

# 高级语言程序设计

## 课设报告

题 目\_\_\_\_\_2\_\_\_\_\_

学 号\_\_\_\_\_21062122\_\_\_\_\_

姓 名\_\_\_\_\_林家棋\_\_\_\_\_

指导教师\_\_\_\_\_周琨\_\_\_\_\_

提交日期\_\_\_\_\_

成绩评价表

| 课设报告内容  | 课设报告结构   | 课设报告图表   | 界面  | 最终成绩 |
|---|--|--|---|------|
| <input type="checkbox"/> 丰富正确<br><input type="checkbox"/> 基本正确<br><input type="checkbox"/> 设计部分少<br><input type="checkbox"/> 过于简单         | <input type="checkbox"/> 完全符合要求<br><input type="checkbox"/> 基本符合要求<br><input type="checkbox"/> 有欠缺                               | <input type="checkbox"/> 符合规范<br><input type="checkbox"/> 基本符合规范<br><input type="checkbox"/> 有一些错误                                     | <input type="checkbox"/> 丰富<br><input type="checkbox"/> 有背景图片<br><input type="checkbox"/> 只有背景色<br><input type="checkbox"/> 按钮效果好 |      |
| 程序功能实现  | 程序执行情况   | 问题回答情况   | 总体评价  |      |
| <input type="checkbox"/> 多个扩展功能<br><input type="checkbox"/> 少量扩展功能<br><input type="checkbox"/> 完成基本功能<br><input type="checkbox"/> 未完成基本功能 | <input type="checkbox"/> 流畅<br><input type="checkbox"/> 界面有闪动<br><input type="checkbox"/> 操作不灵活<br><input type="checkbox"/> 有小错误 | <input type="checkbox"/> 清楚、正确<br><input type="checkbox"/> 基本正确<br><input type="checkbox"/> 回答有部分错误<br><input type="checkbox"/> 不能回答问题 |   |      |

教师签字:\_\_\_\_\_

# 目录

|             |   |
|-------------|---|
| 1. 需求分析     | 4 |
| 1.1. 功能需求   | 4 |
| 1.1.1. 基本功能 | 4 |
| 1.1.2. 扩展功能 | 4 |
| 1.2. 界面需求   | 5 |
| 1.3. 开发环境   | 5 |
| 2. 总体设计     | 5 |
| 2.1. 主要数据结构 | 5 |
| 2.2. 程序总体结构 | 7 |
| 2.3. 子模块设计  | 7 |
| 3. 详细设计     | 9 |
| 3.1. main() | 9 |

|                  |    |
|------------------|----|
| 3.2. game()      | 10 |
| 4. 测试            | 12 |
| 5. 开发中遇到的问题及解决方法 | 14 |
| 6. 总结提高          | 15 |
| 6.1. 课程设计总结      | 15 |
| 7. 附件：程序源代码      | 15 |

# 1. 需求分析

## 1.1. 功能需求

### 1.1.1 基本功能

1. 程序应记录用户名和分数并显示在显示区中。
2. 设置“开始”“暂停”按钮实现游戏进行控制和按钮互相变换。设置“结束”按钮结束游戏，清空游戏区，按“开始”重新游戏。设置“增加”按钮可增加飞机。设置“退出”按钮结束程序并将用户名分数写入文件。
3. 飞机在不低的地方从左往右匀速飞行循环，开始时不少于三架飞机依次随机高度飞出，炸掉后消失。
4. 炮为低处中心，左右键变化方向，空格键射出炮弹。
5. 打出炮弹即扣一分，分数为 0 时不扣分，打到飞机时爆炸加 3 分
6. 动态显示得分，“结束”后得分清零。
7. “退出”后，将用户名和本盘游戏得分存入文件

### 1.1.2 扩展功能

1. 光标移动到按键时会产生蓝色方框移开则消失，添加按钮音效。
2. 显示游戏时间，暂停时时间不计。
3. 增加音乐开关按钮。
4. 实现存盘（将当前游戏的状态存入文件）和导入（将文件中存储的游戏状态恢复到屏幕上，可以继续上次的游戏）。
5. 添加飞机被打爆时爆炸动画和爆炸音效。
6. 排行榜、记录刷新提醒、特殊游戏音效。
7. 两种不同目标飞机，可连发弹药。

## 1.2. 界面需求



## 7.1. 开发环境

Microsoft Visual Studio 2010; EasyX

## 2. 总体设计

### 2.1. 主要数据结构

结构体类型:

1. 炮弹 fire, 包含炮弹是否发出、是否炸到飞机、发射角度, 以及随时横纵坐标
2. 飞机 planes, 包含飞机是否爆炸、飞行高度、初始出现时间、爆炸信息, 以及由于以链表形式存储故有 planes 类型指针 next 指向下一个节点

3. 玩家 Player，包含用户名、分数、飞机数等

文件存储：

1. “游戏存档.txt”用以存储未结束的游戏信息，包含玩家用户名、获得的分数、飞机总数的玩家信息；以及现存飞机，游戏时间的全盘信息；和每架飞机是否爆炸、高度、目前横坐标、飞机种类等飞机信息。

```
1
5 kjaaa 5 7 4
6
91 624 0 1 0
124 324 0 1 0
125 493 0 0 0
79 387 0 0 1
163 487 0 0 1
179 43 0 1 0
171 619 0 0 1
```

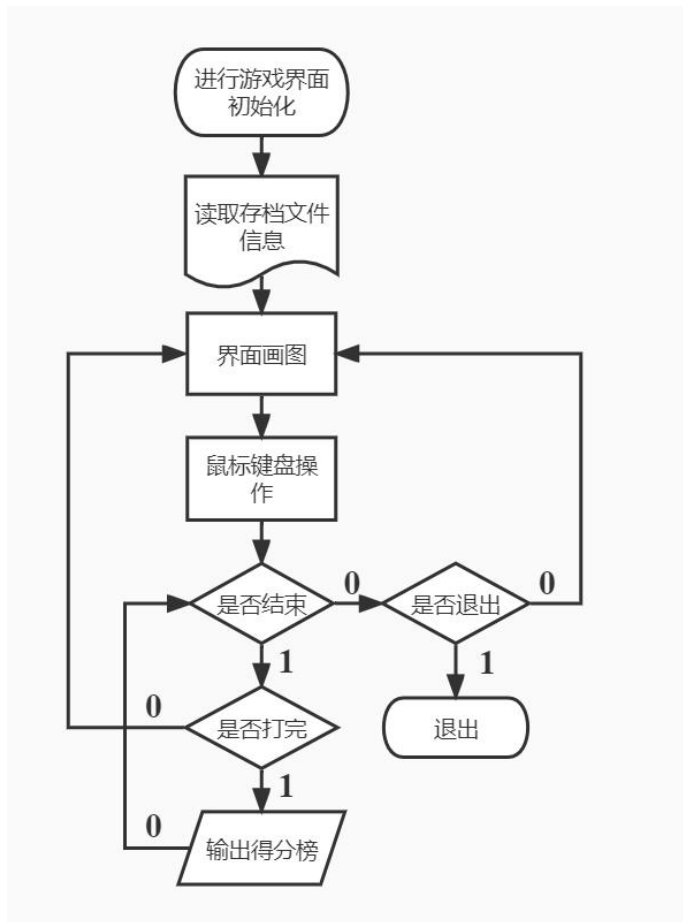
存储格式：

2. “打飞机得分榜.txt”用以存储历史完成游戏玩家信息，包含玩家用户名、分数、排名等信息。

```
5
3 abc 54 58
6 Liu HQ 37 40
5 skjdk 10 17
5 skjdk 9 15
4 abcd 8 14
```

