/\*最后这个代码就是我们目前最完整、运行得最好的代码了，这个代码完美解决了产生敌机

以及让敌机的移动与扫描键盘输入冲突等问题，而且也解决了光标乱闪的问题

在程序的第201，我们采用静态变量，让 飞机的坐标位置改变的循环 参与的是主程序的while循环，

这样修改的话，用户按键交互速度还是不受影响，还能保证敌机正常移动。

在基本功能实现后我们又进一步完善程序，包括加入了启动和结束的界面，添加启动页面BGM、飞机发射

子弹的音效、敌机被摧毁的音效、敌机撞击飞机的音效让游戏更优化；同时我们也在游戏界面右侧加入了

各种静态信息如控制按键提示、加速减速提示、修改颜色提示、暂停提示等等，又加入了各种动态信息如

当前血量、得分信息等等让程序更完善

\*/

#include <stdio.h>

#include <stdlib.h>

#include <conio.h>

#include<pthread.h>

#include<windows.h>

#pragma comment(lib, "Winmm.lib")

// 全局变量

int position\_x,position\_y; // 飞机位置

int bullet\_x,bullet\_y; // 子弹位置

int enemy\_x,enemy\_y; // 敌机位置

int high,width; // 游戏画面尺寸

int score; // 得分

int speed; //敌机速度

int speed1; //敌机速度参照

int flag; //循环执行标示

int flag1; //循环标示参照

int life; //生命数

int box; //军火箱

int box\_x,box\_y; //军火箱的位置

void gotoxy(int x,int y)//类似于清屏函数，光标移动到原点位置进行重画

{

HANDLE hOut;

COORD pos= {x,y};

hOut = GetStdHandle(STD\_OUTPUT\_HANDLE);

SetConsoleCursorPosition(hOut, pos);

}

void GameStart()//游戏开始

{

PlaySound(TEXT("BGM.wav"), NULL, SND\_FILENAME | SND\_ASYNC | SND\_LOOP);

draw();

gotoxy(6, 7); printf("Airplane Fire Game");

gotoxy(11, 9); printf("Welcome!");

gotoxy(12, 11); printf("Hello!");

gotoxy(8, 14); system("pause");

}

void GameOver()//游戏结束

{

PlaySound(TEXT("defeat.wav"), NULL, SND\_FILENAME | SND\_ASYNC);

system("cls");

draw();

gotoxy(9, 8); printf("Game Over");

gotoxy(10, 10); printf("score:%d", score);

gotoxy(2, 12);

printf("Press any key to continue.");

printf("\n");

printf("\n");

printf("\n");

printf("\n");

printf("\n");

printf("\n");

printf("\n");

printf("\n");

printf("\n");

printf("\n");

getch();

exit(0);

}

void startup() // 数据初始化

{

high=20;

width=30;

position\_x=high-4;

position\_y=width/2;

bullet\_x=-2;

bullet\_y=position\_y;

enemy\_x=0;

enemy\_y=position\_y;

box\_x=0;

box\_y=position\_y+7;

score=0;

speed1=20;

flag=0;

flag1=0;

life=5;

}

int draw()

{

int i,j;

for(i=0;i<=30;i++)

{

gotoxy(i,20);

printf("#");

}

for(j=0;j<=19;j++)

{

gotoxy(30,j);

printf("#");

}

return 0;

}

void show() // 显示画面

{

int i,j;

draw();

dynamic\_info();

gotoxy(0,0); // 光标移动到原点位置进行重画清屏

if(box==0)//一次发射一发子弹

{

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

{

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

{

if ((i==position\_x-4) && (j==position\_y)||(i==position\_x-3) && (j==position\_y-1))

printf("\*"); // 输出飞机\*

else if ((i==position\_x-3) && (j==position\_y)||(i==position\_x-3) && (j==position\_y+1))

printf("\*"); // 输出飞机\*

else if ((i==position\_x-2) && (j==position\_y-2)||(i==position\_x-2) && (j==position\_y-1))

printf("\*"); // 输出飞机\*

else if ((i==position\_x-2) && (j==position\_y)||(i==position\_x-2) && (j==position\_y+1))

printf("\*"); // 输出飞机\*

else if ((i==position\_x-2) && (j==position\_y+2)||(i==position\_x-1) && (j==position\_y-1)||(i==position\_x-1) && (j==position\_y+1))

printf("\*"); // 输出飞机\*

else if ((i==enemy\_x) && (j==enemy\_y))

printf("\*"); // 输出敌机\*

else if ((i==box\_x) && (j==box\_y))

printf("@"); // 输出军火箱@

else if ((i==bullet\_x) && (j==bullet\_y))

printf("|"); // 输出子弹|

else

printf(" "); // 输出空格

}

printf("\n");

