# 飞机大战任务7

## 1.1 任务要点

实现游戏历史最高的储存和排序

## 1.2 任务内容

* 在游戏结束后储存分数
* 并将分数进行排名，只存最高的10个
* 游戏储存的函数void historyHishest(Hero \*one,int gametimemin,int gametimesec)

## 1.3 任务实现

当游戏结束后将分数和时间传给文件

**流程图：**

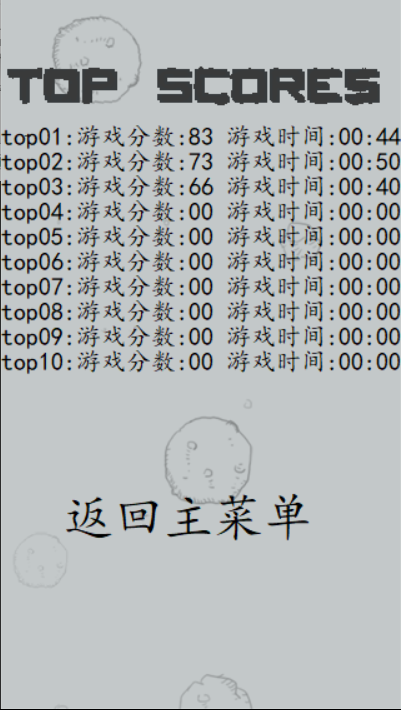
在将文件的数据和传来的结束比较

如果比文件的分数大就储存，反之则否

在主菜单的历史最高中可以查看10个最高分

在历史最高下可以返回主菜单

**实现效果：**



**核心代码：**

void historyHishest(Hero \*one,int gametimemin,int gametimesec)//游戏历史最高记录

{

FILE \*fp;

Historyhishest recordgame[11];

fp=fopen("bin//Debug//historytop.txt","rb");

if(fp==NULL)

printf("not open");

for(int i=0; i<10; i++)

fread(&recordgame[i],sizeof(Historyhishest),1,fp);

fclose(fp);

fp=fopen("bin//Debug//historytop.txt","wb");

if(fp==NULL)

printf("not open");

for(int i=0; i<10; i++)//储存游戏分数和时间

{

if(one->socre>recordgame[i].socre&&i==0)

{

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

{

recordgame[j+1]=recordgame[j];

}

recordgame[0].socre=one->socre;

recordgame[0].gametimemin=gametimemin;

recordgame[0].gametimesec=gametimesec;

break;

}

if(one->socre<recordgame[i].socre&&one->socre>=recordgame[i-1].socre&&i>0)//对游戏分数排序

{

for(int j=9; j>=i-1; j--)

{

recordgame[j+1]=recordgame[j];

}

recordgame[i-1].socre=one->socre;

recordgame[i-1].gametimemin=gametimemin;

recordgame[i-1].gametimesec=gametimesec;

}

}

for(int i=0; i<10; i++)

{

fwrite(&recordgame[i],sizeof(Historyhishest),1,fp);

}

fclose(fp);

}

void historyReading()//历史最高

{

FILE \*fp;

Historyhishest record[11]= {0};

char ranking[11][50]= {0};

char text6[12]="返回主菜单";

fp=fopen("bin//Debug//historytop.txt","rb");

if(fp==NULL)

printf("not open");

for(int i=0; i<11; i++)

{

fread(&record[i],sizeof(Historyhishest),1,fp);

}

fclose(fp);

fp=fopen("bin//Debug//historytop.txt","wb");

if(fp==NULL)

printf("not open");

for(int i=0; i<11; i++)

{

fwrite(&record[i],sizeof(Historyhishest),1,fp);

}

fclose(fp);

initmouse();

for(; ege::is\_run(); delay\_fps(60))

{

ege::cleardevice();

ege::putimage(0,0,Map4);

ege::putimage\_transparent(NULL,history\_top,0,30,0);

//ege::putimage\_transparent(NULL,)

ege::setbkmode(TRANSPARENT);

updatemouse();

for(int i=0; i<10; i++)//打印游戏历史分数

sprintf(ranking[i],"top%02d:游戏分数:%02d 游戏时间:%02d:%02d",i+1,record[i].socre,record[i].gametimemin,record[i].gametimesec);

for(int i=0; i<10; i++)

{

ege::setcolor(BLACK);

ege::setfont(20,0,"楷体");

ege::outtextxy(0,100+20\*i,ranking[i]);

}

ege::setcolor(BLACK);

ege::setfont(40,0,"楷体");

ege::outtextxy(50,395,text6);

if(Mouse\_X>=50&&Mouse\_Y>=395&&Mouse\_X<=250&&Mouse\_Y<=435)

{

ege::setcolor(BLUE);

ege::setfont(40,0,"楷体");

ege::outtextxy(50,395,text6);

if(Mouse\_Left==true && Mouse\_Up==true)//返回主菜单

{

Mouse\_Left=false;

Mouse\_Up=false;

return;

}

}

}

}