存储格式：

## 2.2. 程序总体结构



## 2.3. 子模块设计

读出文件的函数模块：void ReadGrade()

功能：读出历史玩家分数排名

我利用文件读出历史玩家的相关信息。

写入文件的函数模块：void WriteGrade()

功能：写入前五玩家分数排名

然后通过其得分从高到低进行排名，其中得分相同取飞机数较少者，只取前五位。

写入文件的函数模块：void Store\_file(ii)

功能：写入目前未结束游戏的信息

参数：保存状态瞬间 for 循环内循环数 i，long long 型

当点击“保存状态”后，程序将调用该函数，写入飞机的相关信息、玩家用户名、游戏时间等等。

读出文件的函数模块：void Read\_file()

功能：读出上次保存的游戏的信息

新进入游戏时若有存档会提示是否读档，或在游戏中若之前有存档，则可点击读档按键直接读取存档信息继续游戏。

写入文件的函数模块：Clear\_file()

功能：没有需要保存的游戏状态时，重置存储文件内存储信息

当没有“保存状态”且结束游戏时，调用该函数对存档文件进行清空。以避免错误地以为有存档而读出错的存档信息。

删除整条链表的函数模块：void Delet(planes \*head)

功能：删除存储飞机信息的链表

参数：链表的头指针，结构体 planes 型

当结束游戏时，为了进行下一盘游戏以及清理空间，我利用该函数从头指针 head 逐个取 next 并 free 释放空间，该函数在模块 init()内调用。

对游戏状态初始化的函数模块：void init()

功能：结束游戏时，将游戏状态重置以便重新开始游戏

结束游戏时，对游戏所有状态进行初始化，以便于重新开始新的游戏。

初始化界面的函数模块：void Interface()

功能：建立游戏界面和相关美化等

建立游戏界面，同时在 game()和其他子模块中当需要对输出进行更新时，让程序先调用 Interface()，对界面内容进行初始化，再进行需要的内容输出。

鼠标信息处理的函数模块：void mouses()

功能：游戏开始前仅一次的鼠标信息处理操作

新玩家信息读入的函数模块：void Input()

功能：读入玩家用户名、初始飞机数量

链表中飞机插入的函数模块：planes \*AppendNode(planes \*head)

