# C Project Report:*PUBG*

## Abstract

We want to create a game with combining the chess and tank war, in this game, we will use strategy to win the game. During creating the game, we were confronted with several problems. The construction of the graph and the movement of the players is one of the basic problems. The appearance of landmine and first aid kit is also very important. What’s more, the function of NPC is very difficulty as the graph is complex.

## Introduction/Problem Statement

In our childhood, video games maybe one of the best friends of us. Tank war is a pretty exciting one and it just like an old friend of us, it accompanied most of our childhood with the hot blood music and take an important position in us video gamers memory. It’s a game first developed by Namco and released in Japan in May 1985.

In recent years, the PUBG has made our blood boil. Combined with the Tank Wars game we used to play when we were young, we wanted to recreate the PUBG by using the newly learned C language.

To a game, we think the most significant factors are the interactivity and gamer experience. The goal of this game is to survive in bombing and use enemy supplies to destroy the enemy.

First of all, we need to create an initial graphical interface, which is very important for a game. Then we need to design maps and make maps as interesting and reasonable as possible. Then we need to make rules: each person can perform three operations per round, moving or shooting. Players' bullets cannot penetrate walls, and airdrop supplies and bombardment are randomly generated in the map.

After finishing the preparations, we have to decide the conditions of victory. There is a numeric table on the right side of the game to show the player's residual health. Whenever you are attacked by an opponent or bombed, you will reduce your life value. When moving to the airdrop first aid kit, it will increase the life value. When the value of life is zero, the opponent will win. Here we learn from classic games such as Tank Wars and bombers. This is also a more conventional way of judging.

Finally, we do the art design. Our road is green grass, and the walls are grey brick walls. The figures refer to the design of counterterrorism elite, and the first aid package is red and white to pay tribute to the Red Cross. The bombing area is a red flame to warn players to stay away from me.

Compared with other games, we decide to combine two different types of games, and we have added some new elements into the game such as the first aid kit and the landmine.

## Analysis

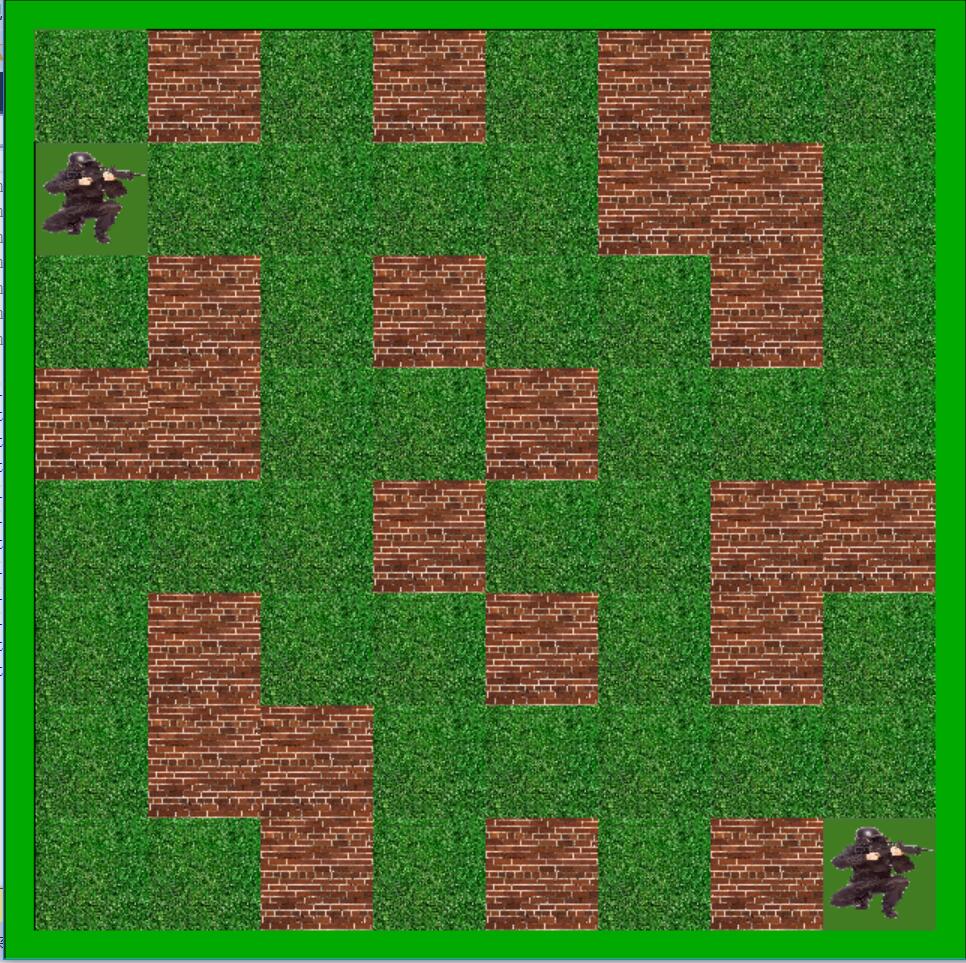
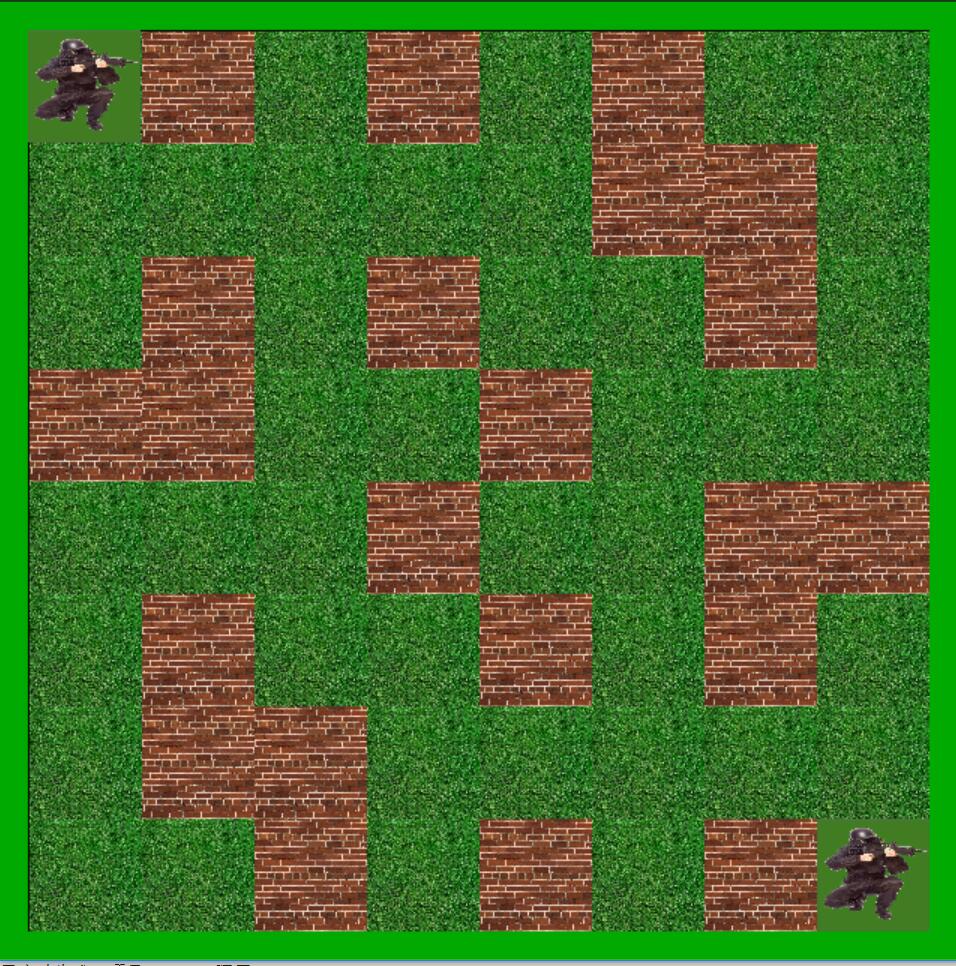
As we can see, the aim of our game is to combine chess and tank war to produce a game. In this game, we should achieve the round system (players will move by turn), and the players should be able to move and attack their opponent. What’s more, we have land mines、first aid kit and NPC to help the game become much more interesting.

