計算機網路實驗期末報告

大逃殺

405250436 蕭閔中

405250577 楊凱鈞

遊戲規則

- 四人遊玩
- 回合制
- 玩家每回合可選擇行動 :
    - 移動(不會超過前一個人，會被卡位)
    - 攻擊
    - 撿拾
    - 回血
- 移動及攻擊只能向前(順時鐘方向)
- 移動有最大步數限制
- 地圖每三回合會重置場上的裝備及藥包
- 玩家起始血量為 15
- 起始最大步數是 6 格
- 任何武器攻擊力都是 5
    - 小刀
        - 攻擊距離一格
        - 減少最大步數一格
    - 衝鋒槍
        - 攻擊距離三格
        - 減少最大步數兩格
    - 狙擊槍
        - 攻擊距離五格
        - 減少最大步數三格
    - 防彈衣
        - 被攻擊則代替血量被消耗
        - 減少最大步數一格
- 移動 > 拾取 > 攻擊 > 回血

Protocal

<地圖訊息>

“10ah202sx1/”

位置 武器 補包 玩家 結束

- 武器 :
  x 無
  k 小刀 m 衝鋒槍
  s 狙擊槍 a 防彈衣
- 補包 : x 無，h 有
- 玩家 : x 無，1 : gamer1…
- 解碼時迴圈一次跳五格直到遇到結束符/

<玩家狀態訊息>

“10510001”

血量 最大步數 武器 防彈衣數量 補包數量

- 武器 : 0 無，1 小刀
  2 衝鋒槍，3 狙擊槍
- 特例：死亡 “/”
- 勝利 “v”，失敗 “l”

<指令訊息>

- 攻擊指令：“a1/”
  a＝ 玩家 1or2or3or4
  1=攻擊指令
- 回血指令：“a2/”
- 撿拾指令：“a3c/”
  c＝1 撿武器，2 撿補包
- 移動指令： “a4c/”
  c＝ 移動步數
- 死亡： “/”

Structure

Sever 端

<struct map>

每一格地圖的狀態

1. Player = 哪個玩家
2. Equip1 = 甚麼武器
3. Equip2 = 補包

並宣告成 24 個元素的陣列

```
struct map
{
    int player=0;
    int equip1=0; //0=nothing, 1=knife, 2=machinegun, 3=sniper, 4=armor
    int equip2=0; //0=nothing, 1=health
};
map mapARRAY[24];
```

<struct gamer>

玩家的狀態

1. Alive = 該玩家是否死亡
2. Blood = 該玩家血量
3. Armor = 該玩家防彈衣數量
4. Maxwalk = 該玩家最大步數

5. Weapon = 該玩家所持武器

6. Locat = 該玩家的位置

7. Heal = 該玩家所持的補包數量

8. Walkespace = 該玩家可移動空間

並宣告成四個元素的陣列

```
struct gamer
{
    int alive = 1;
    int blood=15;
    int armor=0;
    int maxwalk=6;
    int weapon=0;
    int locat;
    int heal=0;
    int walkespace;
};
gamer gamerARRAY[4];
```

Client 端

<struct map>

每一格地圖的狀態

1. Player = 哪個玩家
2. Equip1 = 甚麼武器
3. Equip2 = 補包

並宣告成 24 個元素的陣列

```
struct map
{
    char *player = nothing;
    char *equip1 = nothing;
    char *equip2 = nothing;
};
map mapARRAY[24];
```

