# 泡泡龙任务3

## 1.1 任务要点

* 泡泡的爆炸功能
* 泡泡的掉落功能

## 1.2 任务内容

* 泡泡的爆炸功能：若发射的泡泡固定后与周围的泡泡有三个以上相同颜色，则泡泡爆炸（递归判断）
* 泡泡的掉落功能：若固定的泡泡没有附着泡泡，则掉落。（用比较繁琐的算法，看看之后能否改进）

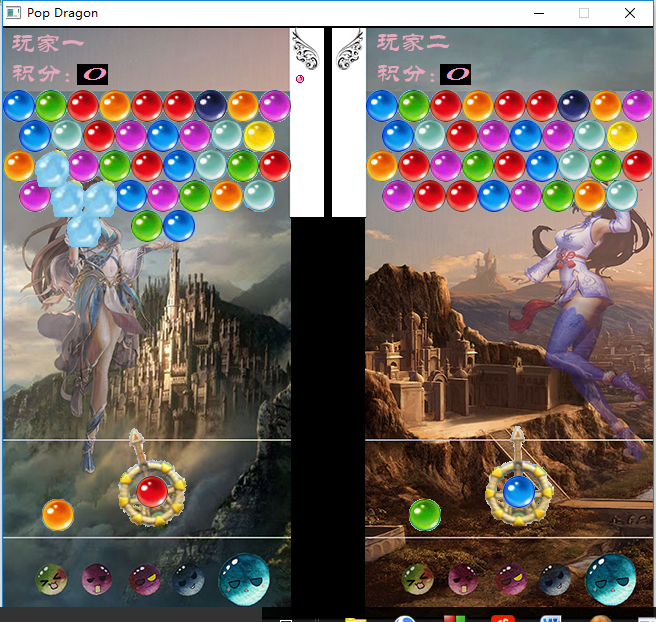
## 1.3 任务实现

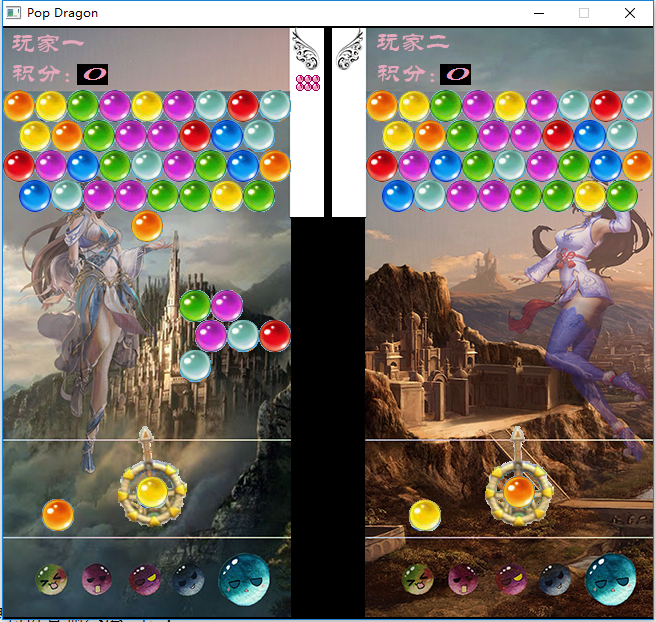
**流程图：**

泡泡发射定位后，判断周围的泡泡是否颜色相同，若相同，则递归，若相同颜色的大于3个，则这些位置泡泡爆炸，产生泡泡动画。

若固定的泡泡没有附着点，则掉落。

效果图：





**核心代码：**

void eliminate(int i, int j,emi b[],int q,position pos[12][9])///判断发射的泡泡周围是否有相同颜色的泡泡

{

emil[z][0] = i;

emil[z++][1] = j;

pos[i][j].flag1 = 1;

if(i%2==0)///奇数排

{

if(i>0 && j>0 && pos[i-1][j-1].flag==1 &&pos[i-1][j-1].flag1==0)///左上

if(pos[i-1][j-1].bubble==b[q].bubble)

{

eliminate(i-1,j-1,b,q,pos);

}

if(i>0 && j<8 && pos[i-1][j].flag==1 &&pos[i-1][j].flag1==0)///右上

if(pos[i-1][j].bubble==b[q].bubble)

{

eliminate(i-1,j,b,q,pos);

}

if(j>0 && pos[i][j-1].flag==1 &&pos[i][j-1].flag1==0)///左边

if(pos[i][j-1].bubble==b[q].bubble)

{

eliminate(i,j-1,b,q,pos);

}

if(j<8 && pos[i][j+1].flag==1 &&pos[i][j+1].flag1==0)///右边

if(pos[i][j+1].bubble==b[q].bubble)

{

eliminate(i,j+1,b,q,pos);

}

if(i<12 && j>0 && pos[i+1][j-1].flag==1 &&pos[i+1][j-1].flag1==0)///左下

if(pos[i+1][j-1].bubble==b[q].bubble)

{

eliminate(i+1,j-1,b,q,pos);

