# 泡泡龙任务2

## 1.1 任务要点

* 泡泡的发射功能
* 泡泡的反弹功能
* 泡泡的黏附功能

## 1.2 任务内容

* 泡泡的发射功能：发射出一个泡泡后，必须等泡泡爆掉或黏附于其他泡泡或在顶端，才能发下一个泡泡。
* 泡泡的反弹功能：可利用两侧垂直的墙壁进行反弹。
* 泡泡的黏附功能：泡泡飞行途中边界与其他泡泡相触，即被黏附，并停止运动。位置由入射角和原来静止的泡泡位置共同决定。

## 1.3 任务实现

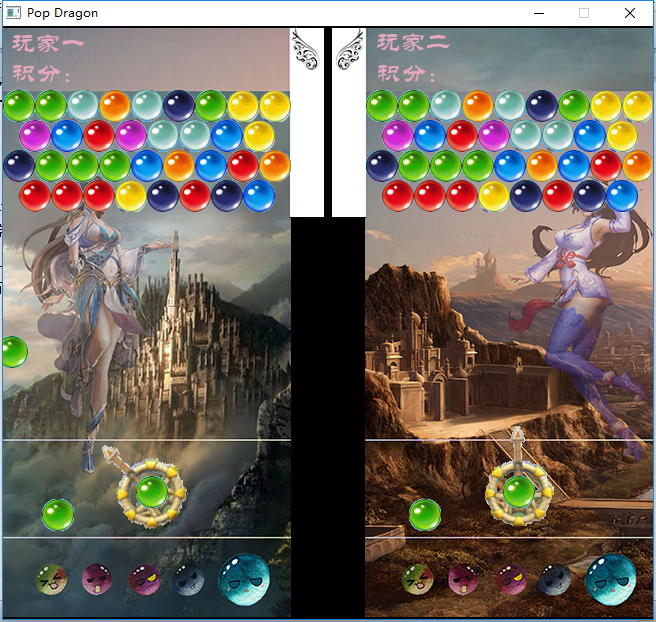
**流程图：**

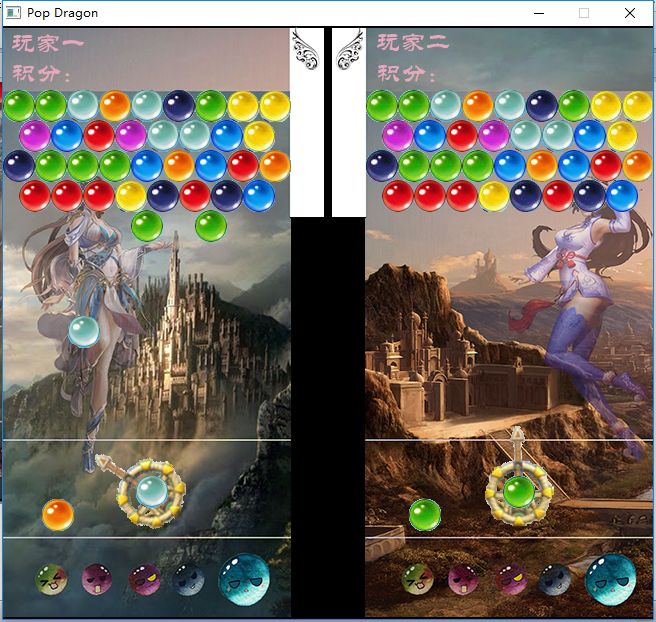
**泡泡发射：弓箭可以转向任意角度，玩家一按“a”，“d”控制弓箭方向，玩家二按“h”，“k”控制方向，玩家一按空格键发射泡泡，玩家二按‘；’发射泡泡**