<struct situation>

玩家的狀態

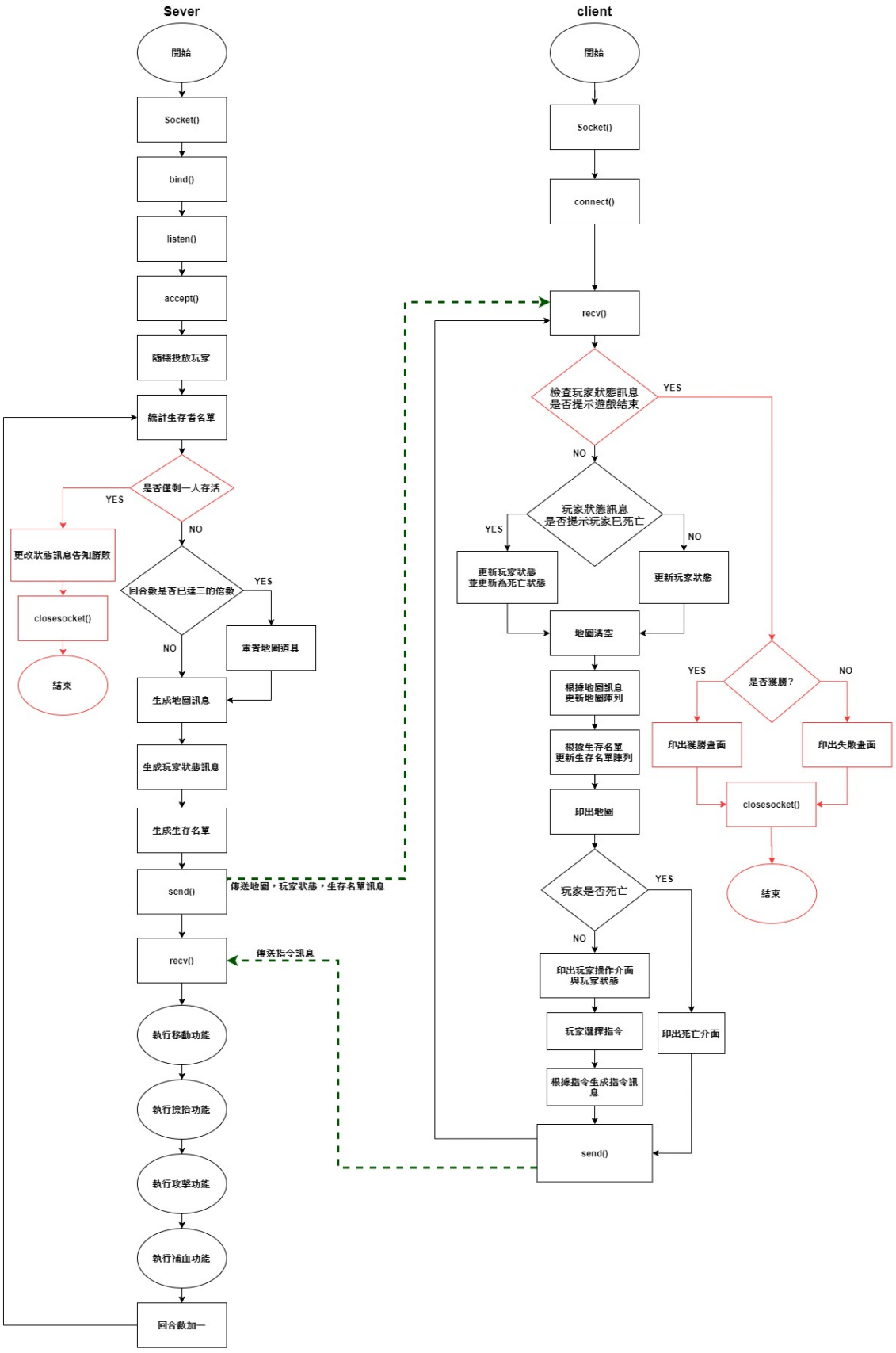
1. Alive = 該玩家是否死亡
2. Blood = 該玩家血量
3. Armor = 該玩家防彈衣數量
4. Maxwalk = 該玩家最大步數
5. Weapon = 該玩家所持武器
6. name = 該玩家的名字

7. Heal = 該玩家所持的補包數量

8. range = 該玩家可攻擊空間

```
struct situation
{
    char* name;
    int blood;
    int heal;
    int maxmove;
    int armor;
    char *weapon;
    int range = 0;
    int alive = 1;
};
situation situ;
```