功能：在存储飞机的链表中尾插入新的飞机节点

参数：飞机链表头指针，结构体 planes 型

返回值：飞机链表头指针

每调用一次时在链表尾部插入一个新节点，返回链表头指针。由于在程序中，我先



在外面建好头指针，故在尾插时不需特判当前是否不存在头指针的问题，只需将当前指针通过 `next` 遍历到尾部在进行新节点插入操作即可。

退出游戏的函数模块：`Exit()`

功能：游戏中进行退出游戏操作

在点击“退出”按键时，调用该函数，提示确认退出游戏或继续游戏，若退出即退出整个程序，其时间算在暂停时间里不会记录到游戏时间。

更新分数板的函数模块：`Updatescore()`

功能：打到飞机或打出子弹后更新分数区域

更新分数，其中输出前将调用 `Interface()` 模块进行界面的初始化，再进行相关内容的输出。

输出得分榜的函数模块：`void print_board()`

功能：游戏结束时输出历史玩家前五得分榜

载入图片和音乐的函数模块：`void Load_picture_music()`

功能：用来载入游戏相关的图片和音乐和音效

读档的选择函数模块：`void If_continue()`

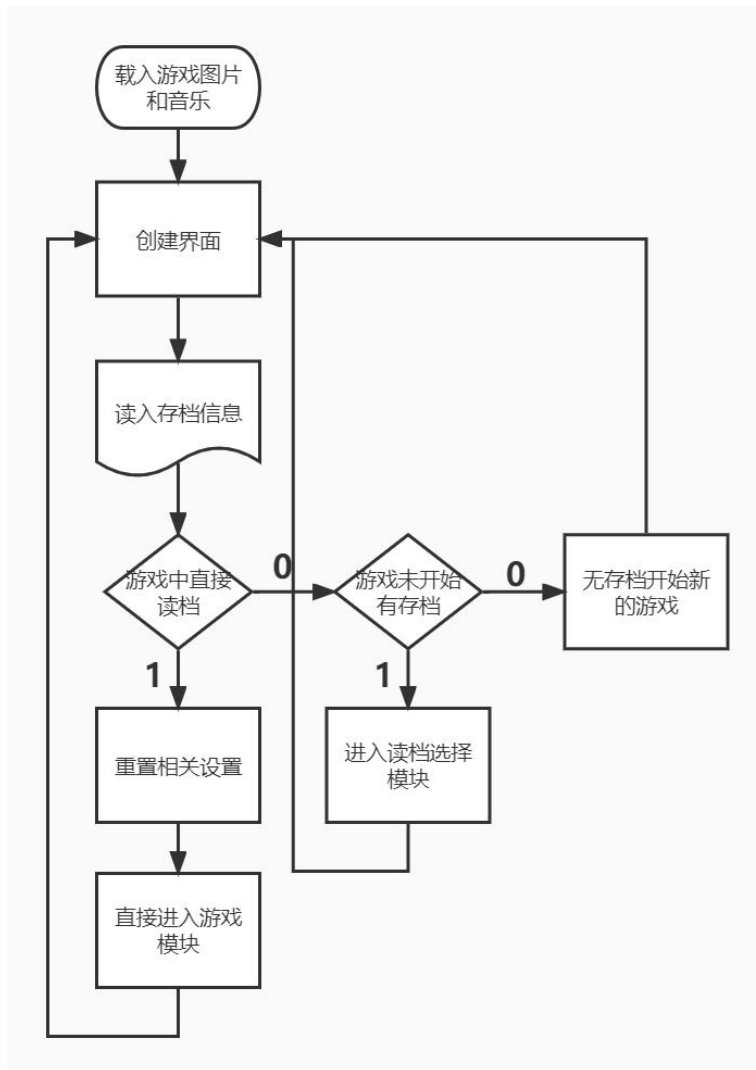
功能：刚进入游戏时若有上局存档，先询问是否读档再进入游戏

## 3. 详细设计

### 3.1 `main()`

```
int main()
```

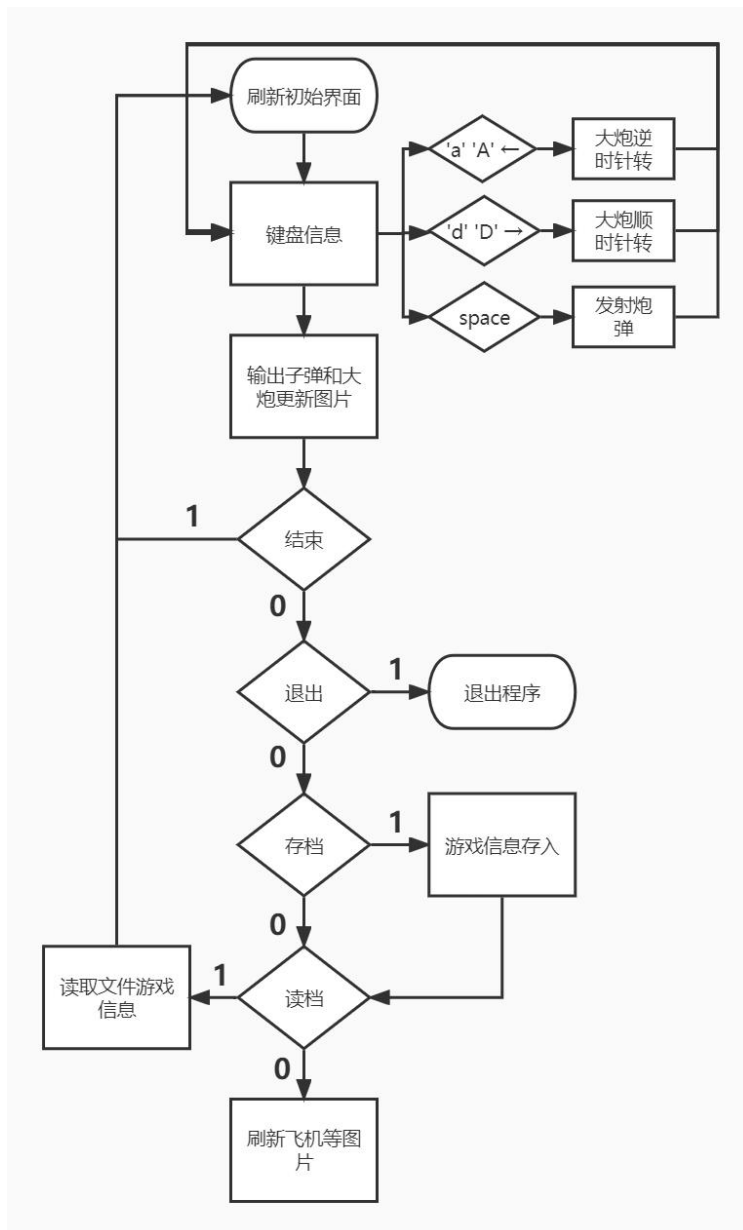
`main` 函数只用于一次的进入接口和彻底重建界面，几乎无任何实际作用实现。



### 3.2 game()

void game()

最核心的游戏操作和输出刷新模块，使用到多个分模块函数。



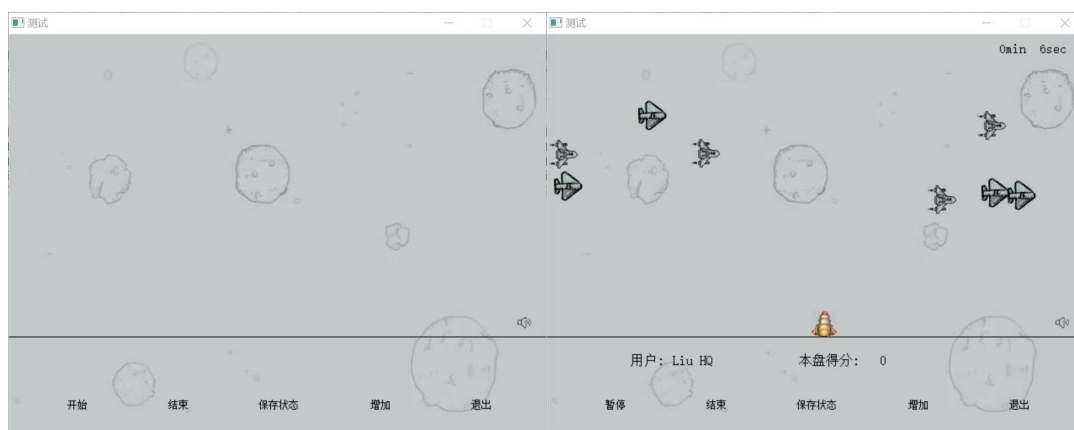
- 发射炮弹时调用 `mcSendString` 放出音乐；遍历结构体数组所有炮弹，遍历到空当炮弹则直接存储信息，若都正在飞行，则增加一个炮弹数量存储信息。
- 大炮转动，按左右键时，`angle` 变量记录大炮角度，每次更新大炮图片时，重新 `loadimage`，防止图片转多次后变黑，然后对图片就行 `rotateimage` 相应的 `angle`，并利用掩码图原理处理，否则会出现转动图片时有方形黑底的情况。
- 炮弹射出后，用 `x,y` 记录每个循环内炮弹的位置，`x,y` 用角度累加速度与 `cos/sin` 乘积累加更新。
- 使用按键时，鼠标移动到区域内出现蓝色方框，移出消失，需要利用 `getimage` 和

putimage, 在初始界面更新时, 需要更新 getimage, 否则务必避免 getimage; 按下按键, 调用 mciSendString 放出按键相关音乐。

- 开启/关闭背景音乐时, 点击输出相应不同音乐图片, 并调用 mciSendString 中的 pause 或 resume bkmusic。
- 显示游戏时间, 定义一个 pause\_time 为进入 pause 和 Exit() 的非游戏时间, game\_time 表达式  $game\_time = (int)((End\_time - Start\_time) / CLOCKS\_PER\_SEC) - pause\_time$  为进入 game() 模块现时间与初始时间差值再与 pause\_time 的差值, 存读档时写出 game\_time 读入-pause\_time, 界面输出分 minutes 和 seconds。
- 贴飞机和炮弹时, 遍历飞机链表和炮弹数组, 如果正在飞行则输出, 若飞机正在爆炸时间内, 则输出相应爆炸阶段的图片 (共 4 个爆炸阶段), 超出时间则不输出飞机和炮弹。

## 4. 测试

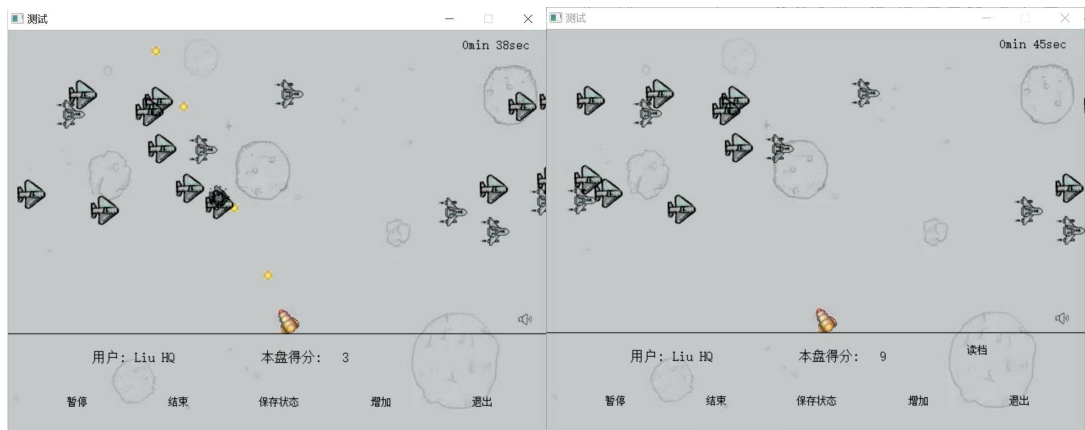
- A. 进入游戏后, 开启游戏音乐, 只能使用“开始”和“退出”按键功能; 点击“开始”, 出现按键音效; 输入用户名和初始飞机数; 开始游戏, 出现两种不同样式飞机, “开始”键自动变成暂停; 此时五个键都能使用。



