COURSE:* *C programming language

CLASS:* *34092102

BRUNEL* *ID:* *2161047

CQUPT ID*:*2021215069

NAME:* *Xukang Liu

1. AIM

Use C language to design a mine sweeping game to realize the purpose of setting the minefield and completing the mine sweeping process.

2. TASK STEPS

2.1 Set header file

```
#include<stdio.h>
#include<stdlib.h>//malloc function need
```

"stdio.h" include many basic functions, and we need "#include<stdlib.h>" to set dynamic memory allocation.

2.2 Mine Field Setting

```
int width=0, height=0;
    while (width <= 0 || height <= 0 || height > 60) {
        printf("Input MINE block width:");
        scanf_s("%d", &width);
        printf("Input MINE block height:");
        scanf_s("%d", &height);
        if (width <= 0 || height <= 0) {
            printf("Please enter the positive numbers.\n");
            printf("Let's do it again.\n\n");
        }
        else if (height > 60) {
            printf("The number of columns you entered is too
large\n");
            printf("Let's do it again.\n\n");
        }
    }
    printf("Building MINE block.....\n");
    printf("This is MINE block.....\n");
```

Before the Mine sweeper begins, we should set the field. So in the first place, we let clients to enter the width and height of the field. But in some places, clients may input error. So we need use "**if**" statement to judge whether the input is correct. If the input is really wrong, we use while loop statement, so you can input it again.

2.3 Assign a value to the array and display it

```
for (int i = 0; i < width; i++)
    {
        for (int j = 0; j < height; j++)
        {
            printf("%c%c ", 0xa8, 0x80);
        }
        printf("\n");
    }
```

Use twice 'for' loop ,so that you can get the mine field.

The statement "printf("%c%c",0xa8,0x80)" can output white blocks.

2.4 Mine setting

```
int x = 0, y = 0, mine = 0;//Define the x, y coordinates and number
of mines when entering mines mine
    char** Mine;//Define the mine array, which represents the
location where the mine exists
    Mine = (char**)malloc(width * sizeof(char*));
    for (int i = 0; i < width; i++) {
        Mine[i] = (char*)malloc(height * sizeof(char));
    }
```

Use dynamic memory allocation. This is the first array. This array is used to store minefields.

2.5 Initialize the array Mine，assign all value to 0

```
for (int i = 0; i < width; i++) {
    for (int j = 0; j < height; j++) {
        Mine[i][j] = 0;
    }
}
```

2.6 Input the number of Mines

```
do {
        printf("Input the number of MINEs you want to set:");
        scanf_s("%d", &mine);
        if (mine <= 0) {
            printf("Please enter the positive number.\n");
        }
        else if (mine > width * height) {
            printf("Out of the rage.\n");
        }
    } while (mine<=0 || mine>width*height);
    for (int i = 0; i < mine; )
    {
            printf("Input the %d-th MINE position:", i + 1);
            scanf_s("%d %d", &x, &y);
            if (Mine[x][y] == 1) {
                printf("The location of the mine is the same.\n");
            }
            else if (x >= 0 && y >= 0 && x <= width && y <= height)
            {
                Mine[x][y] = 1;
                i++;
            }
            else
            {
                printf("Please enter the correct number.\n");
            }
    }
```

Because mine has no initial value, use the "do-while" loop. In this segment, clients can input the value of mines.

In the "do-while "loop, we still set up an "if" statement to solve the input out of bounds problem.

After entering the number of mines, clients can enter the coordinate position of mines. If the position coordinates entered by the player are the same as the previous one, we still have a solution.

2.7 Mine_Sweeper begin

```
//Mine_Sweeper begin
printf("Mine Sweeper begins\n");
int a = 0;
int input_x = 0, input_y = 0;
int** Mine_position;//Definite the array Mineposition, which means
to guess the loation of the Mine
Mine_position = (int**)malloc(width * sizeof(int*));
for (int i = 0; i < width; i++) {
    Mine_position[i] = (int*)malloc(height * sizeof(int));
}
for (int i = 0; i < width; i++) {
    for (int j = 0; j < height; j++) {
        Mine_position[i][j] = 0;
    }
}

while (a<mine) {
    printf("Input the possible position of MINE:");
    scanf_s("%d %d",&input_x,&input_y);
    if (input_x >= 0 && input_x <= width && input_y >= 0 && input_y
<= height)
    {
        if (Mine_position[input_x][input_y] == 1) {
            printf("Please enter different position.\n");
        }else if (Mine[input_x][input_y] == 1)
          {
            Mine_position[input_x][input_y] = 1;
            printf("Congratulations,you get one MINE:\n");
            for (int i = 0; i < width; i++) {
                for (int j = 0; j < height; j++) {
                    if (Mine_position[i][j] == 1) {
                        printf("**");
                    }
                    else
                        printf("%c%c ", 0xa8, 0x80);
                }
```

```
                printf("\n");
            }
            a++;
          }
        else
            printf("Not success,try again...\n");
    }else
        printf("Please input the correct number.\n");
}
printf("\n");
printf("--------------------\n");
printf("-----GAME OVER-----\n");
printf("--------------------\n");
return 0;
```