遊戲流程


Sever
開始
Socket()
bind()
listen()
accept()
隨機投放玩家
統計生存者名單
是否僅剩一人存活
YES
NO
更改狀態訊息告知勝敗
closesocket()
結束
回合數是否已達三的倍數
YES
NO
重置地圖道具
生成地圖訊息
生成玩家狀態訊息
生成生存名單
send()
傳送地圖，玩家狀態，生存名單訊息
recv()
傳送指令訊息
執行移動功能
執行撿拾功能
執行攻擊功能
執行補血功能
回合數加一
client
開始
Socket()
connect()
recv()
檢查玩家狀態訊息
是否提示遊戲結束
YES
NO
玩家狀態訊息
是否提示玩家已死亡
YES
NO
更新玩家狀態
並更新為死亡狀態
更新玩家狀態
地圖清空
根據地圖訊息
更新地圖陣列
根據生存名單
更新生存名單陣列
印出地圖
玩家是否死亡
YES
NO
印出玩家操作介面
與玩家狀態
玩家選擇指令
根據指令生成指令訊息
send()
印出死亡介面
是否獲勝？
YES
NO
印出獲勝畫面
印出失敗畫面
closesocket()
結束

功能實現與問題解決

移動

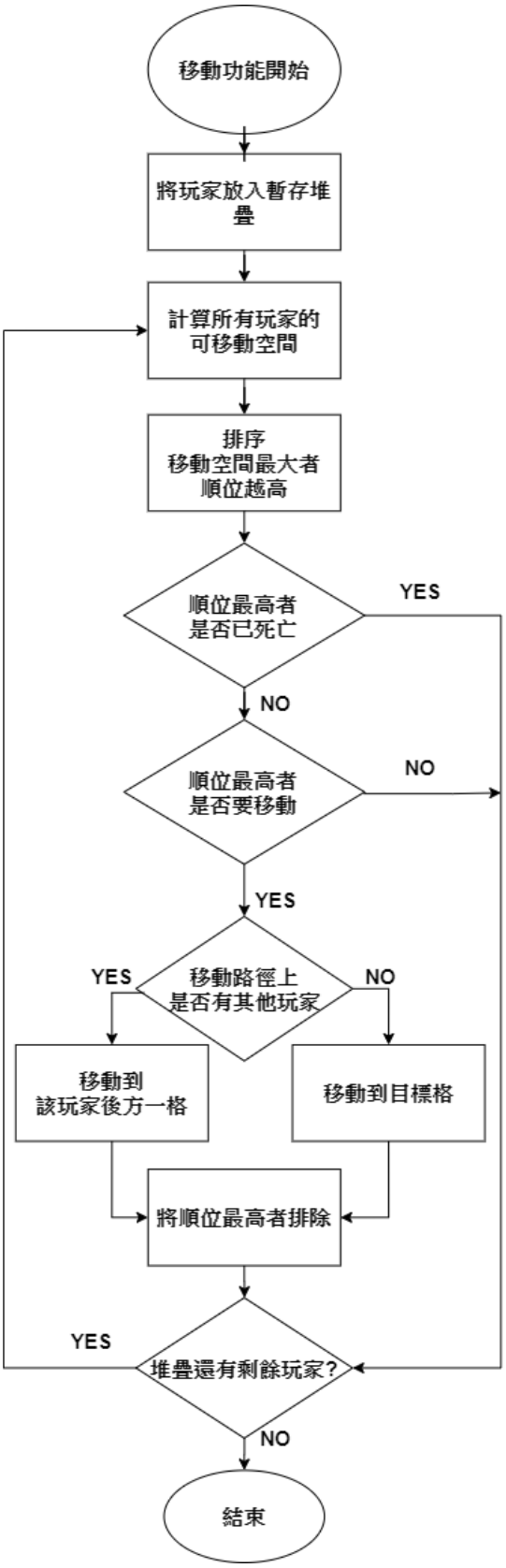
- 問題：

  因為不能超過前一個玩家，所以若多個玩家同時要移動，需要盡可能讓所有玩家可以移動到最多的步數

- 解決:

