Project Description

Players use the WASD arrow keys to move, the basic information includes HP, ACK, money and hidden attribute lucky value. If you move on the map, you can fight monsters to get gold coins and increase ACK. If you go to the store to buy teddy bears, you can get lucky, and you can get lucky with monsters in the future. Increases hit rate when sparring.
(P.S : The end of the game is not fully developed.)

Function Description

char** create_map();
Create a map for the adventure game by using the 2-dimensional array.

void print_map();
Print the 2-dimensional array for the map.
V = Village
I = Player at village
M = Monster
T = Trap
P = Player
G = Gun

void print_player_info();
Print out the player's personal information such as HP, ACK, money.

void setup_village();
Users can input the location of the village by themselves, and limit the number to only.

void setup_monster();
Users can input the location of the monster by themselves, and the number is (*row / 10) + (*col % 10) - 1.

void setup_player();
User enters the player's spawnpoint by himself.

void setup_trap();
Users can input the location of the trap by themselves, and the number is

(*row / 10) + (*col % 10) - 1.

void setup_gun();

User enters the gun on the map.The number is counted by row * col / 20.

int 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.

int check_availability();

Check whether the location is already occupied by the V or M or T. If the location is already taken, show a warning message “the location is occupied”.

void go_up();

Player go up.

void go_down();

Player go down.

void go_left();

Player go left.

void go_right();

Player go right.

int check_move();

If the player go outside the map, show a warning message “you can’t go there”.

int encounter_trap();

Check if the player has taken a trap, if so, show the message "You encounter a trap and lose 1 blood QAQ!", and lose 1 blood.

int encounter_village();

Check if the player has reached the village, if so, enter the village_action( ).

int encounter_gun();

Check if the player walks to 'g', if so get ACK+3

int encounter_monster();

Check if the player has encountered a monster, if so, enter the battle stage and choose the monster level to fight.

void village_action();

void monster_fight();

After selecting the monster level to fight, enter the battle function, compare the player's ACK to monster’s level, and distinguish whether it ACK bigger than lever, if so, enter the ***evasion()*** .

int evasion();

Use rand() to generate a number from 1 to 100. If it is greater than 50, the monster will evade. Otherwise, if the monster evades successfully, it will counterattack the blood drop of the player's monster level.

int isDead();

Determine whether the player is dead, check whether the blood volume is less than or equal to 0

Variable Description

■ Purpose

The purpose of the game is to fight monsters in the map, and you can go to the village to replenish blood and buy weapons.

■ Variable

// MAX_NUMBER : map size within 1000

```
#define False 0
#define True    1
#define MAX_NUMBER 1000
```

//Extra Part、Global var

int isLucky = False;

Reduce the evasion rate of monsters, allowing players to reduce the chance of deduction.

```
//Map's data & ptr
int maprow = MAX_NUMBER;
int mapcol = MAX_NUMBER;
int *ptr_maprow = &maprow;
int *ptr_mapcol = &mapcol;

//Player's position
int Prow = 0;
int Pcol = 0;
int *ptr_Prow = &Prow;
int *ptr_Pcol = &Pcol;

//Player's ability
int HP = 10;
int *ptr_HP = &HP;
int ACK = 2;
int *ptr_ACK = &ACK;

//Player's choice how to do at the map
int choice;
```

Version History

<0.1 – Initial Release>

<0.2 – Modify global var to pointer>

<0.3 – Solve dependency hell of pointer QAQ>

<0.4 – Basic point and Extra part>

<0.5 – Make code look not that ugly>