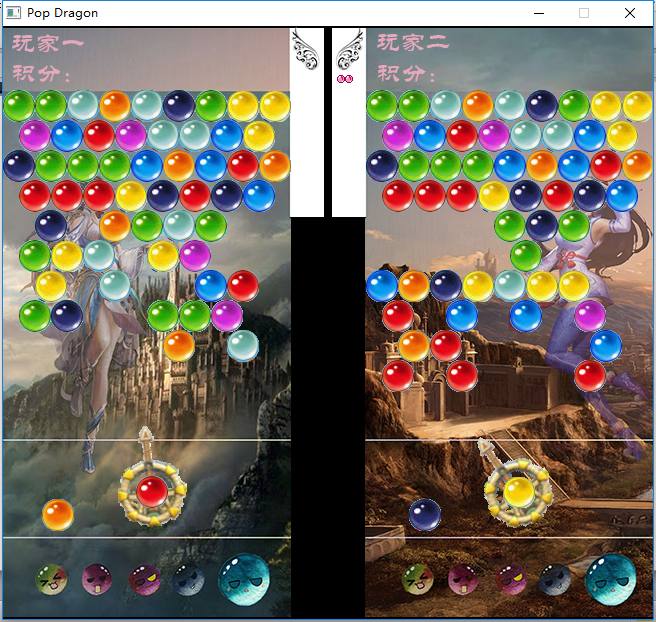
**泡泡反弹：泡泡碰到边界处，横向运行方向反向。**

**泡泡粘附：当泡泡快要碰到已有的泡泡时，定下发射泡泡的位置，标记该位置**

**实现效果：**







**核心代码：**

泡泡发射和反弹：

while(kbhit())

{

k = getch();

switch(k)

{

case ';':///右边发射键

{

int n4,m4;

n4 = angle[p][0];

m4 = angle[p][1];

while(!Judgement(n4,pos,b,p,q))

{

if(b[q].x<=363)

{

n4 \*= -1;

b[q].x = 363;

putimage\_transparent(NULL, a[b[q].bubble],b[q].x,b[q].y,BLACK);

}

if(b[q].x>=618)

{

n4 \*= -1;

b[q].x = 618;

putimage\_transparent(NULL, a[b[q].bubble],b[q].x,b[q].y,BLACK);

}

b[q].x+=n4;

b[q].y+=m4;

putimage\_transparent(NULL, a[b[q].bubble],b[q].x,b[q].y,BLACK);

Sleep(20);

Cover\_Back();

// if(KO==0)

// goto end1;

}

q++;

break;

}

case 'h':

{

p--;

if(p<0) p = 32;

break;

}

case 'k':

{

p++;

if(p>32) p = 0;

break;

}

while(kbhit())

{

k = getch();

switch(k)

{

case ';':///右边发射键

{

int n4,m4;

n4 = angle[p][0];

m4 = angle[p][1];

while(!Judgement(n4,pos,b,p,q))

{

if(b[q].x<=363)

{

n4 \*= -1;

b[q].x = 363;

putimage\_transparent(NULL, a[b[q].bubble],b[q].x,b[q].y,BLACK);

}

if(b[q].x>=618)

{

n4 \*= -1;

b[q].x = 618;

putimage\_transparent(NULL, a[b[q].bubble],b[q].x,b[q].y,BLACK);

}

b[q].x+=n4;

b[q].y+=m4;

putimage\_transparent(NULL, a[b[q].bubble],b[q].x,b[q].y,BLACK);

Sleep(20);

Cover\_Back();

// if(KO==0)

// goto end1;

}

q++;

break;

}

case 'h':

{

p--;

if(p<0) p = 32;

break;

}

case 'k':

{

p++;

if(p>32) p = 0;

break;

}

泡泡定位

int Judgement(int n4,position pos[12][9],emi b[],int p, int q)///判断是不是碰到泡泡

{

b[q].x1 = b[q].x + 16;

b[q].y1 = b[q].y + 16;

int i,j,z,n,m;

for(i = 11; i>=0; i--)

for(j = 9; j>=0; j--)

{

n = (pos[i][j].x1-b[q].x1);///已知泡泡的横坐标-发射泡泡的横坐标

m = (pos[i][j].y1-b[q].y1);///已知泡泡的纵坐标-发射泡泡的纵坐标

z = n\*n + m\*m;

if(pos[i][j].flag==1 && z<=2048)

{

InitFlag1();

Location(n,m,i,j,n4,pos,b,q);///定位

JudgeMoreThree(pos);///泡泡爆炸

return 1;

}

}

return 0;

}