}

if(i<12 && j<8 && pos[i+1][j].flag==1 &&pos[i+1][j].flag1==0)///右下

if(pos[i+1][j].bubble==b[q].bubble)

{

eliminate(i+1,j,b,q,pos);

}

}

else

{

if(pos[i-1][j].flag==1 &&pos[i-1][j].flag1==0)///左上

if(pos[i-1][j].bubble==b[q].bubble)

{

eliminate(i-1,j,b,q,pos);

}

if(pos[i-1][j+1].flag==1 &&pos[i-1][j+1].flag1==0)///右上

if(pos[i-1][j+1].bubble==b[q].bubble)

{

eliminate(i-1,j+1,b,q,pos);

}

if(j>0 && pos[i][j-1].flag==1 &&pos[i][j-1].flag1==0)///左边

if(pos[i][j-1].bubble==b[q].bubble)

{

eliminate(i,j-1,b,q,pos);

}

if(j<7 && pos[i][j+1].flag==1 &&pos[i][j+1].flag1==0)///右边

if(pos[i][j+1].bubble==b[q].bubble)

{

eliminate(i,j+1,b,q,pos);

}

if(pos[i+1][j].flag==1 &&pos[i+1][j].flag1==0)///左下

if(pos[i+1][j].bubble==b[q].bubble)

{

eliminate(i+1,j,b,q,pos);

}

if(pos[i+1][j+1].flag==1 &&pos[i+1][j+1].flag1==0)///右下

if(pos[i+1][j+1].bubble==b[q].bubble)

{

eliminate(i+1,j+1,b,q,pos);

}

}

}

void JudgeMoreThree(position pos[12][9])///是否三个以上，泡泡爆炸

{

int i,j;

if(z<3)

{

z = 0;

InitFlag1();

}

else

{

if(pos[0][0].x==0)

{

amount[1]+=(z-3);

}

else amount[0]+=(z-3);

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

{

pos[emil[i][0]][emil[i][1]].flag = 0;

}

for(i = 0; i<5; i++)

{

Cover\_Back();

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

{

putimage\_transparent(NULL,baozha[i],pos[emil[j][0]][emil[j][1]].x,pos[emil[j][0]][emil[j][1]].y,BLACK);

}

Sleep(20);

}

InitFlag1();

Pop\_drap(pos);

printf("%d %d\n",amount[0],amount[1]);

z = 0;

}

}

int Pop\_drap(position pos[12][9])///泡泡掉落

{

int i,j;

for(i = 0; i<12; i++)

for(j = 0; j<9; j++)

pos[i][j].flag2 = 0;

int l[102][2],t = 0;

for(i = 0; i<12; i++)

{

for(j = 0; j<9; j++)

{

if(i==0 && pos[i][j].flag ==1)

pos[i][j].flag2 = 2;///pos[i][j].flag2=2代表该泡泡不会掉落

if(i!=0 && i%2==0 && pos[i][j].flag==1)

{

if((j!=0 &&pos[i-1][j-1].flag == 1) || pos[i-1][j].flag == 1)

pos[i][j].flag2= 2;

else pos[i][j].flag2 = 1;

}

if(i%2==1 && pos[i][j].flag==1)

{

if(pos[i-1][j].flag==1 || pos[i-1][j+1].flag ==1)

pos[i][j].flag2 = 2;

else pos[i][j].flag2 = 1;

}

}

int z = 0,k,h;

for(j = 0; j<9; j++)

{

if(pos[i][j].flag2==2)

{

if(i!=0 && i%2==0)

{

if(j!=0 && pos[i-1][j-1].flag2==1) pos[i-1][j-1].flag2 = 2;

if(j!=7 && pos[i-1][j].flag2 ==1) pos[i-1][j].flag2 = 2;

}

if(i%2==1)

{

if(pos[i-1][j].flag2==1) pos[i-1][j].flag2 = 2;

if(pos[i-1][j+1].flag2==1) pos[i-1][j+1].flag2 = 2;

}

z = j;

for(k = z-1; k>=0; k--)

{

if(pos[i][k+1].flag2==2 && pos[i][k].flag2 == 1)

pos[i][k].flag2 = 2;

}

for(k = z+1; k<9; k++)

{

if(pos[i][k-1].flag2==2 && pos[i][k].flag2 == 1)

pos[i][k].flag2 = 2;

}

}

}

for(h = 0; h<9; h++)

if(pos[i][h].flag2 == 1)

{

l[t][0] = i\*10+h;

l[t++][1] = pos[i][h].bubble;

pos[i][h].flag = 0;

}

}

if(pos[0][0].x==0)

amount[1]+=t;

else amount[0]+=t;

for(i = 0; i<5; i++)///掉落实现

{

Cover\_Back();

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

{

putimage\_transparent(NULL,a[l[j][1]],pos[l[j][0]/10][l[j][0]%10].x,pos[l[j][0]/10][l[j][0]%10].y+i\*50,BLACK);

}

Sleep(20);

}

}