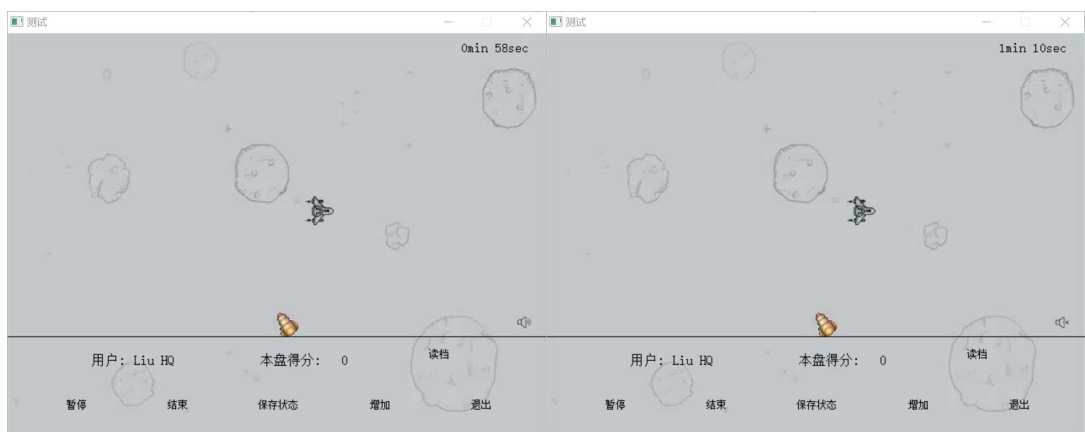
- B. 点击“暂停”, 飞机、时间、大炮等消失, 出现大的“GAME PAUSE”字样, “暂停”键变成“开始”; 此时“开始”“结束”“保存状态”“退出”按键能使用。
- C. 点击“开始”, 继续“暂停”前的游戏画面。
- D. 点击“结束”, 全部恢复初始状态, 见图 1。
- E. 重复 A; 任意按“A”“a”“←”键大炮逆时针转动, 按“D”“d”“→”键大炮顺时针转动, 可长按持续转动, 连续按“SPACE”空格键可连续发射炮弹, 有炮弹发射音效。



- F. 打到飞机，有飞机爆炸连贯动画，飞机爆炸音效，显示区获得加分
- G. 点击“保存状态”，显示区右上角出现“读档”。
- H. 将飞机打完后，点击“读档”，马上恢复“保存状态”时游戏局面。

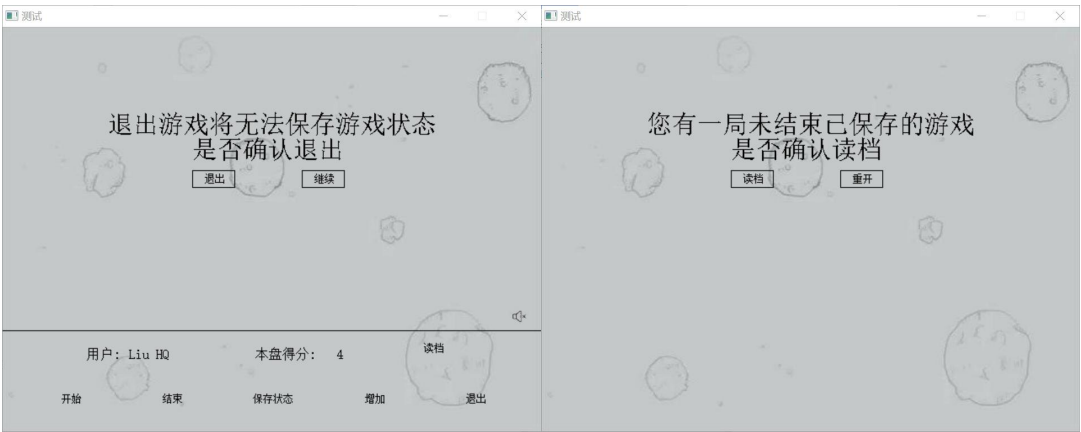


- I. 继续将飞机打完，点击“增加”，发现界面马上随机位置出现一架飞机。
- J. 点击游戏区右下角“音乐”图标，马上变成“音乐关闭”图标，并关闭游戏音乐；再次点击，又变成“音乐”图标，并继续之前关闭的游戏音乐。



- K. 点击“退出”后，出现提示字样，和“退出”“继续”两种选择。

- L. 点击“继续”后返回游戏图 8；重复 J；点击“退出”，直接终止程序。
- M. 启动游戏，由于上局有保存的局，游戏出现提醒字样，和“读档”“重开”两种选择。



- N. 点击“读档”，游戏回到（上局保存状态时局面）图 6；点击“结束”，存档文件清空，游戏局面回到图 1；重复 A，G，K，L，M，点击“重开”，游戏局面回到图 1，存档文件清空。
- O. 重复 A，将飞机打完，点击“结束”，出现“打飞机得分榜”，并出现榜单刷新音效；此时可以使用“开始”“退出”。
- P. 重复 K，L，整个测试结束。



5. 开发中遇到的问题及解决办法

- 1. 在飞机循环飞行时切换背景屏幕会出现闪频问题，经过搜索我找到了一个解决方法使用 BeginBatchDraw(), FlushBatchDraw(), EndBatchDraw() 三个函数用于批量绘图。执行 BeginBatchDraw() 之后暂停绘图，直到 FlushBatchDraw() 或 EndBatchDraw() 绘制出所有未完成的任务。这样便很简单地解决了闪频的问题。
- 2. 在贴图时，尤其是转动大炮时会留有黑色的底，经过了解我知道了利用掩码图这一

奇妙的方法,就是准备一张黑底原图和白底黑图,并在贴图时分别加上 SRCAND 和 SRCINVERT,便可以完美贴出需要的原图了。

## 6. 总结提高

### 6.1. 课程设计总结

这次 C 语言程序设计课设计制作过程中,我学到了很多,首先最显然的便是学习如何利用 EasyX 插件来进行游戏所需图像插入等特殊需求。但更大的收获是,由于小时学过 OI,较看重算法数据结构等,眼高手低,对大代码量低技术量的工作嗤之以鼻,在此次六百多行代码的实践中,我遇到了一些问题,关于处理图像和死循环中正确更新内容的小 bug 修复,以及外观操作等无止境的优化。

我认为我的完成是较为良好的,开始时我不知道从何做起,但也一点一点实现,构建起整个框架,不断测试,逐步完善子功能。再到后来我基本完成了所有基础+预设想扩展功能才发现同学几乎开始,但课设便是与自己比较,不断优化,不断完善。且目前个人工程能力尚浅,距离独立做出一个精美耐玩的游戏还很远,仍需不断实践不断学习。

## 7. 附件：程序源代码

```
#include<stdio.h>
#include<string.h>
#include<graphics.h>
#include<conio.h>
#include<time.h>
#include<stdlib.h>
#include<math.h>
#include<Windows.h>
#pragma comment(lib,"Winmm.lib")
#define mod 640
#define plane_speed 1
#define fire_speed 3
#define PI 3.14159265
#define DU PI/180 //1°
#define ll long long
#define PUTMUSIC mciSendString("close put_music",NULL,0,NULL);mciSendString("open
LOAD/MUSIC/按键.mp3 alias put_music",NULL,0,NULL);mciSendString("play
```

```

put_music", NULL, 0, NULL);
#define CITE int &game_time, int &pause_time, int &player_number, int &start_or_pause, int
&ifplayer, int &fire_number, int &if_file, int &if_store, int &restore, int
&if_music_open, double &angle
#define VALUE
game_time, pause_time, player_number, start_or_pause, ifplayer, fire_number, if_file, if_store,
restore, if_music_open, angle
IMAGE
img_bk, img_bullet, img_Plane1, img_Plane2, img_planeExplode2[4], img_planeExplode1[4], img_f
ire, music, shutdown_music, img_cannon1, img_cannon2;
// 背景图片, 子弹图片, 敌机图片, 爆炸飞机图片, 子弹图片, 音乐键, 音乐关闭键, 大炮 1, 2
clock_t Start_time, End_time; //游戏时间起始
struct fire{//炮弹
    int x, y, present;
    double b_angle;
    bool if_flying;
};
struct planes{//链表存储飞机
    int height, start_time, if_firsttime, if_damaged, plane_kind, explode_x, explode_y;
    ll explode_time;
    planes *next;
};
struct Player{//玩家
    char name[20];
    int score, planes, planes_present, name_len;
    bool if_printed;
}player[100];
planes *HEAD;
inline int pow(int x){return x*x;}
planes *AppendNode(planes *head) {//尾插链表新飞机节点
    planes *p=NULL, *pr=head;
    p=(planes *)malloc(sizeof(planes));
    if(head==NULL) head=p;
    else{
        for(;pr->next!=NULL;pr=pr->next){}
        pr->next=p;
    }
    p->height=rand()%181+50;
    p->start_time=rand()%501;
    p->if_firsttime=1;

