

# 第三章 RPG遊戲

第二章知道遊戲流程是甚麼概念後，就可以製作非常多的回合制遊戲，而這次將要帶大家製作RPG遊戲了！！

## 1. 基本概念

先從遊戲本身的架構開始期待，玩家可以放技能、普攻、回血…之類的，而魔王也可以選擇普攻、放技能、大招…，說來抽象但其實貼切，最近很紅的星穹鐵道就是經典的回合制遊戲了



製作遊戲整體架構不變，我們一樣先從 遊戲設定 下手：

Coding C++

RUNMENU

Auto saved at 07:41:09

```
1 #include<bits/stdc++.h>
2 using namespace std;
3 int main() {
4     srand(time(0));
5     int php=100,patk=10;//pdef,pct
6     int bhp=1000,batk=20;//bdef,bct
7 }
```

第5行→ php 是 player\_HP 的縮寫，patk 是Player攻擊力的縮寫。  
第6行→ bhp 是 Boss\_HP 的縮寫，batk 是Boss攻擊力縮寫  
那其實上述的名詞可以換成自己喜歡的樣子，沒有太多致是規定

有了基本得的數值設定後，可以開始擬出遊戲介面和遊戲流程：

Coding C++RUNMENU

Auto saved at 09:01:12

```
1 #include<bits/stdc++.h>
2 using namespace std;
3 int main() {
4     srand(time(0));
5     int php=100,patk=10;//pdef,pct
6     int bhp=1000,batk=20;//bdef,bct
7     for(;;){
8         cout<<"我的血量："<<php<<endl;
9         cout<<"Boss HP："<<bhp<<endl;
10        char choise;
11        cout<<endl<<"普攻[A]/技能[S]/逃跑[D]："
12        cin>>choise;
13    }
14 }
```

第7行→ 這是一個for迴圈，會一直執行裡面的事情

第8&9行→ 輸出魔王和玩家的血量，<<endl代表換行(鼠標跳到下一行)