}

}

else

{

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

{

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

{

if ((i==position\_x-4) && (j==position\_y)||(i==position\_x-3) && (j==position\_y-1))

printf("\*"); // 输出飞机\*

else if ((i==position\_x-3) && (j==position\_y)||(i==position\_x-3) && (j==position\_y+1))

printf("\*"); // 输出飞机\*

else if ((i==position\_x-2) && (j==position\_y-2)||(i==position\_x-2) && (j==position\_y-1))

printf("\*"); // 输出飞机\*

else if ((i==position\_x-2) && (j==position\_y)||(i==position\_x-2) && (j==position\_y+1))

printf("\*"); // 输出飞机\*

else if ((i==position\_x-2) && (j==position\_y+2)||(i==position\_x-1) && (j==position\_y-1)||(i==position\_x-1) && (j==position\_y+1))

printf("\*"); // 输出飞机\*

else if ((i==enemy\_x) && (j==enemy\_y))

printf("\*"); // 输出敌机\*

else if ((i==box\_x) && (j==box\_y))

printf("@"); // 输出军火箱@

else if ((i==bullet\_x) && (j==bullet\_y)||(i==bullet\_x+1) && (j==bullet\_y-1)||(i==bullet\_x+1) && (j==bullet\_y+1))

printf("|"); // 输出子弹|

else

printf(" "); // 输出空格

}

printf("\n");

}

}

}

void static\_info()//静态信息，如游戏侧栏文字提示

{

gotoxy(high+14,0); printf("%c %c %c %c 控制移动方向", 'w', 'a', 's', 'd');

gotoxy(width+4,2); printf("按p键暂停");

gotoxy(width+4,4); printf("F1加速,F2减速");

gotoxy(width+4,6); printf("得分:");

gotoxy(width+4,8); printf("生命:");

gotoxy(width+4,10); SetConsoleTextAttribute(GetStdHandle(STD\_OUTPUT\_HANDLE), FOREGROUND\_RED |

FOREGROUND\_INTENSITY); printf("F4:■");//红色

gotoxy(width+4,12); SetConsoleTextAttribute(GetStdHandle(STD\_OUTPUT\_HANDLE), FOREGROUND\_GREEN | FOREGROUND\_RED |

FOREGROUND\_INTENSITY); printf("F5:■");//黄色

gotoxy(width+4,14); SetConsoleTextAttribute(GetStdHandle(STD\_OUTPUT\_HANDLE), FOREGROUND\_GREEN |

FOREGROUND\_INTENSITY); printf("F6:■");//绿 色

gotoxy(width+4,16); SetConsoleTextAttribute(GetStdHandle(STD\_OUTPUT\_HANDLE), FOREGROUND\_BLUE | FOREGROUND\_RED |

FOREGROUND\_INTENSITY); printf("F7:■");//紫色

gotoxy(width+4,18); SetConsoleTextAttribute(GetStdHandle(STD\_OUTPUT\_HANDLE), FOREGROUND\_GREEN | FOREGROUND\_BLUE | FOREGROUND\_RED |

FOREGROUND\_INTENSITY); printf("F8:■");//亮白色

gotoxy(width+4,20); SetConsoleTextAttribute(GetStdHandle(STD\_OUTPUT\_HANDLE), FOREGROUND\_GREEN | FOREGROUND\_BLUE |

FOREGROUND\_RED);printf("F9:■");//灰色

}

int dynamic\_info()//动态信息，如游戏得分

{

gotoxy(width+10, 6); printf("%d", score);

gotoxy(width+10, 8); printf("%d", life);