The player should and only should move when we press the correct buttons, and when the players touch the wall or something can’t be touched, they can’t move either. And when the players attack, the computer should judge whether the bullet has touched the wall or the other players. About the land mine and the first aid kit, the appearance of these two things is very important. The first aid kit should appear randomly and the land mines should appear obeying a rule. What’s more the NPC should move according to the movement of players.

Now that we have analyzed the whole game roughly, we shall analyze the problem one by one .

Problem 1: The player moves on the screen by pressing certain buttons.

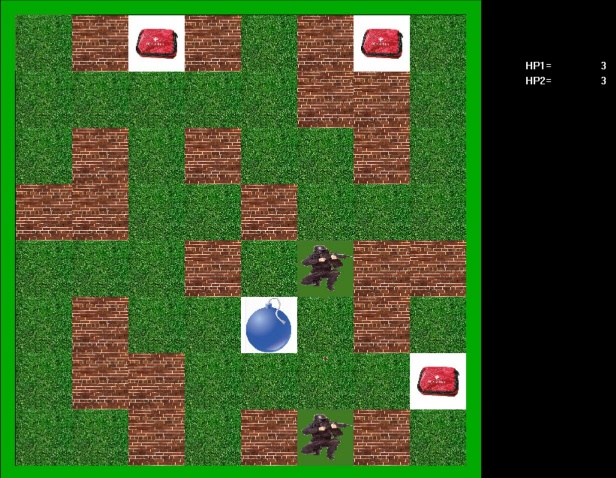
Constraint: What this ask us to do is to move the players, you have to press certain buttons and then the player will move according to the button you press. If you press illegal buttons, players won’t move and if you press the wrong button, the player won’t move as you wish.

Therefore, firstly we should make the player move by erasing the original player and draw it at the target place, by doing this , we can make the player move. And to make the player move as we wish ,we assign different modes of movement to different buttons and therefore, by pressing different buttons, we can move as we wished.

Problem 2: When the bullet touches your enemy, the HP of your enemy minus 1.

Constraint:What the computer should do is to judge whether the bullet has touched the enemy, we should make the computer know which is the enemy and which is the wall, what’s more, which is the empty place. And when the bullet touches the wall, the bullet just disappears and when the bullet touches the enemy, the HP of the enemy will minus 1.

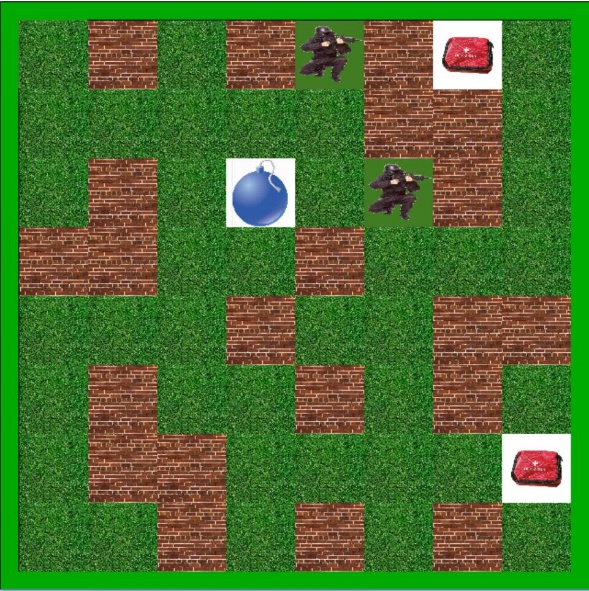
To solve this problem, two-dimension array to assign different values to several important points on the screen which help the computer to judge what the bullet has touched. The value of the wall and the value of the player is different. Therefore, by checking the value of the point which the bullet arrive at, we can judge the thing the bullet has touched. And we also assign another value to the player. When the bullet touches the player, this value will minus one, and we use this value to record the HP of the player.

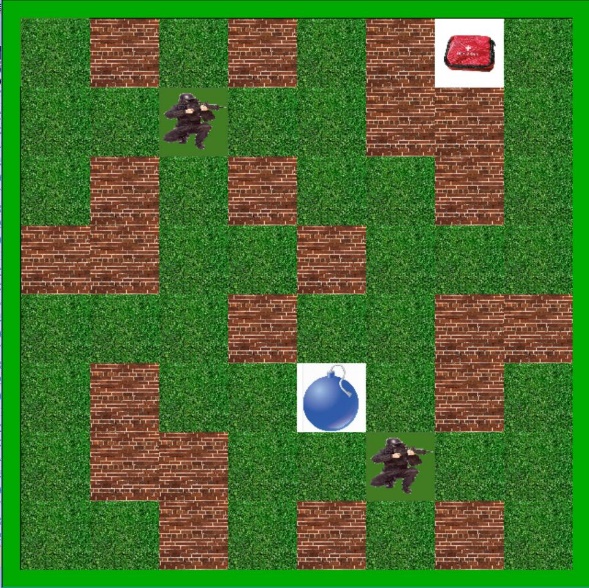


Problem 3: The appearance of the landmine and first aid kit.

Constraint: As we plan, the first aid kit should appear randomly on the empty place, and one first aid kit will appear every two rounds(during each round, each player can move three times), and the landmine will change the location also every two rounds. And the first aid kit and the landmine won’t appear and explode when there is you press some button illegally.

Firstly, about the appearance of the first aid kit, we use a loop to get random numbers and each time we get two random numbers, assuming these two numbers are x and y, we check whether on the coordinate (x,y) is wall or the player, if coordinate (x,y) is empty place, the loop will stop and we put the first aid kit here. About the landmine, we use a integer to record how many steps we have moved, and we use modular algorithm to get how many rounds we have played, and then we can check whether to put the landmine or not.