  1. 將所有玩家推入一個堆疊
  2. 將所有玩家與前方玩家的距離算出此為玩家的移動空間
  3. 根據每個玩家的移動空間排序(Bubble)，移動空間最大的在堆疊最上層
  4. 執行堆疊最上方的玩家並將該推出堆疊
  5. 從 2.開始重複直到堆疊空

流程


移動功能開始
將玩家放入暫存堆疊
計算所有玩家的可移動空間
排序
移動空間最大者順位越高
順位最高者是否已死亡
YES
NO
順位最高者是否要移動
NO
YES
移動路徑上是否有其他玩家
YES
NO
移動到該玩家後方一格
移動到目標格
將順位最高者排除
堆疊還有剩餘玩家?
YES
NO
結束

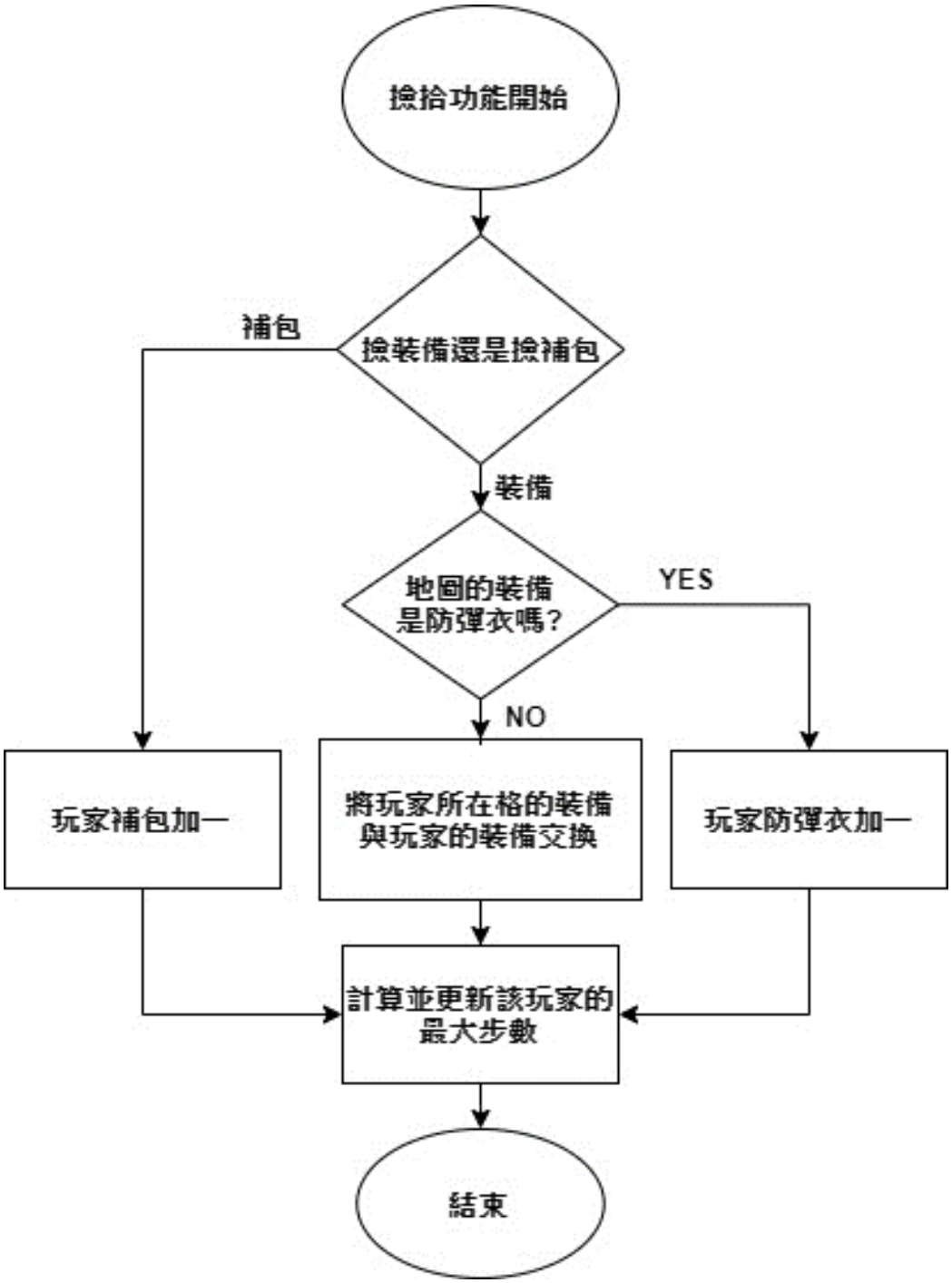
CODE

```
void movefunc()
{
     int moveARRAY[4],n,num1,num2,i,j,loc,max;
     int walknum,maxwalk;
     char* ptr;

     for (n = 0; n < 4; n++)
          moveARRAY[n] = n;

     for (max = 0; max < 4; max++)
     {
          for (n = max; n < 3; n++)//排序，第一個[0]的移動空間最大
          {
               for (i = max; i < 3; i++)
               {
                    num1 = moveARRAY[i];
                    num2 = moveARRAY[i + 1];
                    if (gamerARRAY[num1].walkespace < gamerARRAY[num2].walkespace)
                    {
                         moveARRAY[i + 1] = num1;
                         moveARRAY[i] = num2;
                    }
               }
          }

          i = moveARRAY[max];//i 是玩家陣列的陣列位置
          ptr = inst[i + 1];//陣列位置加一即是對應指令
          if (ptr[1] != '4') continue;

          loc = gamerARRAY[i].locat;
          walknum = str2int(ptr[2]);
          maxwalk = 6 - gamerARRAY[i].armor - gamerARRAY[i].weapon;

          if (walknum <= maxwalk)
          {
```

```
                for (j = 1; j <= walknum; j++)
                {
                     if (mapARRAY[(loc + j)%24].player)
                     {
                          walknum = j - 1;
                          break;
                     }
                }
                mapARRAY[loc].player = 0;//remove gamer from the block right now
                gamerARRAY[i].locat = (loc + walknum) % 24;//update the gamer location
by new location
                mapARRAY[gamerARRAY[i].locat].player = i + 1;//update map
           }
           spacecal();
     }

}
```