第10行→ 設定一個 choise 專門存放行動的

第11行→ 先換行是為了排版美觀，後面的文字則是可行動的選項

第12行→ 輸入東西進到 chiose

輸出結果：

```
我的血量：100
Boss HP：1000

普攻[A]/技能[S]/逃跑[D]：
```

介面遊戲設定好之後，就可以往下一步進到玩家行動以及電腦計算了

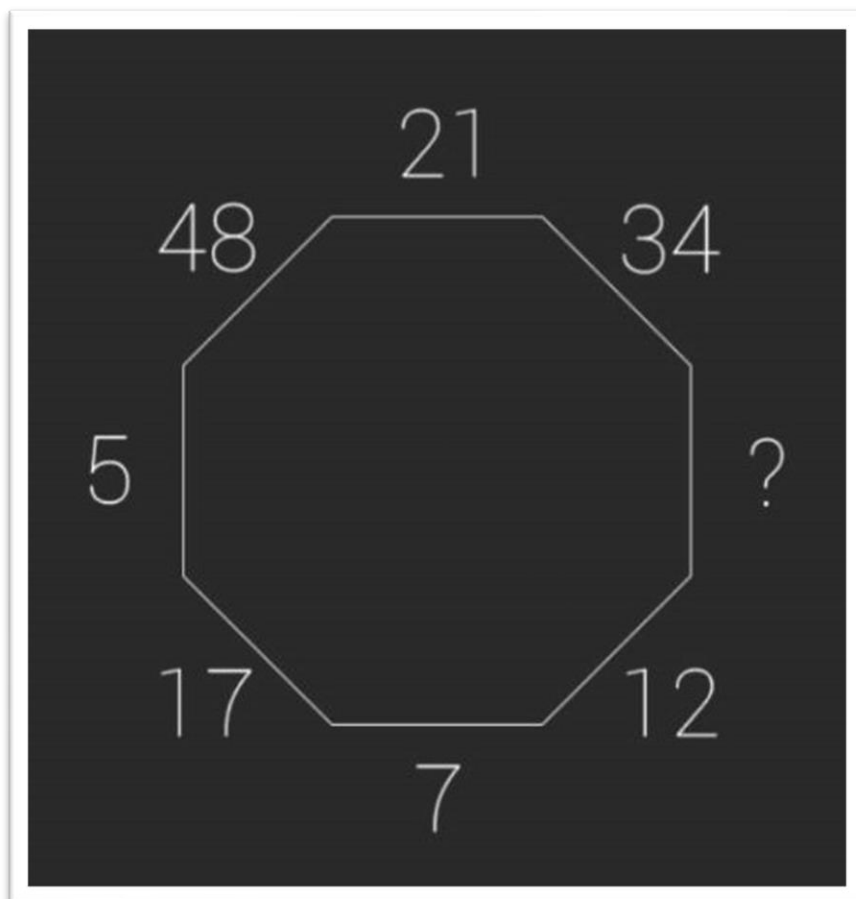
# GAME

$$\begin{array}{rcl}
 & - & 14 = \\
 + & & \div \\
 21 & & - & = & + & - \\
 = & & = & + & = & = \\
 45 & + & 2 = & & & 4 \\
 & + & & = & & \\
 13 - & = & 7 & 4 \\
 & = & & \\
 21 + & = & 28
 \end{array}$$

10	5	24	3	1	5
1	7	33	32	4	6

$$\begin{array}{rcl}
 30 & & 15 & & - & = & \\
 - & & + & & - & - & + \\
 & - & = & 8 & = & = & 2 \\
 = & + & = & = & = & = & \\
 7 - & = & & - & = & 10 & \\
 & = & & & & & \\
 + & = & 30 & & & & \\
 + & & & & & & \\
 27 & 15 \div & = & 5 & & & \\
 = & + & & & & & \\
 - & = & 31 & & & & \\
 & = & & & & & \\
 39 - & = & & & & & 
 \end{array}$$

1	11	6	4	14	5	5
3	16	1	3	14	10	16
41	34	3	27	26		



## 2. 製作互動

這類回合制的遊戲其靈魂大多都在技能和回合的精彩局，但萬事必須有個頭，那就是至少先從普通攻擊開始：

```
Coding C++
Auto saved at 13:56:19
RUN MENU

1 #include<bits/stdc++.h>
2 using namespace std;
3 int main() {
4     srand(time(0));
5     int php=100,patk=10;//pdef,pct
6     int bhp=1000,batk=20;//bdef,bct
7     for(;;){
8         cout<<"我的血量："<<php<<endl;
9         cout<<"Boss HP："<<bhp<<endl;
10        char choose;
11        cout<<endl<<"普攻[A]/技能[S]/逃跑[D]："；
12        cin>>choose;
13        if(choose=='A'){
14            cout<<"使用普通攻擊，造成"<<patk<<"的傷害"<<endl;
15            bhp=bhp-patk;
16        }
17        if(choose=='S'){}
18        if(choose=='D'){}
19    }
20 }
```

第 13&17&18 行→ 讓對應的英文符號產生動作，目前只有普攻

第 14 行→ 輸出造成的傷害資訊

第 15 行→ 更新血量資訊，把舊的魔王 HP 減去玩家攻擊力

輸出結果：

```
我的血量：100
Boss HP：1000

普攻[A]/技能[S]/逃跑[D]：A
使用普通攻擊，造成10的傷害
我的血量：100
Boss HP：990

普攻[A]/技能[S]/逃跑[D]：A
使用普通攻擊，造成10的傷害
我的血量：100
Boss HP：980

普攻[A]/技能[S]/逃跑[D]：
```