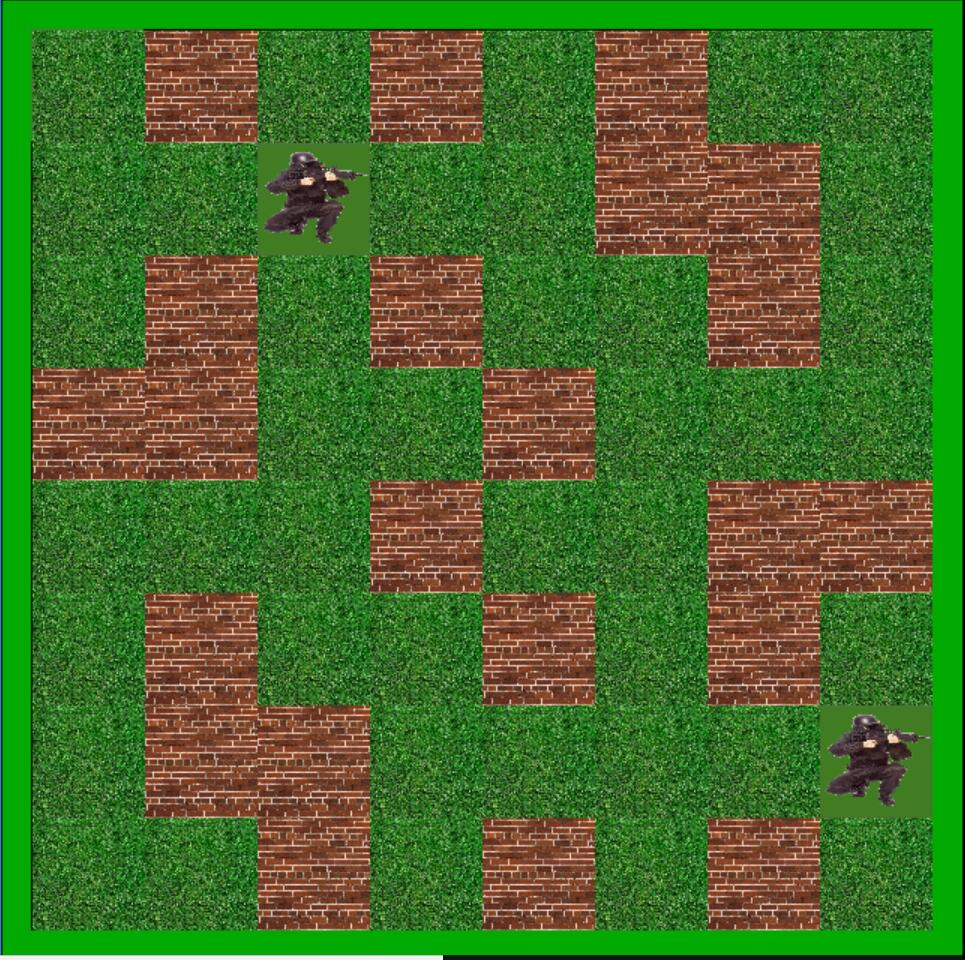
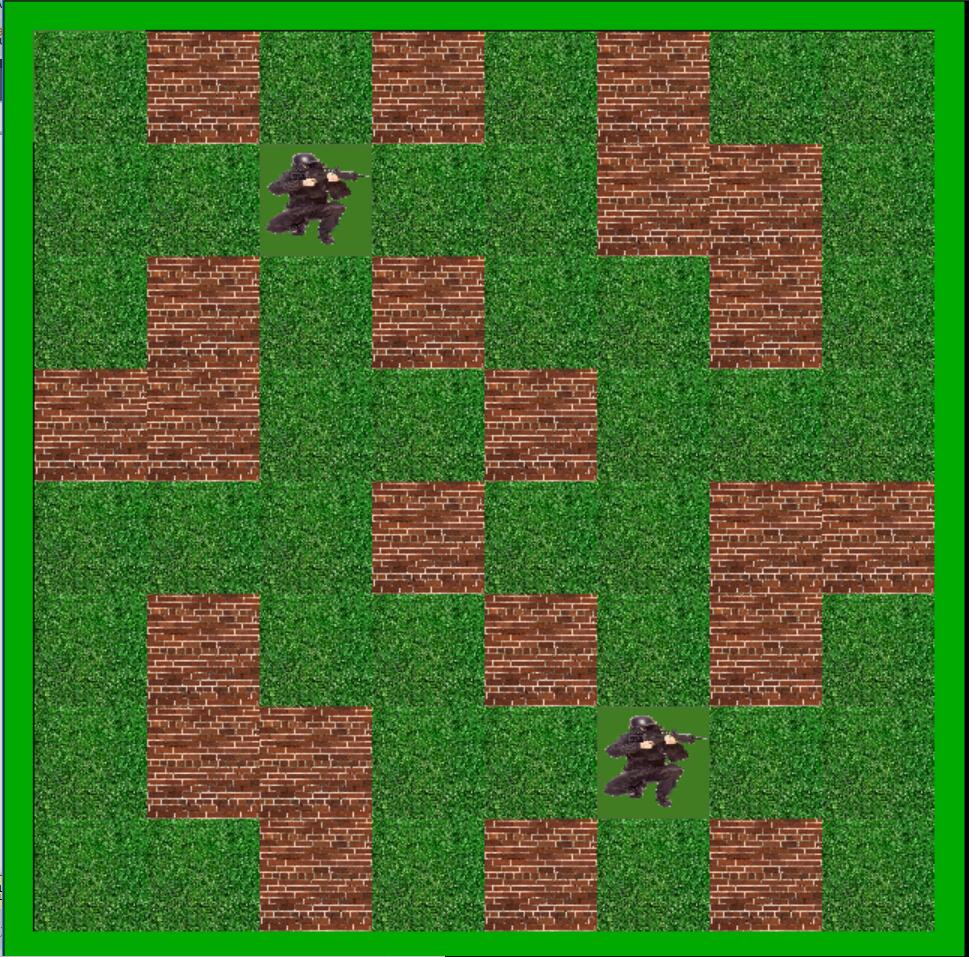




Problem 4: The NPC

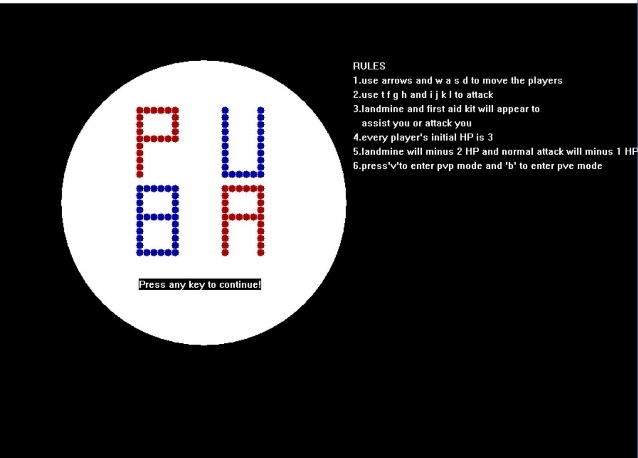
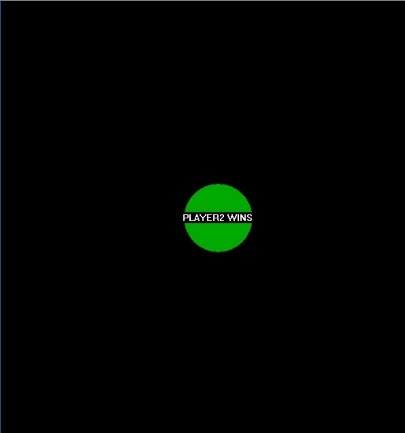
Constraint: The NPC is controlled by the computer, the basic movement and attack is just like the player2. When player1 has moved three steps, the computer will check the relative position of player1 and the NPC and then decide how to move.