撿拾

流程


撿拾功能開始
撿裝備還是撿補包
補包
裝備
地圖的裝備
是防彈衣嗎?
YES
NO
玩家補包加一
將玩家所在格的裝備
與玩家的裝備交換
玩家防彈衣加一
計算並更新該玩家的
最大步數
結束

CODE

```
void takefunc()
{
     char* ptr;
     int n;
     for (n = 1; n < 5; n++)
     {
          ptr=inst[n];
          if (ptr[1] != '3') continue;
          int num = str2int(ptr[0]) - 1; //gamer's number "in array"
          int loc = gamerARRAY[num].locat;//gamer's location
          int stuff = str2int(ptr[2]);//what gamer want to take
          int temp;

          if (stuff == 1)//take eq1
          {
               if (mapARRAY[loc].equip1 == 4)//take armor
               {
                    gamerARRAY[num].armor += 1;//gamer get a armor
                    mapARRAY[loc].equip1 = 0;//remove armor from map
               }
               else //weapon
               {    //switch gamer's weapon & map's weapon
                    temp = gamerARRAY[num].weapon;
                    gamerARRAY[num].weapon = mapARRAY[loc].equip1;
                    mapARRAY[loc].equip1 = temp;
               }
          }
          else if (stuff == 2 && mapARRAY[loc].equip2)//take eq2
          {
               gamerARRAY[num].heal += 1;//gamer get a heal
               mapARRAY[loc].equip2 = 0;//remove eq2 from map
          }
          gamerARRAY[num].maxwalk = 6 - gamerARRAY[num].armor - gamerARRAY[num].weapon;
     }
}
```