```



```

    p->next=NULL;
    p->if_damaged=0;
    p->plane_kind=rand()%2;
    return head;
}

void ReadGrade(CITE) { //从文件读取得分信息
    FILE* pf=fopen("LOAD/FILE/打飞机得分榜.txt", "r");
    fscanf(pf, "%d\n", &player_number); player_number++;
    for(int i=2; i<=player_number; i++) {
        fscanf(pf, "%d ", &player[i].name_len);
        for(int j=0; j<player[i].name_len; j++) fscanf(pf, "%c", &player[i].name[j]);
        fscanf(pf, " %d %d\n", &player[i].planes, &player[i].score);
    }
    fclose(pf);
    pf=NULL;
}

void WriteGrade(CITE) { //将得分信息写入文件
    FILE* pf=fopen("LOAD/FILE/打飞机得分榜.txt", "w");
    int if_print[10];
    int player_num=player_number<5?player_number:5;
    memset(if_print, 0, sizeof(if_print));
    fprintf(pf, "%d\n", player_num);
    settextstyle(25, 0, "微软黑体");
    outtextxy(100, 50, "打飞机得分榜");
    settextstyle(18, 0, "微软黑体");
    outtextxy(100, 90, "排名  玩家      飞机数  得分");
    for(int i=1; i<=player_num; i++) {
        int p=0, max_score=0, min_plane=10000;
        for(int j=1; j<=player_number; j++)

            if(!if_print[j]&&((max_score<player[j].score) || (max_score==player[j].score&&min_plane>player[j].planes)))
                p=j, max_score=player[p].score, min_plane=player[p].planes;
        char buf[100];
        sprintf(buf, "NO.%d  %s", i, player[p].name);
        outtextxy(100, 90+i*30, buf);
        for(int j=1; j<=10-player[p].name_len; j++) outtextxy(120+j*13, 90+i*30, " ");
        sprintf(buf, " %2d    %2d", player[p].planes, player[p].score);
        outtextxy(250, 90+i*30, buf);
        fprintf(pf, "%d ", player[p].name_len);
    }
}

```

```

        for(int j=0;j<player[p].name_len;j++) fprintf(pf, "%c", player[p].name[j]);
        fprintf(pf, " %d %d\n", min_plane, max_score), if_print[p]=1;
    }
    fclose(pf);
    pf=NULL;
    settextstyle(16, 0, "微软黑体");
}

void Store_file(11 ii, CITE) { //存档
    FILE* pf=fopen("LOAD/FILE/游戏存档.txt", "w");
    fprintf(pf, "1\n%d ", player[1].name_len);
    for(int i=0;i<player[1].name_len;i++) fprintf(pf, "%c", player[1].name[i]);
    fprintf(pf, " %d %d %d\n", player[1].score, player[1].planes, player[1].planes_present);
    fprintf(pf, "%d\n", game_time);
    planes *p=HEAD;
    while(p!=NULL) {

        fprintf(pf, "%d %d %d %d %d\n", p->height, (p->start_time-(int)(ii%mod)+mod)%mod, p->if_f
irsttime, p->if_damaged, p->plane_kind);
        p=p->next;
    }
    if_store=1;
    fclose(pf);
    pf=NULL;
}

void Read_file(CITE) { //读档
    FILE* pf=fopen("LOAD/FILE/游戏存档.txt", "r");
    fscanf(pf, "%d\n", &if_file);
    if(if_file) {
        fscanf(pf, "%d ", &player[1].name_len);
        for(int i=0;i<player[1].name_len;i++) fscanf(pf, "%c", &player[1].name[i]);

        fscanf(pf, " %d %d %d\n", &player[1].score, &player[1].planes, &player[1].planes_present)
;

        fscanf(pf, "%d\n", &pause_time); pause_time*=-1;
        planes *head=NULL;
        head=AppendNode(head);
        HEAD=head;
        for(int i=2;i<=player[1].planes;i++) head=AppendNode(head);
        planes *p=HEAD;
        while(p!=NULL) {

```

```

        fscanf(pf, "%d %d %d %d %d\n", &p->height, &p->start_time, &p->if_firsttime, &p->if_damage
d, &p->plane_kind);
        p->explode_time=-10000;
        p=p->next;
    }
}
fclose(pf);
pf=NULL;
}
void Clear_file() { //空档
    FILE* pf=fopen("LOAD/FILE/游戏存档.txt", "w");
    fprintf(pf, "0\n");
    fclose(pf);
    pf=NULL;
}
void Delet(planes *head) { //删除整个飞机链表
    planes *p=head, *pr=NULL;
    while(p!=NULL) {
        pr=p;
        p=p->next;
        free(pr);
    }
}
void init(CITE) { //结束时的初始化数据
    restore=0, if_store=0, start_or_pause=1, ifplayer=0, player[1].if_printed=0;
    game_time=0, pause_time=0, fire_number=1;
    Delet(HEAD);
    for(int i=1; i<=player_number; i++) {
        strcpy(player[i].name, "");
        player[i].planes=0, player[i].score=0;
    }
    Clear_file();
}
void Interface(CITE) { //初始化界面
    setfillcolor(WHITE);
    fillrectangle(0, 0, 640, 480);
    putimage(0, 0, &img_bk);
    player[1].if_printed=0;
    setcolor(BLACK);

```

```

line(0, 360, 640, 360);
setbkmode(TRANSPARENT);
if_music_open?putimage(600, 330, &music, SRCAND):putimage(600, 330, &shutdown_music, SRCAND);
ND);
if(ifplayer){
    settextstyle(16, 0, "微软黑体");
    outtextxy(100, 380, "用户:");outtextxy(150, 380, player[1].name);
    char buf[100];
    sprintf(buf, "本盘得分: %d", player[1].score);
    outtextxy(300, 380, buf);
}
settextstyle(12, 0, "微软黑体");
setcolor(BLACK);
start_or_pause?outtextxy(70, 435, "开始"):outtextxy(70, 435, "暂停");
if(if_store) outtextxy(500, 375, "读档");
outtextxy(190, 435, "结束");
outtextxy(297, 435, "保存状态");
outtextxy(430, 435, "增加");
outtextxy(550, 435, "退出");
}
void Exit(CITE){//退出确认
    Interface(VALUE);
    IMAGE img3, img4;
    getimage(&img4, 0, 0, 640, 480);
    settextstyle(30, 0, "微软黑体");
    if(ifplayer&&player[1].planes_present)outtextxy(125, 100, "退出游戏将无法保存游戏状态");
    outtextxy(225, 130, "是否确认退出");
    settextstyle(12, 0, "微软黑体");
    setcolor(BLACK);
    rectangle(225, 170, 275, 190);
    rectangle(355, 170, 405, 190);
    outtextxy(240, 174, "退出");
    outtextxy(370, 174, "继续");
    getimage(&img3, 0, 0, 640, 480);
    bool exit_=1;
    MOUSEMSG m;
    while(exit_){
        EndBatchDraw();
        m=GetMouseMsg();
    }
}