When we design the NPC, when the NPC and the player are in the same line, the computer will check whether there is any obstacle between them and then the NPC will move if there is obstacle and attack if not. When they are not in the same line, the NPC will move according to the surroundings and the steps.



Problem 5: The whole game

Constraint:To make the game more organized, we have to use main function to start the game and end the game, and we should decide which button should we use to choose the mode (player vs player and player vs computer). And by the end, we should judge which player wins and show “playerx wins” on the screen.

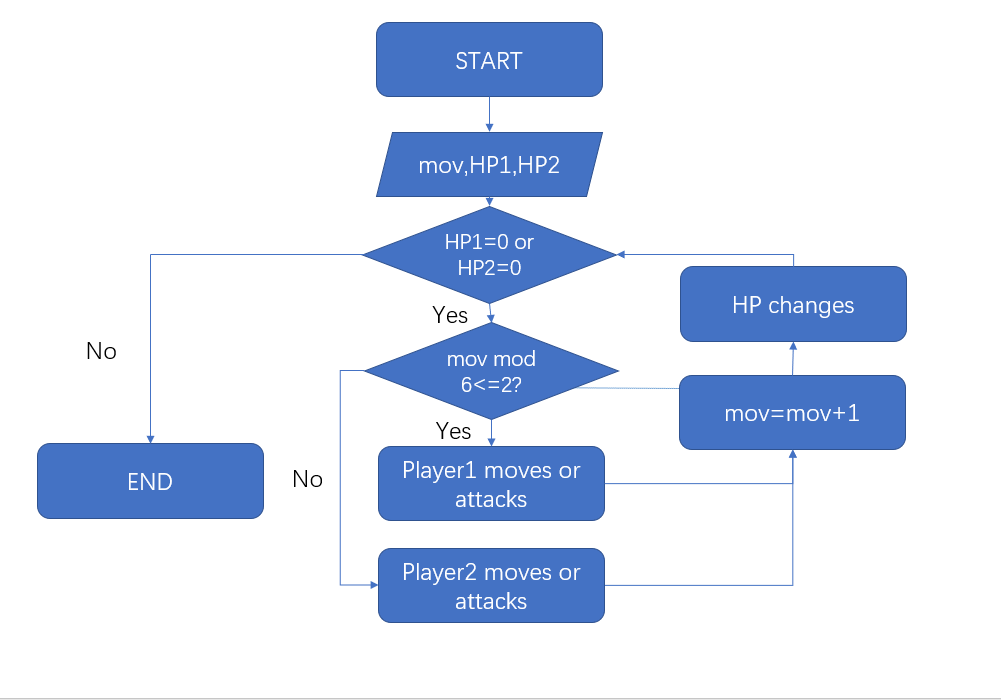
 We use switch sentence to choose mode. And to know which player wins the game, we have set the return type of the game’s body function integer. By assigning different values to the function, the computer will know which player wins.

Problem 6: The graphical interface

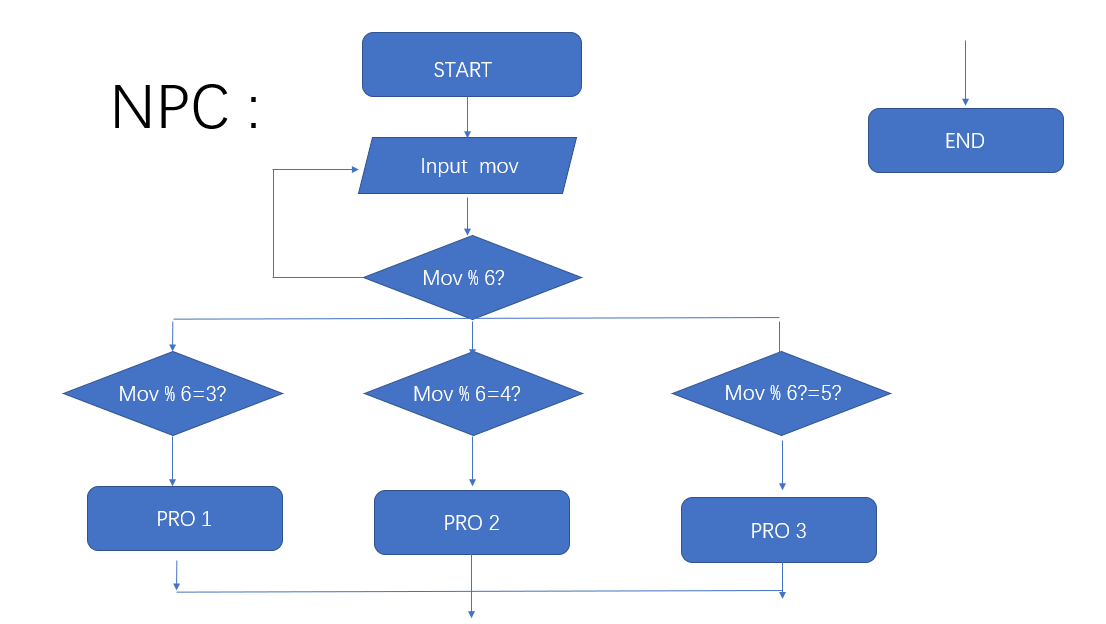
Constraint:The graphical interface should be simple and the arrangement of the wall should make the game more interesting while at the same time the game can’t be too difficult.

To draw the graph, we use the easyX to help us.