攻擊

- 問題：

  攻擊須走直線，故轉角處須注意

- 解決:

  1. 檢查玩家下一格
  2. If(已經到角落&且沒其他玩家)

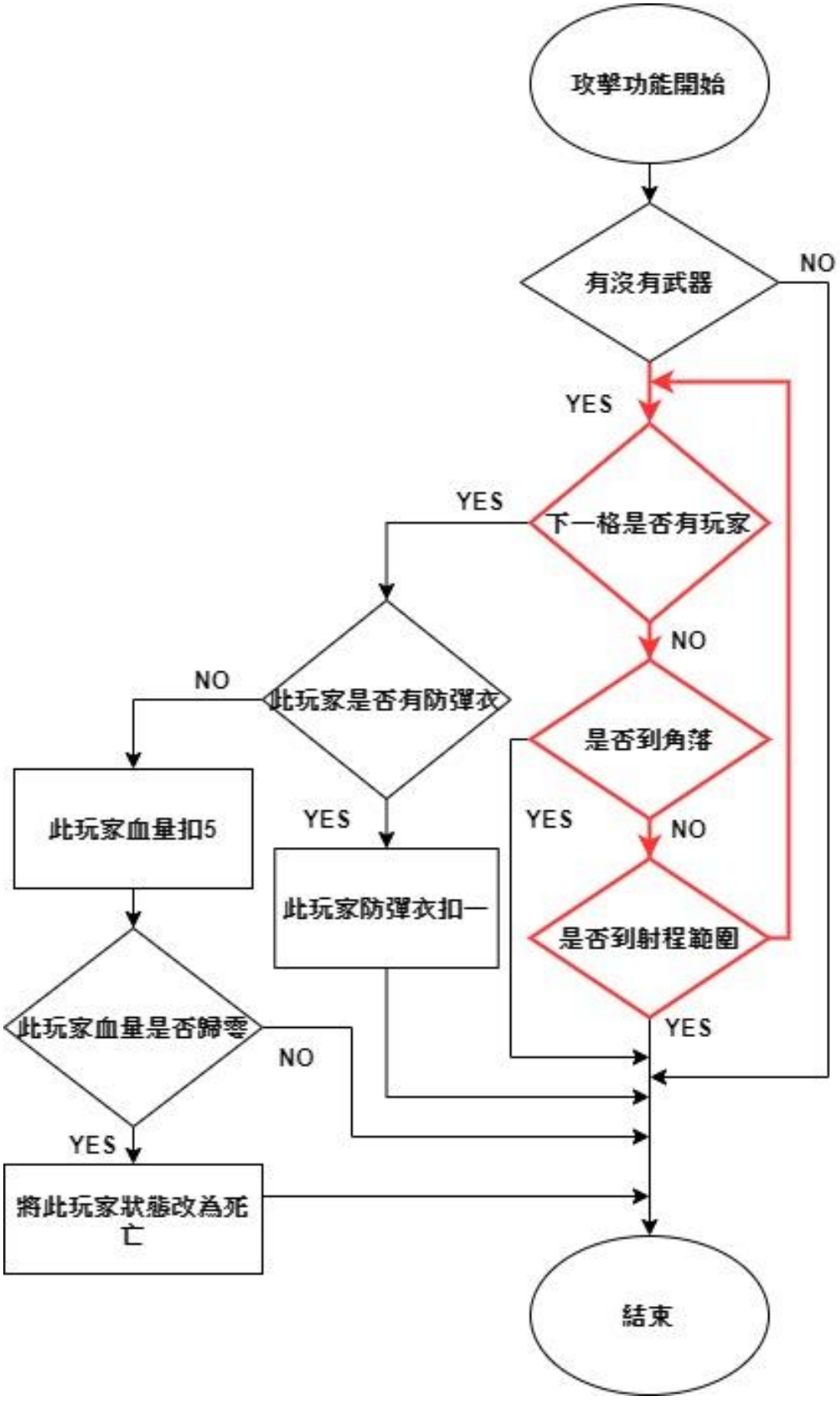
     Then break
  3. If(玩家存在)