}

void updateWithoutInput() // 与用户输入无关的更新

{

if (bullet\_x>-2)

bullet\_x--;

if (box==0)

{

if ((bullet\_x==box\_x) && (bullet\_y==box\_y)) // 子弹击中火药箱

{

PlaySound(TEXT("boom.wav"), NULL, SND\_FILENAME | SND\_ASYNC);

box=1;

box\_x=-1; // 产生新的火药箱

box\_y=rand()%width;

if(box\_y==1)box\_y++;

if(box\_y==29)box\_y=box\_y-3;

if(box\_y==28)box\_y--;

bullet\_x=-3; // 子弹无效

}

if ((bullet\_x==enemy\_x) && (bullet\_y==enemy\_y)) // 子弹击中敌机

{

PlaySound(TEXT("boom.wav"), NULL, SND\_FILENAME | SND\_ASYNC);

score++;

dynamic\_info(); // 分数加1

enemy\_x=-1; // 产生新的飞机

enemy\_y=rand()%width;

if(enemy\_y==1)enemy\_y++;

if(enemy\_y==29)enemy\_y=enemy\_y-3;

if(enemy\_y==28)enemy\_y--;

bullet\_x=-3; // 子弹无效

}

}

else

{

if ((bullet\_x==box\_x) && (bullet\_y==box\_y)||(bullet\_x+1==box\_x) && (bullet\_y-1==box\_y)

||(bullet\_x+1==box\_x) && (bullet\_y+1==box\_y)) // 子弹击中火药箱

{

PlaySound(TEXT("boom.wav"), NULL, SND\_FILENAME | SND\_ASYNC);

box=1;

box\_x=-1; // 产生新的火药箱

box\_y=rand()%width;

if(box\_y==1)box\_y++;

if(box\_y==29)box\_y=box\_y-3;

if(box\_y==28)box\_y--;

bullet\_x=-3; // 子弹无效

}

if ((bullet\_x==enemy\_x) && (bullet\_y==enemy\_y)||(bullet\_x+1==enemy\_x) && (bullet\_y-1==enemy\_y)

||(bullet\_x+1==enemy\_x) && (bullet\_y+1==enemy\_y)) // 子弹击中敌机

{

PlaySound(TEXT("boom.wav"), NULL, SND\_FILENAME | SND\_ASYNC);

score++;

dynamic\_info(); // 分数加1

enemy\_x=-1; // 产生新的飞机

enemy\_y=rand()%width;

if(enemy\_y==1)enemy\_y++;

if(enemy\_y==29)enemy\_y=enemy\_y-3;

if(enemy\_y==28)enemy\_y--;

bullet\_x=-3; // 子弹无效

}

}

if (enemy\_x>high) // 敌机跑出显示屏幕

{

life--;

dynamic\_info();

flag1++;

enemy\_x=-1; // 产生新的飞机

enemy\_y=rand()%width;

if(enemy\_y==1)enemy\_y++;

if(enemy\_y==29)enemy\_y=enemy\_y-3;

if(enemy\_y==28)enemy\_y--;

}

if (box\_x>high) // 火药箱跑出显示屏幕

{

box\_x=-1; // 产生新的火药箱

box\_y=rand()%width;

if(box\_y==1)box\_y++;

if(box\_y==29)box\_y=box\_y-3;

if(box\_y==28)box\_y--;

}

if ((enemy\_x==position\_x-4)&&(enemy\_y==position\_y)||(enemy\_x==position\_x-3)&&(enemy\_y==position\_y-1) //敌机碰撞飞机

||(enemy\_x==position\_x-3)&&(enemy\_y==position\_y+1)||(enemy\_x==position\_x-2)&&(enemy\_y==position\_y-2)

||(enemy\_x==position\_x-2)&&(enemy\_y==position\_y+2))

{

box=0;

PlaySound(TEXT("strike.wav"), NULL, SND\_FILENAME | SND\_ASYNC);

life--;

dynamic\_info();

flag1++;

enemy\_x=-1; // 产生新的飞机

enemy\_y=rand()%width;

if(enemy\_y==1)enemy\_y++;

if(enemy\_y==29)enemy\_y=enemy\_y-3;

if(enemy\_y==28)enemy\_y--;