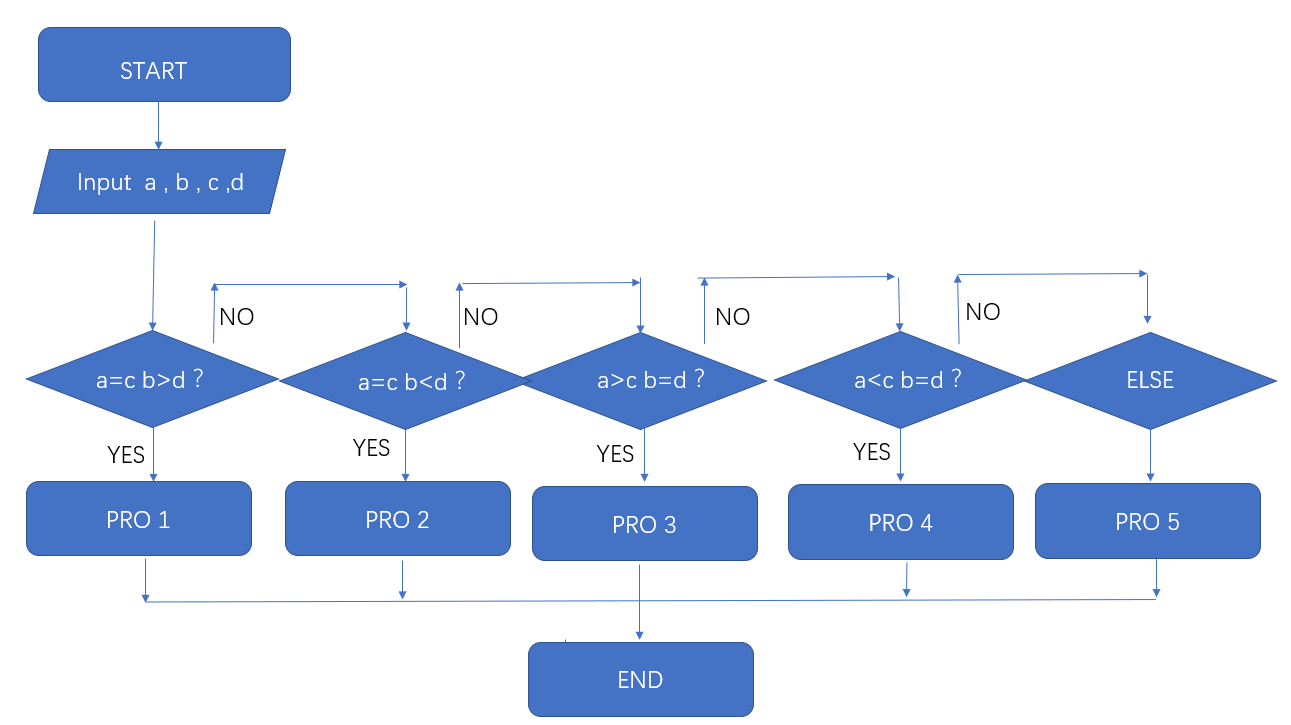
## Design



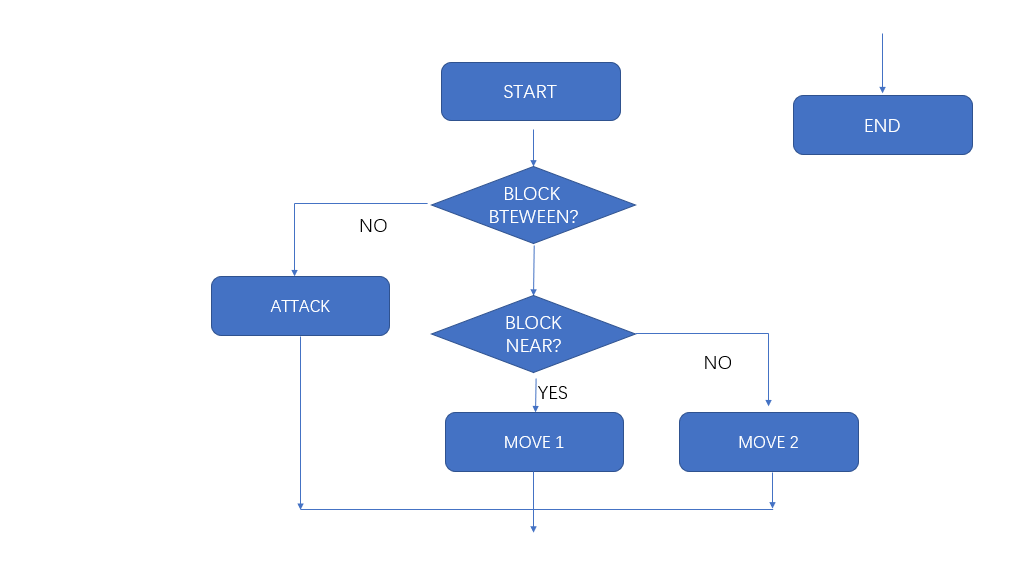
This is one of our basic application . to record number of rounds , ,we have to judge if the health points of players are zero .And then by comparing the size of mov we can get which player can move or attack .Finally ,it is time for the changing of HP and mov .



This is our part of NPC , because our game is a Turn-based game , we must to know when turns to NPC to move . so firstly , we input a variate called mov and take it the remainder of six . when the remainder is three , four or five , we know it time for NPC to move .



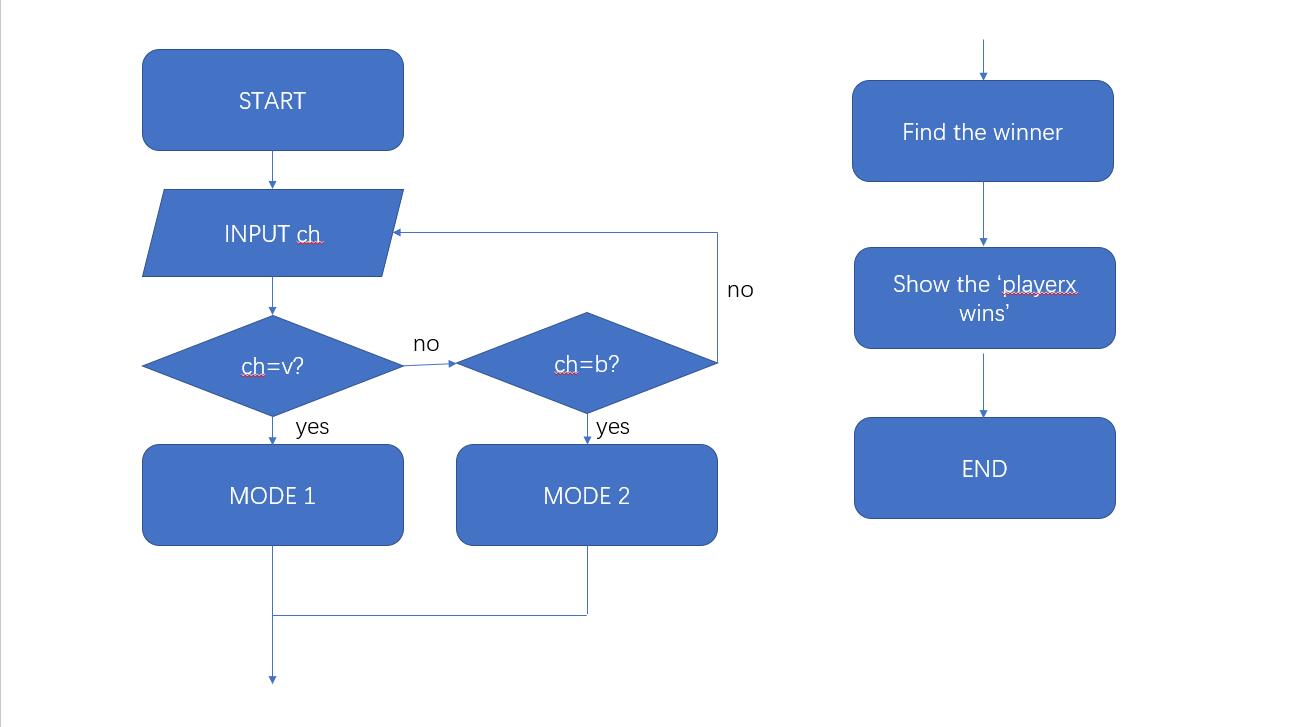
This flow chart is to judge the relative location between player and NPC . we input four variate a b c and d . they are coords of npc and player ,by comparing the size of them ,we get the relative location and proceed into next program .



in the last one , firstly we judge if there are blocks .if there are no blocks , we can let npc to attack player , but if yes . we have to judge if there are blocks near the npc and choose different ways to move .