```

```

        if(m.x>=225&& m.x<=275&& m.y>=170&& m.y<=190) { //退出
            setlinecolor(BLUE);
            rectangle(220,165,280,195);
            if(m.uMsg==WM_LBUTTONDOWN) {
                if(!if_store) Clear_file();
                exit(0);
            }
        }
        else if(m.x>=355&& m.x<=405&& m.y>=170&& m.y<=190) { //继续
            setlinecolor(BLUE);
            rectangle(350,165,410,195);
            if(m.uMsg==WM_LBUTTONDOWN) {
                Interface(VALUE);
                putimage(0,0,&img4);
                exit_=0;
            }
        }
        else putimage(0,0,&img3);
    }
}

void input() { //输入用户开局信息
    player[1].score=0;
    outtextxy(100,200,"请输入用户名: ");
    char c;int len=-1;
    while((c=_getch())!= '\r') {
        player[1].name[++len]=c;
        outtextxy(180+len*8,200,c);
    }
    player[1].name_len=len+1;
    outtextxy(100,240,"请输入飞机数: ");
    char c1;len=-1;int ini_pla=0;
    while((c1=_getch())!= '\r') {
        ++len;
        ini_pla+=(c1-'0')*(!len?1:10);
        outtextxy(180+len*8,240,c1);
    }
    player[1].planes=ini_pla;
    planes *head=NULL;
    head=AppendNode(head);
    HEAD=head;
}

```

```

        for(int i=2;i<=player[1].planes;i++) head=AppendNode(head);
        player[1].planes_present=player[1].planes;
    }
    void UpdateScore(CITE) { //更新得分信息
        setfillcolor(WHITE);
        fillrectangle(300,380,420,405);
        Interface(VALUE);
        char buf[100];
        settextstyle(16,0,"微软黑体");
        sprintf(buf,"本盘得分:  %d",player[1].score);
        setcolor(BLACK);
        outtextxy(300,380,buf);
    }
    void print_board(CITE) { //游戏结束输出得分榜
        IMAGE img3;
        EndBatchDraw();
        ReadGrade(VALUE);
        start_or_pause=!start_or_pause?1:0;
        Interface(VALUE);
        int maxn=player_number==1?0:player[2].score;
        settextstyle(16,0,"微软黑体");
        if(player[1].score>maxn){
            mciSendString("close score_music",NULL,0,NULL);
            mciSendString("open LOAD/MUSIC/获得最高分.mp3 alias score_music",NULL,0,NULL);
            mciSendString("play score_music",NULL,0,NULL);
            char buf[100];
            sprintf(buf,"恭喜你打破最高记录,最高记录更新为%d",player[1].score);
            outtextxy(0,0,buf);
        }
        else{
            mciSendString("close board_music",NULL,0,NULL);
            mciSendString("open LOAD/MUSIC/进入得分榜.mp3 alias board_music",NULL,0,NULL);
            mciSendString("play board_music",NULL,0,NULL);
        }
        WriteGrade(VALUE);
        player[1].if_printed=1;
        getimage(&img3,0,0,640,480);
        MOUSEMSG m;
        while(1){
            if(MouseHit()){

```

```

        m=GetMouseMsg();
        if(m.x>=40&& m.x<=120&& m.y>=420&& m.y<=460) { //开始
            setlinecolor(BLUE);
            rectangle(30, 410, 130, 470);
            if(m.uMsg==WM_LBUTTONDOWN) {
                EndBatchDraw();
                putimage(0, 0, &img3);
                return;
            }
        }
        else if(m.x>=520&& m.x<=600&& m.y>=420&& m.y<=460) { //退出
            setlinecolor(BLUE);
            rectangle(510, 410, 610, 470);
            if(m.uMsg==WM_LBUTTONDOWN) {
                EndBatchDraw();
                putimage(0, 0, &img3);
                Exit(VALUE);
                putimage(0, 0, &img3);
            }
        }
        else putimage(0, 0, &img3);
    }
}

void game(CITE) { //正式游戏流程
    Start_time=clock();
    EndBatchDraw();
    IMAGE img3, img4;
    bool if_img4=0;
    fire *bullet=(fire*) malloc(sizeof(bullet)*100);
    bullet->x=330, bullet->y=360, bullet->if_flying=0;
    angle=PI/2;
    Interface(VALUE);
    MOUSEMSG m;
    setcolor(BLACK);
    IMAGE temp;
    for(11 i=1;;i++) {
        if(!player[1].if_printed) Interface(VALUE);
        while(_kbhit()) { //控制大炮瞄准方向
            char key=_getch();

```

```

        if(key==77||key=='d'||key=='D') angle-=5*DU;//大炮左转
        if(key==75||key=='a'||key=='A') angle+=5*DU;//大炮右转
        if(key==32){//发射弹药
            mciSendString("close bullet_music", NULL, 0, NULL);
            mciSendString("open LOAD/MUSIC/发射弹药.mp3 alias
bullet_music", NULL, 0, NULL);
            mciSendString("play bullet_music", NULL, 0, NULL);
            bool if_bullet_enough=0;
            for(int j=0;j<fire_number;j++)
                if(!(bullet+j)->if_flying){

                    (bullet+j)->if_flying=1, (bullet+j)->b_angle=angle, (bullet+j)->present=i;
                    player[1].score-=(!player[1].score)?0:1, UpdateScore(VALUE);
                    if_bullet_enough=1;
                    break;
                }
            if(!if_bullet_enough){

                (bullet+fire_number)->if_flying=1, (bullet+fire_number)->b_angle=angle, (bullet+fire_
number++)->present=i;
                player[1].score-=(!player[1].score)?0:1, UpdateScore(VALUE);
            }
        }
        loadimage(&img_cannon1, "LOAD/IMAGE/大炮 1. jpg");
        loadimage(&img_cannon2, "LOAD/IMAGE/大炮 2. jpg");
        rotateimage(&img_cannon2, &img_cannon2, angle-90*DU, BLACK);
        rotateimage(&img_cannon1, &img_cannon1, angle-90*DU, WHITE);
        putimage(315, 330, &img_cannon1, SRCAND);
        putimage(315, 330, &img_cannon2, SRCINVERT);
    }
    for(int j=0;j<fire_number;j++)
        if((bullet+j)->if_flying)

        if((bullet+j)->x>=640|| (bullet+j)->x<=0|| (bullet+j)->y>360|| (bullet+j)->y<=0){
            (bullet+j)->if_flying=0;
            (bullet+j)->x=330;
            (bullet+j)->y=360;
        }
        else{