}

// 用来控制敌机向下移动的速度。每隔几次循环，才移动一次敌机

// 这样修改的话，用户按键交互速度还是保持很快，但我们NPC的移动显示可以降速

static int speed = 0;

if (speed<speed1)

speed++;

if (speed==speed1)

{

enemy\_x++;

box\_x++;

speed = 0;

}

}

void updateWithInput() // 与用户输入有关的更新

{

char input;

if(kbhit()) // 判断是否有输入

{

input=getch(); // 根据用户的不同输入来移动，不必输入回车

if (input=='a')

{position\_y--;

if(position\_y==1)position\_y++;

} // 位置左移

if (input=='d')

{position\_y++;

if(position\_y==28)position\_y--;

} // 位置右移

if (input=='w')

{position\_x--;

if(position\_x==3)position\_x++;

} // 位置上移

if (input=='s')

{position\_x++;

if(position\_x==21)position\_x--;

} // 位置下移

if (input=='p')while(getch()!='p');

if (input==59 )

{

speed1--;

if(speed1==6)speed1++;

}

if (input==60 )

{

speed1++;

if(speed1==36)speed1=speed1--;

}

if (input==62 )

{system("color 0C"); static\_info(); SetConsoleTextAttribute(GetStdHandle(STD\_OUTPUT\_HANDLE), FOREGROUND\_RED |

FOREGROUND\_INTENSITY); }

if (input==63 )

{system("color 0E"); static\_info(); SetConsoleTextAttribute(GetStdHandle(STD\_OUTPUT\_HANDLE), FOREGROUND\_GREEN | FOREGROUND\_RED |

FOREGROUND\_INTENSITY); }

if (input==64 )

{system("color 0A"); static\_info(); SetConsoleTextAttribute(GetStdHandle(STD\_OUTPUT\_HANDLE), FOREGROUND\_GREEN |

FOREGROUND\_INTENSITY); }

if (input==65 )

{system("color 0D"); static\_info(); SetConsoleTextAttribute(GetStdHandle(STD\_OUTPUT\_HANDLE), FOREGROUND\_BLUE | FOREGROUND\_RED |

FOREGROUND\_INTENSITY); }

if (input==66 )

{system("color 0F"); static\_info(); SetConsoleTextAttribute(GetStdHandle(STD\_OUTPUT\_HANDLE), FOREGROUND\_GREEN | FOREGROUND\_BLUE |

FOREGROUND\_RED | FOREGROUND\_INTENSITY); }

if (input==67 )

{system("color 08"); static\_info(); SetConsoleTextAttribute(GetStdHandle(STD\_OUTPUT\_HANDLE), FOREGROUND\_GREEN | FOREGROUND\_BLUE |

FOREGROUND\_RED);}

if (input==' ') // 发射子弹

{

PlaySound(TEXT("fire(1).wav"), NULL, SND\_FILENAME | SND\_ASYNC);

bullet\_x=position\_x-2; // 发射子弹的初始位置在飞机的正上方

bullet\_y=position\_y;

}

}

}

void sweep()//清光标

{

HANDLE hOut=GetStdHandle(STD\_OUTPUT\_HANDLE);

CONSOLE\_CURSOR\_INFO cci;

GetConsoleCursorInfo(hOut,&cci);//获取光标信息

cci.bVisible=FALSE; //隐藏光标

SetConsoleCursorInfo(hOut,&cci);//设置光标信息

}

void direction()

{

FILE \*fp;

char a;

if ((fp=fopen("Airplane fire文档说明.txt","r"))==NULL)

{printf("Cannot open file!\n");exit(0);}

while((a=fgetc(fp))!=EOF)

putchar(a);

fclose(fp);

printf("\n");

system("pause");

system("cls");

}

void main()

{

direction();

GameStart();

PlaySound(NULL, NULL, SND\_FILENAME | SND\_ASYNC | SND\_LOOP);

startup(); // 数据初始化

static\_info();

sweep();

while (!flag) // 游戏循环执行

{

show(); // 显示画面

updateWithoutInput(); // 与用户输入无关的更新

if(flag1==5)flag=!flag;

updateWithInput(); // 与用户输入有关的更新

}

GameOver();

}