This is our function of first-aid packet . Firstly , the first-aid packet will appear at any blank , by comparing the size of cords , we can increase the size of player’s HP and remove the first-aid packet .



This flow chart shows how our whole game works, we will check the input and when the input is ‘v’, we enter mode 1, and when the input is ‘b’ we enter mode 2, when there is some illegal input, the computer won’t respond. And when the game is over, the computer will find the winner and show the ‘playerx wins’ and the whole program is over.

|  |  |  |
| --- | --- | --- |
| TEST CHART | | |
| INPUT | EXPECT | RESULTS |
| ‘a’ | Player1 move upward when there is no obstacle and it’s its turn |  |
| ‘f’ | Player1 won’t move because it’s illegal. |  |
| up arrow | Player2 move upward when there is no obstacle and it’s its turn |  |
| ‘t’ and it’s player1’s turn | The bullet appears above player1 and keeps moving upward until it touched something |  |
| ‘t’ and it’s player2’s turn | The bullet won’t appear. |  |
| ‘i’ and it’s player2’s turn | The bullet appears above player2 and keeps moving upward until it touches something |  |
| Player1 attacks player2 successfully | HP2 minus1 |  |
| The landmine attacks player2 successfully | HP2 minus 2 |  |
| Round1 is over | The landmine should change the place and one first aid kit appears. |  |
| Round3 is over | 3 first aid kits should appear at three random places during the three rounds. |  |
| Player1 touches the first aid kit. | HP1 plus 1. |  |
| Press ‘s’ and ‘s’ and ‘d’ in the beginning of the NPC mode. | The NPC should move upward and then turn right and move forward |  |
| HP1=0 | The image of player1 disappear and then if you press any button, ‘PLAYER2 WINS’ will appear on the screen. |  |
| When player2 moves at Round2 and Round4 | The bomb will explode |  |
| When the bomb explodes. | There will be flame around the bomb and then everything around the bomb return to the original way. |  |
| When you are in the same line with the NPC | The NPC will shoot at you if it’s NPC’s first step or third step during this Round. |  |
| Press ‘a’ for five times | Player1 will move when it’s its turn and it can move and it won’t move when it’s not. |  |
| Press ‘t’ for five times | Player1 will attack when it’s its turn and won’t if not. |  |
| Press ‘v’ at the beginning of the game | You will enter the player vs player mode |  |
| Press ‘b’ at the beginning of the game | You will enter the player vs computer mode |  |
| Press ‘g’ at the beginning of the game | Nothing will happen |  |

## Implementation

1.Part of the movement code

case 't':

if (20 <= \*pa + 10 && \*pa + 10 <= 590 && arr[\*pa + 37][\*pc - 3] <= 98 && \*pc - 3 <= 590 && \*pc - 3 >= 20 && \*pmov % 6 <= 2)

{

for (up = 3; 20 <= \*pa + 10 && \*pa + 10 <= 590 && arr[\*pa + 37][\*pc - up] <= 98 && \*pc - up <= 590 && \*pc - up >= 20 && \*pmov % 6 <= 2; up++)

{

getimage(&background1, \*pa + 35, \*pc - up - 7, 5, 5);

putimage(\*pa + 35, \*pc - up - 2, &bullet);

Sleep(10);

putimage(\*pa + 35, \*pc - up - 2, &background1);

putimage(\*pd, \*pe, &img2);

}

\*pmov += 1;

if (arr[\*pa + 37][\*pc - up] == 99)

{

\*pHP2 = \*pHP2 - 1;

}

}

break;

2.Part of the attack code

case 's':

if (arr[\*pa + 37][\*pc + 37 + 75] == 96)

{

\*pHP1 += 1;

}

if (\*pc <= 545 && arr[\*pa + 37][\*pc + 37 + 75] <= 98 && \*pmov % 6 <= 2)

{

arr[\*pa + 37][\*pc + 37] = 0;

clearrectangle(\*pa, \*pc, \*pa + 74, \*pc + 74);

putimage(\*pa, \*pc, &background);

putimage(\*pa, \*pc + \*pb, &img1);

\*pa = \*pa;

\*pc = \*pc + \*pb;

arr[\*pa + 37][\*pc + 37] = 100;

\*pmov += 1;

}

else

{

putimage(\*pa, \*pc, &img1);

}

break;

3.The landmine code

void landmine(int \*pmov, int \*pHP1, int \*pHP2, int arr[640][640])

{

IMAGE flame, STORE;

IMAGE landmine1;

IMAGE wall, background;

int i, q;

loadimage(&wall, L"D:/Cprogram/timg2.jpg", 75, 75);

loadimage(&background, L"D:/Cprogram/timg3.jpg", 75, 75);

int wid[50], hei[50];

loadimage(&flame, L"D:/Cprogram/flame.jpg", 75, 75);

loadimage(&landmine1, L"D:/Cprogram/timg5.jpg", 75, 75);

q = \*pmov / 12;

//find the coordinate

for (i = 0; i <= 49; i++)

{

if (i % 2 == 0)

{

wid[i] = 3;

}

else

{

wid[i] = 4;

}

}

for (i = 0; i <= 49; i++)

{

if (i % 2 == 0)

{

hei[i] = 2;

}

else

{

hei[i] = 5;

}

}

if (\*pmov % 12 == 0) // put the landmine

{

putimage(wid[q] \* 75 + 21, hei[q] \* 75 + 21, &landmine1);

}

if (\*pmov % 12 == 11 && \*pmov >= 12) //the landmine explodes

{

getimage(&STORE, wid[q] \* 75 - 75 + 21, hei[q] \* 75 - 75 + 21, 225, 225);

putimage(wid[q] \* 75 - 75 + 21, hei[q] \* 75 + 21, &flame);

putimage(wid[q] \* 75 + 75 + 21, hei[q] \* 75 + 21, &flame);

putimage(wid[q] \* 75 + 21, hei[q] \* 75 - 75 + 21, &flame);

putimage(wid[q] \* 75 + 21, hei[q] \* 75 + 75 + 21, &flame);

Sleep(100);

putimage(wid[q] \* 75 - 75 + 21, hei[q] \* 75 - 75 + 21, &STORE);

putimage(wid[q] \* 75 + 21, hei[q] \* 75 + 21, &wall);

if (arr[wid[q] \* 75 + 75 + 21 + 37][hei[q] \* 75 + 21 + 37] == 99) //judge whether the bomb attacks the players.