     Then 執行攻擊
  4. 重複 1 直到已達攻擊範圍

CODE

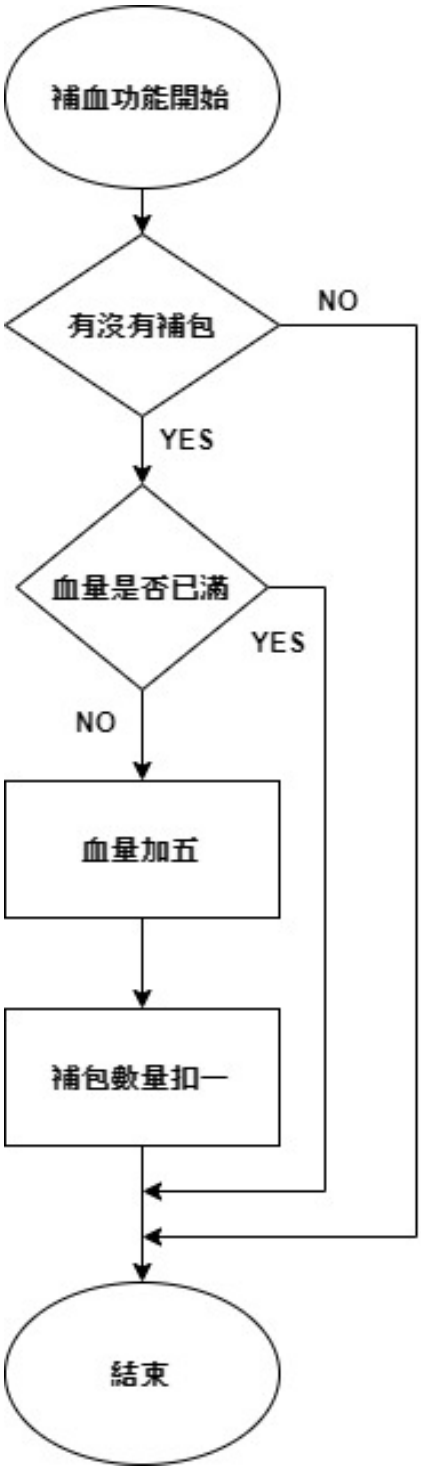
```
void attackfunc()
{
     char* ptr;
     int n;
     for (n = 1; n < 5; n++){
          ptr = inst[n];
          if (ptr[1] != '1') continue;
          int num = str2int(ptr[0]) - 1; //gamer's number "in array"
          int loc = gamerARRAY[num].locat;//gamer's location
          int weapon = gamerARRAY[num].weapon;
          if (weapon == 0) continue;//if no weapon don't continu
          int range = weapon * 2 - 1;
          int n, i, aim;
          for (n = 1; n <= range; n++){
               aim = (loc + n) % 24;
               if ((aim == 0 || aim == 6 || aim == 12 || aim == 18))
                    break; //if range touch the corner
               if (mapARRAY[aim].player  &&  mapARRAY[aim].player != str2int(ptr[0]))
               {    //find the enemy in the range except player
                    i = mapARRAY[aim].player - 1;
                    if (gamerARRAY[i].armor)//if enemy has armor
                         gamerARRAY[i].armor -= 1;
                    else
                    {
                         gamerARRAY[i].blood -= 5;
                         if (gamerARRAY[i].blood == 0)
                         {
                              gamerARRAY[i].alive = 0;
                              loc = gamerARRAY[i].locat;
                              mapARRAY[loc].player = 0;}
                    }
                    break;}
          }
     }
}
```

流程


攻擊功能開始
有沒有武器
NO
YES
下一格是否有玩家
YES
NO
此玩家是否有防彈衣
NO
是否到角落
此玩家血量扣5
YES
YES
NO
此玩家防彈衣扣一
是否到射程範圍
此玩家血量是否歸零
YES
NO
YES
將此玩家狀態改為死亡
結束