```



```

(bullet+j)->x+=(int) (cos((bullet+j)->b_angle)*fire_speed*100)/100;

(bullet+j)->y+=(int) (sin((bullet+j)->b_angle)*fire_speed*100)/100;
    }
    if(!player[1].if_printed) getimage(&img3,0,0,640,480);
    if(MouseHit()) {
        m=GetMouseMsg();
        if(!player[1].if_printed&& m.x>=40&& m.x<=120&& m.y>=420&& m.y<=460) { //暂停
            setlinecolor(BLUE);
            rectangle(30,410,130,470);
            if(m.uMsg==WM_LBUTTONDOWN) {
                PUTMUSIC
                clock_t start_pause_time=clock();
                EndBatchDraw();
                start_or_pause=!start_or_pause?1:0;
                Interface(VALUE);
                while(start_or_pause) {
                    settextstyle(50,0,"微软黑体");
                    outtextxy(200,100,"GAME PAUSE");
                    settextstyle(12,0,"微软黑体");
                    if(MouseHit()) {
                        m=GetMouseMsg();
                        if(!if_img4) {getimage(&img4,0,0,640,480);if_img4=1;}
                        if(m.x>=40&& m.x<=120&& m.y>=420&& m.y<=460) { //开始
                            setlinecolor(BLUE);
                            rectangle(30,410,130,470);
                            if(m.uMsg==WM_LBUTTONDOWN) {
                                PUTMUSIC
                                EndBatchDraw();
                                putimage(0,0,&img4);
                                start_or_pause=!start_or_pause?1:0;
                                Interface(VALUE);
                                clock_t end_pause_time=clock();

                                pause_time+=(int) ((end_pause_time-start_pause_time)/CLOCKS_PER_SEC);
                                break;
                            }
                        }
                    }
                }
                else if(m.x>=160&& m.x<=240&& m.y>=420&& m.y<=460) { //结束(暂停时)

```

```

        setlinecolor(BLUE);
        rectangle(150, 410, 250, 470);
        if(m. uMsg==WM_LBUTTONDOWN) {
            PUTMUSIC
            EndBatchDraw();
            putimage(0, 0, &img4);

            if(!player[1].planes_present&&!player[1].if_printed)print_board(VALUE); //飞机打完了
            结束游戏

            for(int j=0; j<fire_number; j++)

            free(bullet+j);

            init(VALUE);
            return;
        }
    }
    else if(m. x>=520&&m. x<=600&&m. y>=420&&m. y<=460) { //退出 (暂停时)

        setlinecolor(BLUE);
        rectangle(510, 410, 610, 470);
        if(m. uMsg==WM_LBUTTONDOWN) {
            PUTMUSIC
            EndBatchDraw();
            putimage(0, 0, &img4);
            Exit(VALUE);
            settextstyle(50, 0, "微软黑体");
            outtextxy(200, 100, "GAME PAUSE");
            settextstyle(12, 0, "微软黑体");
        }
    }
    else
    if(m. x>=280&&m. x<=360&&m. y>=420&&m. y<=460) { //保存状态 (暂停时)
        setlinecolor(BLUE);
        rectangle(270, 410, 370, 470);
        if(m. uMsg==WM_LBUTTONDOWN) {
            PUTMUSIC
            EndBatchDraw();
            if_img4=0;
            putimage(0, 0, &img4);
            Store_file(i, VALUE);
        }
    }
}

```

```

    }
    else
if (m. x>=480&&m. x<=540&&m. y>=368&&m. y<=394) { //读档
    setlinecolor(BLUE);
    rectangle(480, 368, 540, 394);
    if (m. uMsg==WM_LBUTTONDOWN) {
        PUTMUSIC
        EndBatchDraw();
        putimage(0, 0, &img4);
        restore=1;
        return;
    }
}
else
if (m. x>=600&&m. x<=625&&m. y>=330&&m. y<=355&&m. uMsg==WM_LBUTTONDOWN) { //音乐关闭打开
    if_music_open=(if_music_open+1)%2;
    Interface(VALUE);
    !if_music_open?mciSendString("pause
bkmusic", NULL, 0, NULL):mciSendString("resume bkmusic", NULL, 0, NULL);
    getimage(&img4, 0, 0, 640, 480);
}
else putimage(0, 0, &img4);
}
}
}
}
else if (m. x>=160&&m. x<=240&&m. y>=420&&m. y<=460) { //结束 (游戏中)
    setlinecolor(BLUE);
    rectangle(150, 410, 250, 470);
    if (m. uMsg==WM_LBUTTONDOWN) {
        EndBatchDraw();
        PUTMUSIC
        putimage(0, 0, &img3);

        if (!player[1]. planes_present&&!player[1]. if_printed) print_board(VALUE); //飞机打完了
        结束游戏

        free(bullet);
        init(VALUE);
        return;
    }
}

```

```

    }
    else
if(!player[1].if_printed&& m.x>=400&&m.x<=480&&m.y>=420&&m.y<=460) { //增加飞机
    setlinecolor(BLUE);
    rectangle(390,410,490,470);
    if(m.uMsg==WM_LBUTTONDOWN) {
        PUTMUSIC
        EndBatchDraw();
        HEAD=AppendNode(HEAD);
        player[1].planes++;
        player[1].planes_present++;
        putimage(0,0,&img3);
    }
}
else if(m.x>=520&&m.x<=600&&m.y>=420&&m.y<=460) { //退出
    setlinecolor(BLUE);
    rectangle(510,410,610,470);
    if(m.uMsg==WM_LBUTTONDOWN) {
        PUTMUSIC
        clock_t start_pause_time=clock();
        EndBatchDraw();
        putimage(0,0,&img3);
        Exit(VALUE);
        clock_t end_pause_time=clock();

pause_time+=(int)((end_pause_time-start_pause_time)/CLOCKS_PER_SEC);
    }
}
else if(m.x>=280&&m.x<=360&&m.y>=420&&m.y<=460) { //保存状态
    setlinecolor(BLUE);
    rectangle(270,410,370,470);
    if(m.uMsg==WM_LBUTTONDOWN) {
        PUTMUSIC
        EndBatchDraw();
        if_img4=0;
        putimage(0,0,&img4);
        Store_file(i,VALUE);
    }
}
else if(m.x>=480&&m.x<=540&&m.y>=368&&m.y<=394) { //读档

```

```

        setlinecolor(BLUE);
        rectangle(480, 368, 540, 394);
        if(m. uMsg==WM_LBUTTONDOWN) {
            PUTMUSIC
            EndBatchDraw();
            putimage(0, 0, &img4);
            restore=1;
            return;
        }
    }
    else
        if(m. x>=600&&m. x<=625&&m. y>=330&&m. y<=355&&m. uMsg==WM_LBUTTONDOWN) { //音乐关闭打开
            if_music_open=(if_music_open+1)%2;
            Interface(VALUE);
            !if_music_open?mciSendString("pause
bkmusic", NULL, 0, NULL):mciSendString("resume bkmusic", NULL, 0, NULL);
            getimage(&img4, 0, 0, 640, 480);
        }
        else putimage(0, 0, &img3);
    }
    if(!player[1]. if_printed) {
        BeginBatchDraw(); //开始在缓冲区绘图
        End_time=clock();
        game_time=(int)((End_time-Start_time)/CLOCKS_PER_SEC)-pause_time; //游戏时
        间=总时间-暂停时间
        char buf[100];
        sprintf_s(buf, "%2dmin %2dsec", game_time/60, game_time%60);
        putimage(315, 330, &img_cannon1, SRCAND);
        putimage(315, 330, &img_cannon2, SRCINVERT);
        settextstyle(15, 0, "微软黑体");
        outtextxy(530, 10, buf);
        settextstyle(12, 0, "微软黑体");
        planes *p=HEAD;
        bool if_damage=0;
        setcolor(BLACK);
        for(int j=0; j<fire_number; j++) if((bullet+j)->if_flying)
            putimage((bullet+j)->x, (bullet+j)->y, &img_fire);
        while(p!=NULL) { //贴飞机
            for(int j=0; j<fire_number; j++) //遍历炮弹

