# 飞机大战任务5

## 1.1 任务要点

我方飞机的死亡和游戏分数及时间的记录

## 1.2 任务内容

* 实现我方飞机的死亡判断
* 实现我方飞机的死亡时的爆炸
* 以及计算游戏分数和游戏时间

## 1.3 任务实现

**流程图：**

游戏时间记录

游戏分数记录

我方飞机死亡

获得游戏开始时间和游戏每一次打印时间

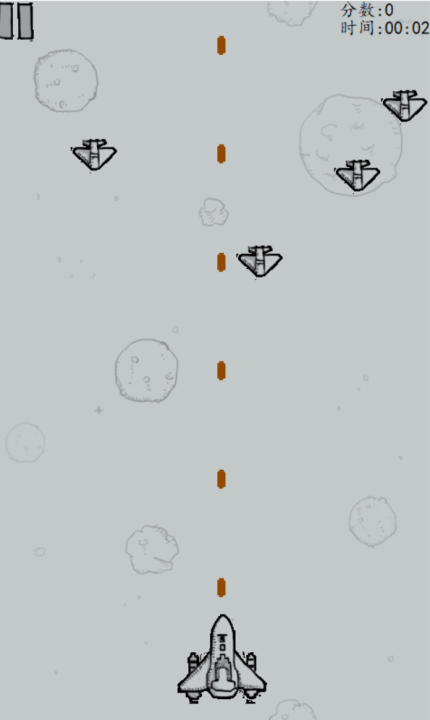
子弹打死敌方飞机就分数增加

当敌方飞机与我方飞机有重叠部分即发生碰撞

用两者的时间差来表示游戏时间

我方飞机发生碰撞后死亡，打印爆炸画面

**实现效果：**



**核心代码：**

void heroDeath(Hero \*one,Enemyhuge \*enemy\_big,Enemymudium \*enemy\_middle,Enemysmall \*enemy\_little)//我方飞机死亡判断

{

for(int i=0;i<4;i++)

{

if((enemy\_little[i].x)>(one->x)&&(enemy\_little[i].x)<(one->x+100)&&(enemy\_little[i].y)>=(one->y)&&(enemy\_little[i].y<one->y+100))

{

one->life=0;

break;

}

else if(enemy\_middle[i].x>one->x&&enemy\_middle[i].x<one->x+100&&enemy\_middle[i].y>=one->y&&enemy\_middle[i].y<one->y+100&&i<2)

{

one->life=0;

break;

}

else if(enemy\_big[0].x>one->x&&enemy\_big[0].x<one->x+100&&enemy\_big[0].y>=one->y&&enemy\_big[0].y<one->y+100&&i==0)

{

one->life=0;

break;

}

}

if(one->life==0)//简单打印爆炸画面

{

ege::putimage\_transparent(NULL,hero\_death[0],one->x,one->y,0);

Sleep(60);

ege::putimage\_transparent(NULL,hero\_death[1],one->x,one->y,0);

Sleep(60);

ege::putimage\_transparent(NULL,hero\_death[2],one->x,one->y,0);

Sleep(60);

ege::putimage\_transparent(NULL,hero\_death[3],one->x,one->y,0);

Sleep(600);

mus[1].Close();

initgraph(320,568);

playinterface();

}

}

for(int i=0; i<20; i++)//游戏分数计算和敌方死亡判断

{

if(bullet\_our[i].x>enemy\_little[0].x&&bullet\_our[i].x<enemy\_little[0].x+51&&bullet\_our[i].y<=enemy\_little[0].y+39&&bullet\_our[i].y>=enemy\_little[0].y&&enemy\_little[0].life==1&&bullet\_our[i].life==1)

{

enemy\_little[0].life=0;

one.socre+=1;

bullet\_our[i].life=0;

}

else if(bullet\_our[i].x>enemy\_little[1].x&&bullet\_our[i].x<enemy\_little[1].x+51&&bullet\_our[i].y<=enemy\_little[1].y+39&&bullet\_our[i].y>=enemy\_little[1].y&&enemy\_little[1].life==1&&bullet\_our[i].life==1)

{

enemy\_little[1].life=0;

one.socre+=1;

bullet\_our[i].life=0;

}

else if(bullet\_our[i].x>enemy\_little[2].x&&bullet\_our[i].x<enemy\_little[2].x+51&&bullet\_our[i].y<=enemy\_little[2].y+39&&bullet\_our[i].y>=enemy\_little[2].y&&enemy\_little[2].life==1&&bullet\_our[i].life==1)

{

enemy\_little[2].life=0;

one.socre+=1;

bullet\_our[i].life=0;

}

else if(bullet\_our[i].x>enemy\_little[3].x&&bullet\_our[i].x<enemy\_little[3].x+51&&bullet\_our[i].y<=enemy\_little[3].y+39&&bullet\_our[i].y>=enemy\_little[3].y&&enemy\_little[3].life==1&&bullet\_our[i].life==1)

{

enemy\_little[3].life=0;

one.socre+=1;

bullet\_our[i].life=0;

}

else if(bullet\_our[i].x>enemy\_middle[0].x&&bullet\_our[i].x<enemy\_middle[0].x+69&&bullet\_our[i].y<=enemy\_middle[0].y+89&&bullet\_our[i].y>=enemy\_middle[0].y&&enemy\_middle[0].life>=1&&bullet\_our[i].life==1)

{

enemy\_middle[0].life--;

if(enemy\_middle[0].life==0)

one.socre+=3;

bullet\_our[i].life=0;

}

else if(bullet\_our[i].x>enemy\_middle[1].x&&bullet\_our[i].x<enemy\_middle[1].x+69&&bullet\_our[i].y<=enemy\_middle[1].y+89&&bullet\_our[i].y>=enemy\_middle[1].y&&enemy\_middle[1].life>=1&&bullet\_our[i].life==1)

{

enemy\_middle[1].life--;

if(enemy\_middle[1].life==0)

one.socre+=3;

bullet\_our[i].life=0;

}

else if(bullet\_our[i].x>enemy\_big[0].x&&bullet\_our[i].x<enemy\_big[0].x+165&&bullet\_our[i].y<=enemy\_big[0].y+246&&bullet\_our[i].y>=enemy\_big[0].y&&enemy\_big[0].life>=1&&bullet\_our[i].life==1)

{

enemy\_big[0].life--;

if(enemy\_big[0].life==0)

one.socre+=5;

bullet\_our[i].life=0;

}

}

//获取游戏时间

endtime=fclock();

ege::setbkmode(TRANSPARENT);

ege::setcolor(BLACK);

ege::setfont(20,0,"楷体");

sprintf(scoredisplay,"分数:%d",one.socre);

ege::outtextxy(380,0,scoredisplay);

difftimesec=(int) endtime-starttime;

if(difftimesec>=60)

{

difftimemin=difftimesec/60;

difftimesec=difftimesec%60;

}

ege::setbkmode(TRANSPARENT);

ege::setcolor(BLACK);

ege::setfont(20,0,"楷体");

sprintf(timedisplay,"时间:%02d:%02d",difftimemin,difftimesec);

ege::outtextxy(380,20,timedisplay);