基本的輸出結果如附圖，會發現文字不易閱讀，所以可以適當加一些換行讓版面舒服一些些

有了玩家打魔王的流程，那魔王也要會對玩家造傷害，上一段程式和下一段程式其實都已經是電腦計算的步驟了，所以算完的結果直接輸出了：

```
Coding C++
Auto saved at 19:31:51
RUN MENU

1 #include<bits/stdc++.h>
2 using namespace std;
3 int main() {
4     srand(time(0));
5     int php=100,patk=10;//pdef,pct
6     int bhp=1000,batk=20;//bdef,bct
7     for(;;){
8         cout<<"-----"<<endl;
9         cout<<"我的血量："<<php<<endl;
10        cout<<"Boss HP："<<bhp<<endl;
11        char choose;
12        cout<<endl<<"普攻[A]/技能[S]/逃跑[D]："；
13        cin>>choose;cout<<endl;
14        if(choose=='A'){
15            cout<<"使用普通攻擊，造成"<<patk<<"的傷害"<<endl;
16            bhp=bhp-patk;
17        }
18        if(choose=='S'){
19        }
20        if(choose=='D'){
21            cout<<"Boss發動攻擊，造成"<<batk<<"的傷害"<<endl;
22            php=php-batk;
23        }
24    }
```

第 8 行→ 會發現在這裡多了分隔線，這是為了讓版面看起來舒服的關鍵

第 13 行→ 這行的後面多了換行，也可以讓版面看起來舒適

第 20 行→ 這行做了跟 15 行相同的輸出，讓玩家知道被摸到多痛

第 21 行→ 這行則是跟 16 行有相同的計算，只是對象相反

輸出結果：

```
-----
我的血量：100
Boss HP：1000

普攻[A]/技能[S]/逃跑[D]：A

使用普通攻擊，造成10的傷害
Boss發動攻擊，造成20的傷害
-----
我的血量：80
Boss HP：990

普攻[A]/技能[S]/逃跑[D]：
```

這樣有了分隔線後，分隔線之間就代表了「一個回合」，而且也可以清晰知道造成的傷害以及對魔王的傷害

# GAME

$\square + \square = 38$										$7 - \square = \square$									
$\square - \square = \square$										$\square + \square = \square$									
$\square \times \square = \square$										$\square - \square = \square$									
$\square = \square$										$\square = \square$									
16	$\square + \square = \square$									$\square - \square = 9$									
+	$\square + \square = 24$									$\square + \square = 9$									
=	$\square = \square$									$\square + \square = \square$									
$\square + \square = \square$										$8 \quad 14 + \square = \square$									
$\square - \square = \square$										$\square \times \square = \square$									
4	$- \square = 3$									$14 + \square = 21$									
$\square = \square$										$\square = \square$									
$\square + 19 = 37$										$3 + \square = 66$									

1	3	63	6	9	34	54	30	20
1	3	8	11	16	18	13	3	3
7	17	18	39	6	4	2	29	46

$\square + 3 = \square$																			
$\square - \square = \square$																			
$\square + \square = 30$										$\square + \square = 32$									
$\square = \square$										$\square = \square$									
16	$\times \square = 16$									$4 + \square = \square$									
=	$\square = \square$									$\square \times \square = \square$									
13	$- \square = 4$									$15 + \square = \square$									
$\square = \square$										$\square = \square$									
$\square + 14 = 40$																			

1	26	29	12	1	20	2
8	5	5	1	9	31	

$6 = 30$ $3 = 15$ $7 = 35$ $2 = ?$	$5, 3 = 28$ $7, 6 = 55$ $4, 5 = 21$ $3, 8 = ?$
---	---



### 3. 遊戲結束

如果細心的可以發現，當今天自己的血量歸零或是魔王血量歸零時，遊戲並沒有終止反而還在進行，所以接下來要添加一遊戲的終點。其實在選項裡有一個選項叫做「逃跑」，那也是要來結束遊戲的。

```
11     char choise;
12     cout<<endl<<"普攻[A]/技能[S]/逃跑[D] : ";
13     cin>>choise;cout<<endl;
14     if(choise=='A'){
15         cout<<"使用普通攻擊，造成"<<patk<<"的傷害"<<endl;
16         bhp=bhp-patk;
17     }
18     if(choise=='S'){
19         if(choise=='D'){
20             cout<<"你選擇了逃跑..."<<endl;
21             break;
22         }
23         cout<<"Boss發動攻擊，造成"<<back<<"的傷害"<<endl;
24         php=php-batk;
25         if(php<=0){
26             cout<<"你在這場戰鬥殞命..."<<endl;
27             break;
28         }
29         if(bhp<=0){
30             cout<<"英勇的瞬間即成傳說..."<<endl;
31             break;
32         }
33     }
34 }
```