```

```

        if(pow((bullet+j)->x-(10+(i-p->start_time+mod)%mod))+pow((bullet+j)->y-10-p->height)
<=200&&!p->if_damaged){//飞机命中
            mciSendString("close explode_music", NULL, 0, NULL);
            mciSendString("open LOAD/MUSIC/飞机爆炸.mp3 alias
explode_music", NULL, 0, NULL);
            mciSendString("play explode_music", NULL, 0, NULL);
            p->explode_time=i;
            p->explode_x=(i-p->start_time+mod)%mod;
            p->explode_y=p->height;
            player[1].score+=3;
            if_damage=1;
            p->if_damaged=1;
            (bullet+j)->if_flying=0;
            player[1].planes_present--;
            (bullet+j)->x=330;
            (bullet+j)->y=360;
        }
        if(!p->if_firsttime){

            if(!p->if_damaged)p->plane_kind?putimage((i-p->start_time+mod)%mod,p->height,&img_P
lane1, SRCAND):putimage((i-p->start_time+mod)%mod,p->height,&img_Plane2, SRCAND);
            else{
                ll time=(i-p->explode_time)/15;
                if(time<4)
p->plane_kind?putimage(p->explode_x,p->explode_y,&img_planeExplode1[time], SRCAND):putim
age(p->explode_x,p->explode_y,&img_planeExplode2[time], SRCAND);
            }
        }
        if(p->if_firsttime&&i>=p->start_time) p->if_firsttime=0;
        p=p->next;
    }
    FlushBatchDraw();Sleep(1);//刷新输出界面
    if(if_damage) UpdateScore(VALUE);
}
}

void mouses(CITE){//初始时选择功能
    MOUSEMSG m;
    IMAGE img5;
    getimage(&img5, 0, 0, 640, 480);

```

```

while(1) {
    m=GetMouseMsg();
    if(m.x>=40&&m.x<=120&&m.y>=420&&m.y<=460) { // “开始/暂停”
        setlinecolor(BLUE);
        rectangle(30, 410, 130, 470);
        if(m.uMsg==WM_LBUTTONDOWN) {
            PUTMUSIC
            EndBatchDraw();
            putimage(0, 0, &img5);
            if(!ifplayer) input(), ifplayer=1; //注册用户名和初始飞机数量
            if(start_or_pause) {
                start_or_pause=0;
                game(VALUE);
                return;
            }
        }
    }
    else if(m.x>=520&&m.x<=600&&m.y>=420&&m.y<=460) { // “退出”
        setlinecolor(BLUE);
        rectangle(510, 410, 610, 470);
        if(m.uMsg==WM_LBUTTONDOWN) {
            PUTMUSIC
            EndBatchDraw();
            putimage(0, 0, &img5);
            Exit(VALUE);
        }
    }
    else if(m.x>=600&&m.x<=625&&m.y>=330&&m.y<=355&&m.uMsg==WM_LBUTTONDOWN) { // 音乐
        关闭打开
        if_music_open=(if_music_open+1)%2;
        Interface(VALUE);
        getimage(&img5, 0, 0, 640, 480);
        !if_music_open?mciSendString("pause
bkmusic", NULL, 0, NULL):mciSendString("resume bkmusic", NULL, 0, NULL);
    }
    else putimage(0, 0, &img5);
}

void Load_picture_music() { //加载游戏各图片和音乐
    mciSendString("open LOAD/MUSIC/背景音乐.mp3 alias bkmusic", NULL, 0, NULL);
}

```

```

mciSendString("play bkmusic repeat", NULL, 0, NULL);
loadimage(&music, "LOAD/IMAGE/音乐键. jpg");
loadimage(&shutdown_music, "LOAD/IMAGE/音乐关闭键. jpg");
loadimage(&img_bk, "LOAD/IMAGE/背景. jpg");
loadimage(&img_cannon1, "LOAD/IMAGE/大炮 1. jpg");
loadimage(&img_cannon2, "LOAD/IMAGE/大炮 2. jpg");
loadimage(&img_Plane1, "LOAD/IMAGE/飞机 1. jpg");
loadimage(&img_Plane2, "LOAD/IMAGE/飞机 2. jpg");
loadimage(&img_fire, "LOAD/IMAGE/子弹. jpg");
loadimage(&img_planeExplode2[0], "LOAD/IMAGE/飞机 2 爆炸 1. jpg");
loadimage(&img_planeExplode2[1], "LOAD/IMAGE/飞机 2 爆炸 2. jpg");
loadimage(&img_planeExplode2[2], "LOAD/IMAGE/飞机 2 爆炸 3. jpg");
loadimage(&img_planeExplode2[3], "LOAD/IMAGE/飞机 2 爆炸 4. jpg");
loadimage(&img_planeExplode1[0], "LOAD/IMAGE/飞机 1 爆炸 1. jpg");
loadimage(&img_planeExplode1[1], "LOAD/IMAGE/飞机 1 爆炸 2. jpg");
loadimage(&img_planeExplode1[2], "LOAD/IMAGE/飞机 1 爆炸 3. jpg");
loadimage(&img_planeExplode1[3], "LOAD/IMAGE/飞机 1 爆炸 4. jpg");
}

void If_continue(CITE) { //刚进入游戏时是否读档
    if_file=0;
    setfillcolor(WHITE);
    fillrectangle(0, 0, 640, 480);
    putimage(0, 0, &img_bk);
    setbkmode(TRANSPARENT);
    IMAGE img3, img4;
    setcolor(BLACK);
    getimage(&img4, 0, 0, 640, 480);
    settextstyle(30, 0, "微软黑体");
    outtextxy(125, 100, "您有一局未结束已保存的游戏");
    outtextxy(225, 130, "是否确认读档");
    settextstyle(12, 0, "微软黑体");
    rectangle(225, 170, 275, 190);
    rectangle(355, 170, 405, 190);
    outtextxy(240, 174, "读档");
    outtextxy(370, 174, "重开");
    getimage(&img3, 0, 0, 640, 480);
    MOUSEMSG m;
    while(1) {
        EndBatchDraw();
        m=GetMouseMsg();
    }
}

```



```

        if(m.x>=225&& m.x<=275&& m.y>=170&& m.y<=190) { //读档
            setlinecolor(BLUE);
            rectangle(220,165,280,195);
            if(m.uMsg==WM_LBUTTONDOWN) {
                ifplayer=1,start_or_pause=0;
                Interface(VALUE);
                game(VALUE);
                break;
            }
        }
    }
    else if(m.x>=355&& m.x<=405&& m.y>=170&& m.y<=190) { //重开
        setlinecolor(BLUE);
        rectangle(350,165,410,195);
        if(m.uMsg==WM_LBUTTONDOWN) {
            Interface(VALUE);
            mouses(VALUE);
            break;
        }
    }
    else putimage(0,0,&img3);
}

int main() {
    int
    game_time=0,pause_time=0,player_number=0,start_or_pause=1,ifplayer=0,fire_number=1,if_f
    ile=0,if_store=0,restore=0,if_music_open=1;
    //游戏时间；暂停时间；历史玩家数量；开始/暂停；是否有玩家正在游戏；弹药已有数量；是
    否有存档；是否读档；是否恢复；音乐是否打开
    double angle=PI/2; //大炮旋转角度
    Load_picture_music();
    while(1) {
        initgraph(640,480);
        graphdefaults();
        Read_file(VALUE);
        if(restore) {
            restore=0,ifplayer=1,start_or_pause=0;
            Interface(VALUE);
            game(VALUE);
        }
        else if(if_file) If_continue(VALUE);
    }
}

```

```
        else Interface(VALUE),mouses(VALUE);
    }
    EndBatchDraw();
    closegraph();
    return 0;
}
```