補血

流程

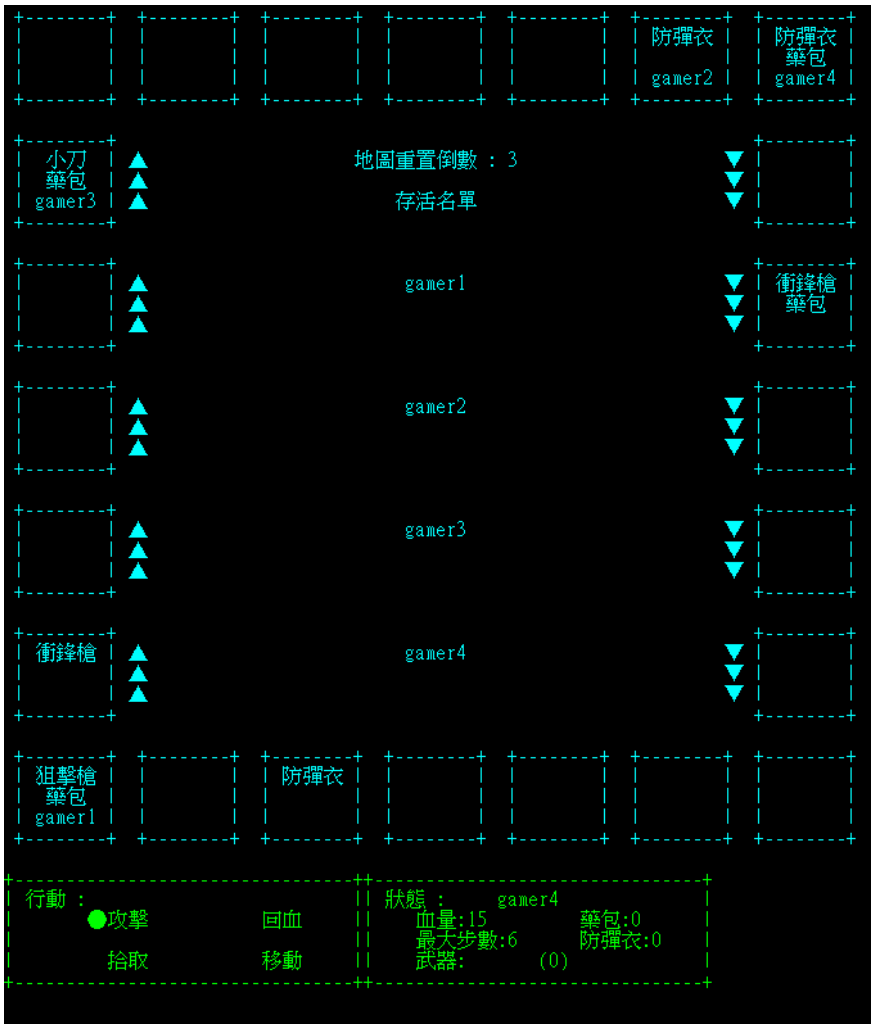


CODE

```
void healfunc()
{
     char* ptr;
     int n;
     for (n = 1; n < 5; n++)
     {
          ptr = inst[n];
          if (ptr[1] != '2') continue;
          int num = str2int(ptr[0]) - 1; //gamer's number "in array"
          if (gamerARRAY[num].heal && gamerARRAY[num].blood < 15)
          {    //if gamer has heal && blood is not full
               gamerARRAY[num].heal -= 1;
               gamerARRAY[num].blood += 5;
          }
          else
               continue;
     }
}
```

遊戲畫面


enter port :


防彈衣
gamer2
防彈衣
藥包
gamer4
小刀
藥包
gamer3
地圖重置倒數 : 3
存活名單
gamer1
衝鋒槍
藥包
gamer2
gamer3
衝鋒槍
gamer4
狙擊槍
藥包
gamer1
防彈衣
行動 :
攻擊
回血
拾取
移動
狀態 :
gamer4
血量:15
藥包:0
最大步數:6
防彈衣:0
武器:
(0)