第 19~22 行→ 當今天輸入 D 時，代表著逃跑

第 25~28 行→ 當玩家血量小於 0，代表玩家死了

第 29~32 行→ 當魔王血量小於 0，代表魔王死了

輸出結果：

```
-----
我的血量：20
Boss HP：960
```

```
普攻[A]/技能[S]/逃跑[D]：A
```

```
使用普通攻擊，造成10的傷害
Boss發動攻擊，造成20的傷害
你在這場戰鬥殞命...
```

```
[Process completed - press Enter]
```

```
-----
我的血量：100
Boss HP：1000
```

```
普攻[A]/技能[S]/逃跑[D]：D
```

```
你選擇了逃跑...
```

```
[Process completed - press Enter]
```

# GAME

21 +  =

+  +

-  = 2       + 12 =

=  =  +  -

26  ÷  = 5

=  =

9 +  =        +  = 18

-  ×

+  = 7      12 ÷  = 12

-  =  =  ×

1  - 13 =

=  =

3      29 - 21 =

-  = 13       + 14 =

-  -

- 6 =        ×  =

=  =  =  =

21  ÷  =        =  -

-        1

÷ 6 =       =

=

-  = 3

15	2	25	8	1	8	21
3	4	1	45	12	3	24
2	4	17	6	10	3	

4	3	28	15	12	24	2	28
1	18	2	7	2	9	4	2
12	12						

6			
7	5		
11	3	?	
20	2	4	10

8, 2 = 20

12, 3 = 40

10, 5 = 52

6, 1 = ?



## 4. 技能

回合制的遊戲，最豐富多樣的的就是技能，那在右戲中多的時代技能也是千千萬萬種，接下來會介紹這麼多技能都怎麼運作。

壹、【強化攻擊】	將 patk 乘上倍數，造成傷害後變回 1 倍
貳、【多段攻擊】	一般攻擊變很多次，可以有自己的倍率和打的次數
參、【回血】	可以直接加數值，或是乘上特定倍律
肆、【斬殺】	計算已損血量給予傷害
伍、【吸血】	將計算傷害時，將傷害用一定倍率加回自己得 HP
陸、【獻祭】	減少自己 HP，變成額外傷害或是強化攻擊
柒、【格檔免傷】	在 Boss 回合需減除傷害，目前架構較難實現
捌、【中毒】	用計數器紀錄次數，每回合計算一次，目前較難實現
玖、【暈眩】	用計數器紀錄次數，每回合計算一次，目前較難實現

每項都會提供片段程式碼，只需要把程式安插在是當的位子即可，如下：

```
4      if(choise=='A'){
5          cout<<"使用普通攻擊，造成"<<patk<<"的傷害"<<endl;
6          bhp=bhp-patk;
7      }
8      if(choise=='S'){
9          cout<<"技能一[A]/技能二[S]/技能三[D]："<<endl;
10         char skill;cin>>skill;
11         if(skill=='A'){
12             cout<<"施放技能一"<<endl;
13         }
14         if(skill=='S'){
15             cout<<"施放技能二"<<endl;
16         }
17         if(skill=='D'){
18             cout<<"施放技能三"<<endl;
19         }
20     }
21     if(choise=='D'){
22         cout<<"你選擇了逃跑..."<<endl;
23         break;
24     }
```

第 18 行：當今天先按下 S 時會進到技能列表（第 19 行）

第 20 行：skill 會記錄按下的按鍵

第 21～23 行：按下 A 會走到技能一

第 21～23 行：按下 S 會走到技能二

第 21～23 行：按下 D 會走到技能三

接下來將要逐一介紹先前表格所列舉的技能

#### 壹、【強化攻擊】

```
21     if(skill=='A'){
22         cout<<"玩家施放重擊，造成"<<patk*25<<"的傷害"<<endl;
23         bhp=bhp-patk*25;
24     }
```

