        printf("the location is occupied\n");
        return 0;
    }
}
void travel_map(){
    print_map();
    while(1){
        int choose;
        printf("[1] go up [2] go down [3] go right [4] go left [5] exit: ");
        scanf("%d", &choose);
        if(choose == 1)
            go_up();
        else if(choose == 2)
            go_down();
        else if(choose == 3)
            go_right();
        else if(choose == 4)
            go_left();
        else if(choose == 5)
            exit(0);
    }
}
void init_game(){
    while(1){
        int choose;
        printf("[1] Create a new map [2] Start travel the map: ");
        scanf("%d", &choose);
        if(choose == 1){
            create_map();
            setup_village();
            setup_monster();
```

```
        }
        else if(choose == 2){
            setup_player();
            setup_gun();
            setup_trap();
            travel_map();
        }
    }
}
```