In this segment, we set the second array, still use dynamic memory allocation. The array " Mine_position[]" means to contrast with the first array "Mine".If "Mine[input_x][input_y] == 1" ,let Mine_position[input_x][input_y] == 1. In the final output display process, we use many "for" loop to print the results.

3. SOURCE CODE

```
#include<stdio.h>
#include<stdlib.h>//malloc function need

int main() {
    //Mine Field Seeting
    int width=0, height=0;
    while (width <= 0 || height <= 0 || height > 60) {
```

```
        printf("Input MINE block width:");
        scanf_s("%d", &width);
        printf("Input MINE block height:");
        scanf_s("%d", &height);
        if (width <= 0 || height <= 0) {
            printf("Please enter the positive numbers.\n");
            printf("Let's do it again.\n\n");
        }
        else if (height > 60) {
            printf("The number of columns you entered is too
large\n");
            printf("Let's do it again.\n\n");
        }
    }
    printf("Building MINE block.....\n");
    printf("This is MINE block.....\n");
    //Assign a value to the array and display it
    for (int i = 0; i < width; i++)
    {
        for (int j = 0; j < height; j++)
        {
            printf("%c%c ", 0xa8, 0x80);
        }
        printf("\n");
    }

    //Mine setting
    int x = 0, y = 0, mine = 0;//Define the x, y coordinates and
number of mines when entering mines mine
    char** Mine;//Define the mine array, which represents the
location where the mine exists
    Mine = (char**)malloc(width * sizeof(char*));
    for (int i = 0; i < width; i++) {
        Mine[i] = (char*)malloc(height * sizeof(char));
    }
    //Initialize the array Mine, assign all value to 0
    for (int i = 0; i < width; i++) {
        for (int j = 0; j < height; j++) {
            Mine[i][j] = 0;
        }
    }
    //Input the number of Mines
    do {
```

```
        printf("Input the number of MINEs you want to set:");
        scanf_s("%d", &mine);
        if (mine <= 0) {
            printf("Please enter the positive number.\n");
        }
        else if (mine > width * height) {
            printf("Out of the rage.\n");
        }
    } while (mine<=0 || mine>width*height);

    for (int i = 0; i < mine; )
    {
            printf("Input the %d-th MINE position:", i + 1);
            scanf_s("%d %d", &x, &y);
            if (Mine[x][y] == 1) {
                printf("The location of the mine is the same.\n");
            }
            else if (x >= 0 && y >= 0 && x <= width && y <= height)
            {
                Mine[x][y] = 1;
                i++;
            }
            else
            {
                printf("Please enter the correct number.\n");
            }
    }

    //Mine_Sweeper begin
    printf("Mine Sweeper begins\n");
    int a = 0;
    int input_x = 0, input_y = 0;
    int** Mine_position;//Definite the array Mineposition, which
means to guess the loation of the Mine
    Mine_position = (int**)malloc(width * sizeof(int*));
    for (int i = 0; i < width; i++) {
        Mine_position[i] = (int*)malloc(height * sizeof(int));
    }
    for (int i = 0; i < width; i++) {
        for (int j = 0; j < height; j++) {
            Mine_position[i][j] = 0;
        }
    }
```

```
    while (a<mine) {
        printf("Input the possible position of MINE:");
        scanf_s("%d %d",&input_x,&input_y);
        if (input_x >= 0 && input_x <= width && input_y >= 0 &&
input_y <= height)
        {
            if (Mine_position[input_x][input_y] == 1) {
                printf("Please enter different position.\n");
            }else if (Mine[input_x][input_y] == 1)
              {
                Mine_position[input_x][input_y] = 1;
                printf("Congratulations,you get one MINE:\n");
                for (int i = 0; i < width; i++) {
                    for (int j = 0; j < height; j++) {
                        if (Mine_position[i][j] == 1) {
                            printf("**");
                        }
                        else
                            printf("%c%c ", 0xa8, 0x80);
                    }
                    printf("\n");
                }
                a++;
              }
            else
                printf("Not success,try again...\n");
        }else
            printf("Please input the correct number.\n");
    }
    printf("\n");
    printf("--------------------\n");
    printf("-----GAME OVER-----\n");
    printf("--------------------\n");
    return 0;
}
```