第 22 行→  $patk * 25$  中，乘上的數字可以輸入自己喜歡的算式

第 23 行→ 在計算魔王血量(bhp)時需要跟上行的數字相同

#### 貳、【多段攻擊】

```
25     if(skill=='S'){
26         cout<<"玩家施放流星斬"<<endl;
27         for(int t=0;t<7;t++){
28             int damage=patk*0.5+rand()%5;
29             if(rand()%100<50)damage=damage*3;
30             cout<<"造成"<<damage<<"的傷害"<<endl;
31             bhp=bhp-damage;
32         }
33     }
```

第 26 行→ 輸出技能名稱，可以自己修改喜歡的樣子

第 27 行→ 連續打 7 下，可以自己改喜歡的數字

第 28 行→ 將單下傷害設成 damage，可以自己改算式

第 29 行→  $if(rand()\%100<50)$  表示有 50%的機率產生 3 倍傷害

第 30 行→ 輸出打單下多少

第 31 行→ 每下對魔王的傷害都要扣除

#### 參、【回血】

```
34     if(skill=='D'){
35         //cout<<"玩家施放恢復，恢復"<<(100-php)*0.5<<"點HP"<<endl;
36         cout<<"玩家施放治療，恢復自己"<<30+rand()%40<<endl;
37         //php=php+(100-php)*0.5;
38         php=php+30+rand()%40;
39     }
```

第 36 行→ 恢復 30~70 HP 的計算式，可以自己更改成其他的算式

第 38 行→ 恢復的數值加回自己的血量

第 35 行→ 這行計算式是：回復已損 HP 的 50%

第 37 行→ 同第 38 行

<pre> ----- 我的血量：100 Boss HP：1000  普攻[A]/技能[S]/逃跑[D]：S  技能一[A]/技能二[S]/技能三[D]：A 玩家施放重擊，造成250的傷害 Boss發動攻擊，造成20的傷害 ----- 我的血量：80 Boss HP：750 </pre>	<pre> ----- 我的血量：60 Boss HP：500  普攻[A]/技能[S]/逃跑[D]：S  技能一[A]/技能二[S]/技能三[D]：D 玩家施放治療，恢復自己49 Boss發動攻擊，造成20的傷害 ----- 我的血量：94 Boss HP：500  普攻[A]/技能[S]/逃跑[D]： </pre>
<pre> ----- 我的血量：100 Boss HP：1000  普攻[A]/技能[S]/逃跑[D]：S  技能一[A]/技能二[S]/技能三[D]：S 玩家施放流星斬 造成15的傷害 造成21的傷害 造成6的傷害 造成5的傷害 造成18的傷害 造成5的傷害 造成27的傷害 Boss發動攻擊，造成20的傷害 ----- 我的血量：80 Boss HP：903 </pre>	<p>這些就是技能大概會呈現的效果，在數值公式、放技能的文字都是可以改成自己喜歡的樣子</p>

玩家的三項技能設計好了之後，可以設計魔王的攻擊模式，如下：

<pre> 46   if(rand()%100&lt;30){ 47       cout&lt;&lt;"Boss發動攻擊，造成"&lt;&lt;batk&lt;&lt;"的傷害"&lt;&lt;endl; 48       php=php-batk; 49   } 50   else{ 51       int sk=rand()%100; 52       if(sk&gt;70){ 53           cout&lt;&lt;"魔王發動技能一"&lt;&lt;endl; 54       } 55       else if(sk&lt;40){ 56           cout&lt;&lt;"魔王發動技能二"&lt;&lt;endl; 57       } 58       else{ 59           cout&lt;&lt;"魔王發動技能三"&lt;&lt;endl; 60       } 61   } </pre>	<p>普攻機率：30%</p> <p>技能：70%</p> <p>    技能一：30%</p> <p>    技能二：40%</p> <p>    技能三：30%</p>
<p>第 46 行→ 普攻機率設定為 30%，可以自己調整</p> <p>第 50 行→ else 代表另外的 70% (技能)</p> <p>第 51 行→ sk 代表隨機取 0-100 一個數字用來機率判斷</p> <p>第 52,55,58 行→ 在右上角有技能施放機率的分配，一樣可以隨意修改</p> <p>if(sk&gt;70)是 70-100 的區間，else if(sk&lt;40)是 0-40 的區間，else 指剩餘的</p>	