{

\*pHP2 -= 1;

}

if (arr[wid[q] \* 75 - 75 + 21 + 37][hei[q] \* 75 + 21 + 37] == 99)

{

\*pHP2 -= 1;

}

if (arr[wid[q] \* 75 + 21 + 37][hei[q] \* 75 + 75 + 21 + 37] == 99)

{

\*pHP2 -= 1;

}

if (arr[wid[q] \* 75 + 21 + 37][hei[q] \* 75 - 75 + 21 + 37] == 99)

{

\*pHP2 -= 1;

}

if (arr[wid[q] \* 75 + 75 + 21 + 37][hei[q] \* 75 + 21 + 37] == 100)

{

\*pHP1 -= 1;

}

if (arr[wid[q] \* 75 - 75 + 21 + 37][hei[q] \* 75 + 21 + 37] == 100)

{

\*pHP1 -= 1;

}

if (arr[wid[q] \* 75 + 21 + 37][hei[q] \* 75 + 75 + 21 + 37] == 100)

{

\*pHP1 -= 1;

}

if (arr[wid[q] \* 75 + 21 + 37][hei[q] \* 75 - 75 + 21 + 37] == 100)

{

\*pHP1 -= 1;

}

}

}

4.The first aid kit code

void cure(int \*pmov, int arr[640][640])

{

int l;

int q1;

int store = 0;

int sub3[500], sub4[500];

int i, sub = 0;

int wid1[50], hei1[50] = {};

IMAGE bag1;

IMAGE wall, background;

loadimage(&wall, L"D:/Cprogram/timg2.jpg", 75, 75);

loadimage(&background, L"D:/Cprogram/timg3.jpg", 75, 75);

srand((unsigned)time(NULL));

for (i = 0; i <= 49; i++)

{

wid1[i] = i % 3 + 2;

}

//hei1[10] = {};

q1 = \*pmov / 12;

loadimage(&bag1, L"D:/Cprogram/timg6.jpg", 75, 75);

for (i = 0; i <= 499; i++)

{

sub3[i] = rand() % 8;

}

for (i = 0; i <= 499; i++)

{

sub4[i] = rand() % 8;

} //get the coordinate randomly

for (l = 0; l <= 49; l++)

{

for (i = store; arr[wid1[l] \* 75 + 21 + 37][hei1[l] \* 75 + 21 + 37] >= 98; i++)

{

wid1[l] = sub3[i];

hei1[l] = sub4[i];

}

store = i + 1;

}

if (arr[wid1[q1] \* 75 + 21 + 37][hei1[q1] \* 75 + 21 + 37] <= 97 && \*pmov % 12 == 0)

{

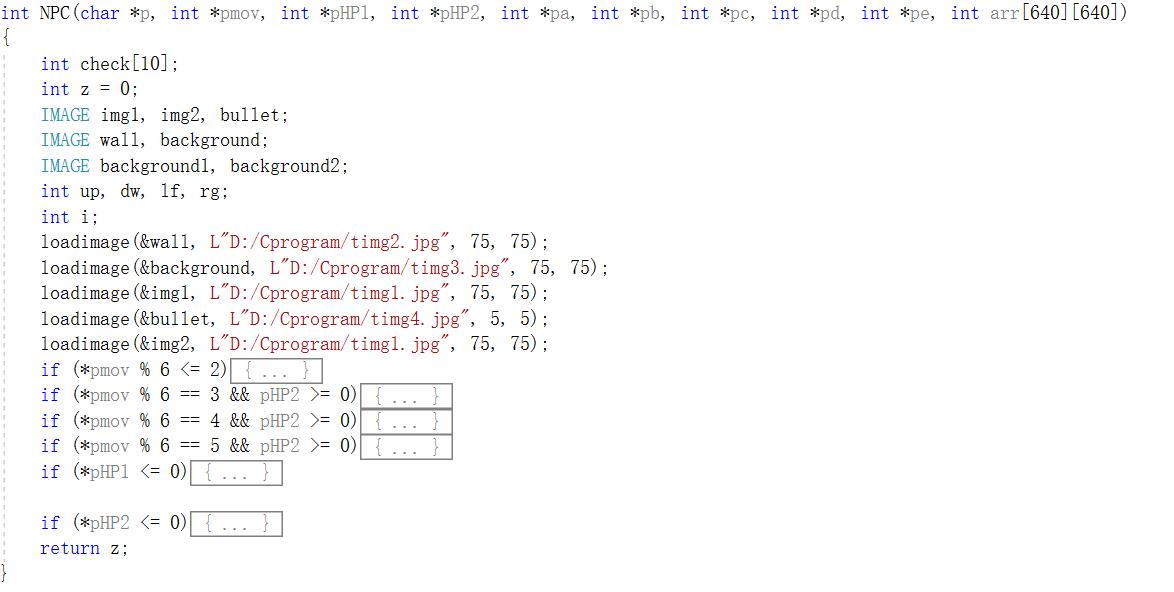
putimage(wid1[q1] \* 75 + 21, hei1[q1] \* 75 + 21, &bag1);

arr[wid1[q1] \* 75 + 21 + 37][hei1[q1] \* 75 + 21 + 37] = 96;

sub = q1;

}

}

5.Part of the NPC code

6. The whole game

int main()

{

int iw = 0;

int mov = 6;

int HP1 = 3, HP2 = 3;

int \*pHP1 = &HP1, \*pHP2 = &HP2;

int z = 0;

int \*pz = &z;

char chwhole;

int a, b, c, d, e;

int \*pa = &a, \*pb = &b, \*pc = &c, \*pd = &d, \*pe = &e;

int \*pmov = &mov;

typedef int arr640\_i\_t[640];

arr640\_i\_t \* arr = (arr640\_i\_t\*)malloc(640 \* sizeof(arr640\_i\_t));

arr[640][640] = {};

initgraph(900, 640);

a = 21;

b = 75;

c = 21;

d = 546;

e = 546;

startup();

while (HP1 != 0 && HP2 != 0)

{

if (\_kbhit())

{

chwhole = \_getch();

switch (chwhole)

{

case 'v':

closegraph();

initgraph(900, 640);

map(arr);

Whole(pmov, pHP1, pHP2, pa, pb, pc, pd, pe, arr);

z = Whole(pmov, pHP1, pHP2, pa, pb, pc, pd, pe, arr);

clearrectangle(1, 1, 640, 640);

break;

case 'b':

closegraph();

initgraph(900, 640);

map(arr);

Whole2(pmov, pHP1, pHP2, pa, pb, pc, pd, pe, arr);

z = Whole2(pmov, pHP1, pHP2, pa, pb, pc, pd, pe, arr);

clearrectangle(1, 1, 640, 640);

break;

}

}

}

end(pz);

}

6. The map

void map(int arr[640][640])

{

IMAGE wall, background;

int i, j, m, q;

loadimage(&background, L"D:/Cprogram/timg3.jpg", 75, 75);

for (m = 21, q = 75, i = 0; i <= 7; i++)

{

for (j = 0; j <= 7; j++)

{

putimage(m + i \* q, m + j \* q, &background);

}

}

loadimage(&wall, L"D:/Cprogram/timg2.jpg", 75, 75);

putimage(96, 21, &wall);

for (i = 95; i <= 170; i++)

{

for (j = 21; j <= 95; j++)

{

arr[i][j] = 101;

}

}

putimage(96, 171, &wall);

for (i = 95; i <= 170; i++)

{

for (j = 171; j <= 245; j++)

{

arr[i][j] = 101;

}

}

}

## Testing and Debugging

|  |  |  |
| --- | --- | --- |
| TEST CHART | | |
| INPUT | EXPECT | RESULTS |
| ‘a’ | Player1 will move upward when there is no obstacle and it’s its turn | Player1 moves upward when there is no obstacle and it’s its turn. However when you press it when its not player1’s turn, sometimes the first aid kit will appear. |
| ‘f’ | Player1 won’t move because it’s illegal. | Player1 doesn’t move because it’s illegal. |
| up arrow | Player2 will move upward when there is no obstacle and it’s its turn | Player2 moves upward when there is no obstacle and it’s its turn |
| ‘t’ and it’s player1’s turn | The bullet will appear above player1 and keeps moving upward until it touched something | The bullet appears above player1 and keeps moving upward until it touched something |
| ‘t’ and it’s player2’s turn | The bullet won’t appear. | The bullet doesn’t appear. |
| ‘i’ and it’s player2’s turn | The bullet will appear above player2 and keeps moving upward until it touches something | The bullet appears above player2 and keeps moving upward until it touches something |
| Player1 attacks player2 successfully | HP2 minus1 | HP2 minus1 |
| The landmine attacks player2 successfully | HP2 minus 2 | HP2 minus 2 |
| Round1 is over | The landmine should change the place and one first aid kit appears. | The landmine changes the place and one first aid kit appears. |
| Round3 is over | 3 first aid kits should appear at three random places during the three rounds. | 3 first aid kits appear at three random places during the three rounds. |
| Player1 touches the first aid kit. | HP1 plus 1. | HP1 plus 1. |
| Press ‘s’ and ‘s’ and ‘d’ in the beginning of the NPC mode. | The NPC should move upward and then turn right and move forward | The NPC moves upward and then turn right and move forward |
| HP1=0 | The image of player1 will disappear and then if you press any button, ‘PLAYER2 WINS’ will appear on the screen. | The image of player1 disappears and then if you press any button, ‘PLAYER2 WINS’ appears on the screen. However, in the NPC mode, sometimes, when HP1=0, the game is still operating and you have to do some other movements to end the game. |
| When player2 moves at Round2 and Round4 | The bomb will explode | The bomb explodes |
| When the bomb explodes. | There will be flame around the bomb and then everything around the bomb return to the original way. | There is flame around the bomb and then everything around the bomb return to the original way. |
| When you are in the same line with the NPC | The NPC will shoot at you if it’s NPC’s first step or third step during this Round. | The NPC shoots at you if it’s NPC’s first step or third step during this Round. |
| Press ‘a’ for five times | Player1 will move when it’s its turn and it can move and it won’t move when it’s not. | Player1 moves when it’s its turn and it can move and it doesn’t move when it’s not. However sometimes, the first aid kit will appear when it’s not player1’s round |
| Press ‘t’ for five times | Player1 will attack when it’s its turn and won’t if not. | Player1 attacks when it’s its turn and won’t if not. |
| Press ‘v’ at the beginning of the game | You will enter the player vs player mode | You enter the player vs player mode |
| Press ‘b’ at the beginning of the game | You will enter the player vs computer mode | You enter the player vs computer mode |
| Press ‘g’ at the beginning of the game | Nothing will happen | Nothing happens |

## Result & Conclusion

Our program can basically operate as we expect. Player will attack or move 3 times each round. Every player has 3 points of blood in the beginning. Once one player’s blood turns to zero, the game is over.

On the map medical kits appear randomly, which can enrich the blood by one point. And booms appears by turn in two fixed position, which explode every two rounds, deduct two points of blood of players around it. The functions of kits and booms’ appearing and playing works perfectly.

And the NPC will move as the player moves, but in the mode of fighting with NPC, there are no medical kits or booms on the map. The function of NPC needs to be better.

Sometimes, the NPC won’t act as we expect.

During the process of programming, we are confronted with many different problems and have a better understand of C language.

Firstly, we have tried many ways to solve the problem of how to make the computer know where is wall and where is empty place. We firstly choose to use the coordinate. But we find that if we want to add more tools such as the first aid kit into this game, it will be very hard to control. Having tried some other ways, we finally decide to use two-dimension array to assign some different values to different place, and then we can do almost everything by comparing the values. In this process, we have felt the error and trial process during creating a game, and what’s more, the importance of group work.

Secondly, there is still some problems about the point. So far, our game can only achieve some easy functions, but we have used a lot of points in one function, and with the project growing bigger, the connection between every function will become much more complicated. Although we have changed the return type of the function to reduce the number of points we have used, there are still a lot of points. The following code shows this problem.

int Attack1(char \*p, int \*pmov, int \*pHP1, int \*pHP2, int \*pa, int \*pb, int \*pc, int \*pd, int \*pe, int arr[640][640]);

int Whole(int \*pmov, int \*pHP1, int \*pHP2, int \*pa, int \*pb, int \*pc, int \*pd, int \*pe, int arr[640][640]);

int Whole2(int \*pmov, int \*pHP1, int \*pHP2, int \*pa, int \*pb, int \*pc, int \*pd, int \*pe, int arr[640][640]);

What’s more, there are still some problems in our program.

Firstly, sometimes when we press the button illegally, the first aid kit will appear. We have found the reason. It’s because each time we hit the keyboard, the program will run one time, and because there is no limitation on the production of the first aid kit, it will produce each time we hit the keyboard when it’s the time for the first aid kit to appear. To solve the problem, I think even we can’t limit the appearance of the first aid kit , we may be able to make the first aid kit appear at the same place when we press the illegal button, and by doing this, the first aid kit won’t appear randomly on the screen. The way to achieve this is to store the random number in some two other variables and we will get the random number outside (don’t in the function which need to hit the keyboard to run).

Second, sometimes, the judgement of the death sometimes is not that accurate. Sometimes, although the HP=0, the game won’t be over but will continue, will means the player can continue to attack, and at this time, the HP will still change and the result will change. This often happen in the player vs computer mode. To solve this problem, we should check the function carefully and to change the parameter to make the game more accurate.

Third, the NPC isn’t that smart. To make the NPC move much more smartly, we have to design different algorithm for the NPC when the situation is different. Now we only have roughly done the basic algorithm of several important situations. Therefore, to beat the NPC is pretty easy.

Considering all above, the accuracy of the judgement and the NPC function is two important question to be solved as these two question will influence the experience playing the game, as no one won’t errors when playing a game.