#### 肆、【斬殺】

```
52     if(sk>70){
53         cout<<"魔王發動【斬殺】，造成"<<(100-php)*0.16<<endl;
54         php=php-(100-php)*0.16;
55     }
```

第 53 行→ 發動技能一的文字，後面接傷害計算公式，可以改成自己喜歡的  
第 54 行→  $(100 - \text{php}) * 0.16$  的意思是已損血量的 16%

#### 伍、【吸血】

```
56     else if(sk<40){
57         cout<<"魔王發動【噬魂】"<<endl;
58         cout<<"吸收玩家"<<(1000-bhp)*0.05+10<<"點傷害" <<endl;
59         cout<<"恢復自己"<<(1000-bhp)*0.1+20<<"點HP" <<endl;
60         php=php-(1000-bhp)*0.05-10;
61         bhp=bhp+(1000-bhp)*0.1+20;
62     }
```

第 57 行→ 發動技能二的文字，可以改成自己喜歡的  
第 58&59 行→ 輸出技能影響的數值變化，一樣可以自己調成喜歡的樣子  
第 60&61 行→ 運算技能的傷害，更新數值，傷害公式可以自己調整

#### 陸、【獻祭】

```
63     else{
64         cout<<"魔王發動【血願】，獻祭自己一半的生命"<<endl;
65         cout<<"造成"<<bhp/10<<"的傷害"<<endl;
66         php=php-bhp/10;
67         bhp=bhp/2;
68     }
```

第 64 行→ 發動技能三的文字，可以改成自己喜歡的  
第 65 行→ 輸出技能影響的數值變化，一樣可以自己調成喜歡的樣子  
第 66&67 行→ 運算技能的傷害，更新數值，傷害公式可以自己調整

肆、【斬殺】	伍、【吸血】	陸、【獻祭】
<pre>----- 我的血量：35 Boss HP：880  普攻[A]/技能[S]/逃跑[D]：A  使用普通攻擊，造成10的傷害 魔王發動【斬殺】，造成10.4 ----- 我的血量：24 Boss HP：870  普攻[A]/技能[S]/逃跑[D]：A</pre>	<pre>----- 我的血量：67 Boss HP：522  普攻[A]/技能[S]/逃跑[D]：A  使用普通攻擊，造成10的傷害 魔王發動【噬魂】 吸收玩家34.4點傷害 恢復自己68.8點HP ----- 我的血量：32 Boss HP：580  普攻[A]/技能[S]/逃跑[D]：</pre>	<pre>----- 我的血量：83 Boss HP：702  普攻[A]/技能[S]/逃跑[D]：A  使用普通攻擊，造成10的傷害 魔王發動【血願】，獻祭自己一半的生命 造成69的傷害 ----- 我的血量：14 Boss HP：346  普攻[A]/技能[S]/逃跑[D]：</pre>

# GAME

										4	-		=	
										-		+		-
3	-		=				7							
		+		=		=		=						
			+		=				2					
		=												
19		7	+		=			8		26				
-				-		-		÷		-				
					12		16	-		=				
=					=		=		=	=				
	-	7	=				×		=	14				

16	1	11	1	2	8	7	1
3	4	2	6	23	8	12	2
4							

										3	+		=	
										+		-		
27	-		=				12		80					
-				=		=		-						
					×		=							
=								=		+				
	-		=	1		6	÷		=					
									=					
						1	+	7	=					

3	2	2	3	8	78	6
25	1	25	28	5	13	24

42 = 58
64 = 36
12 = 88
77 = ?

3, 4, 6 = 2412
5, 7, 1 = 735
9, 2, 8 = 1618
6, 7, 4 = ?