

(2021.6.17)——版本4.0

```
/* 版本:陨石撞飞机4.0
 * 时间: 2021.6.17
 * 作者: orall
 * 完善了: 1.增加声音(爆炸声音和开始声音)
 *          2.界面闪烁
 *          3.改变游戏的高度和宽度,增强游戏的可玩性
 *          4.增加难度
 *          5.得分机制和胜利机制
 *          6.完善界面(开始的界面)
 * 本版本完善
 *          7.飞机和陨石碰撞判定的准确度
 *          8.重新开局飞机会炸
 *          8.飞机可以移动到边界
 * 待完善: 1.得分标签的闪烁
 *          2.重新开始界面没有出现“按空格键开始”
 */

// planeDlg.cpp: 实现文件
//

#include "pch.h"
#include "framework.h"
#include "plane.h"
#include "planeDlg.h"
#include "afxdialogex.h"

//game相关的头文件
#include<vector>
#include <iostream>
#include <stdlib.h>
#include <time.h>
#include <conio.h>
#include <windows.h>

//playsound的头文件
#include <mmSystem.h>
#pragma comment(lib,"winmm.lib")

int start = -1;           //判断游戏是否开始(-1表示在主菜单)
int m_pause = 1;

//使图片透明化(参考地址: https://blog.csdn.net/hust\_xy/article/details/9313239)
void TransparentPNG(CImage* png)
```

```

{
for (int i = 0; i < png->GetWidth(); i++)
{
    for (int j = 0; j < png->GetHeight(); j++)
    {
        unsigned char* pucColor = reinterpret_cast<unsigned char*>(png->GetPixelAddress(i,
j));

        pucColor[0] = pucColor[0] * pucColor[3] / 255;
        pucColor[1] = pucColor[1] * pucColor[3] / 255;
        pucColor[2] = pucColor[2] * pucColor[3] / 255;
    }
}
}

#ifdef _DEBUG
#define new DEBUG_NEW
#endif

// 用于应用程序“关于”菜单项的 CAboutDlg 对话框

class CAboutDlg : public CDialogEx
{
public:
    CAboutDlg();

// 对话框数据
#ifdef AFX_DESIGN_TIME
    enum { IDD = IDD_ABOUTBOX };
#endif

protected:
    virtual void DoDataExchange(CDataExchange* pDX);    // DDX/DDV 支持

// 实现
protected:
    DECLARE_MESSAGE_MAP()
};

CAboutDlg::CAboutDlg() : CDialogEx(IDD_ABOUTBOX)
{
}

void CAboutDlg::DoDataExchange(CDataExchange* pDX)
{
    CDialogEx::DoDataExchange(pDX);
}

```

```

}

BEGIN_MESSAGE_MAP(CAboutDlg, CDialogEx)
END_MESSAGE_MAP()

// CplaneDlg 对话框

CplaneDlg::CplaneDlg(CWnd* pParent /*=nullptr*/)
    : CDialogEx(IDD_PLANE_DIALOG, pParent)
{
    m_hIcon = AfxGetApp()->LoadIcon(IDR_MAINFRAME);
}

void CplaneDlg::DoDataExchange(CDataExchange* pDX)
{
    CDialogEx::DoDataExchange(pDX);
}

BEGIN_MESSAGE_MAP(CplaneDlg, CDialogEx)
    ON_WM_SYSCOMMAND()
    ON_WM_PAINT()
    ON_WM_QUERYDRAGICON()
    ON_BN_CLICKED(IDC_BUTTON1, &CplaneDlg::OnBnClickedButton1)
    ON_BN_CLICKED(IDC_BUTTON3, &CplaneDlg::OnBnClickedButton3)
    ON_BN_CLICKED(IDC_BUTTON2, &CplaneDlg::OnBnClickedButton2)
    ON_WM_KEYDOWN()
    ON_WM_TIMER()
    ON_WM_CTLCOLOR()
END_MESSAGE_MAP()

// CplaneDlg 消息处理程序

BOOL CplaneDlg::OnInitDialog()
{
    CDialogEx::OnInitDialog();

    // 将“关于...”菜单项添加到系统菜单中。

    // IDM_ABOUTBOX 必须在系统命令范围内。
    ASSERT((IDM_ABOUTBOX & 0xFFF0) == IDM_ABOUTBOX);
    ASSERT(IDM_ABOUTBOX < 0xF000);

```

```

CMenu* pSysMenu = GetSystemMenu(FALSE);
if (pSysMenu != nullptr)
{
    BOOL bNameValid;
    CString strAboutMenu;
    bNameValid = strAboutMenu.LoadString(IDS_ABOUTBOX);
    ASSERT(bNameValid);
    if (!strAboutMenu.IsEmpty())
    {
        pSysMenu->AppendMenu(MF_SEPARATOR);
        pSysMenu->AppendMenu(MF_STRING, IDM_ABOUTBOX, strAboutMenu);
    }
}

// 设置此对话框的图标。 当应用程序主窗口不是对话框时，框架将自动
// 执行此操作
SetIcon(m_hIcon, TRUE);          // 设置大图标
SetIcon(m_hIcon, FALSE);         // 设置小图标

// TODO: 在此添加额外的初始化代码
//显示控件(注意静态文本框需要改一个ID名才可以进行相关操作,改成其他控件的别名就行)

//设置窗口大小( 其中SetWindowPos函数的第四和第五个参数分别表示窗口的宽度,和高度,大家可以调整这两个参数
来改变自己窗口的大小。)
//SetWindowPos(NULL, 0, 0, 700, 500, SWP_NOMOVE);

return TRUE;  // 除非将焦点设置到控件，否则返回 TRUE
}

void CplaneDlg::OnSysCommand(UINT nID, LPARAM lParam)
{
    if ((nID & 0xFFF0) == IDM_ABOUTBOX)
    {
        CAboutDlg dlgAbout;
        dlgAbout.DoModal();
    }
    else
    {
        CDialogEx::OnSysCommand(nID, lParam);
    }
}

// 如果向对话框添加最小化按钮，则需要下面的代码
// 来绘制该图标。 对于使用文档/视图模型的 MFC 应用程序，
// 这将由框架自动完成。

```

```

void CplaneDlg::OnPaint()
{
    if (IsIconic())
    {
        CPaintDC dc(this); // 用于绘制的设备上下文

        SendMessage(WM_ICONERASEBKGND, reinterpret_cast<WPARAM>(dc.GetSafeHdc()), 0);

        // 使图标在工作区矩形中居中
        int cxIcon = GetSystemMetrics(SM_CXICON);
        int cyIcon = GetSystemMetrics(SM_CYICON);
        CRect rect;
        GetClientRect(&rect);
        int x = (rect.Width() - cxIcon + 1) / 2;
        int y = (rect.Height() - cyIcon + 1) / 2;

        // 绘制图标
        dc.DrawIcon(x, y, m_hIcon);
    }
    else
    {
        CDialogEx::OnPaint();
    }
    if (start == -1) {

        //设置背景（参考链接: https://blog.csdn.net/qq321772514/article/details/96478790）
        GetClientRect(&m_client);
        //注意这里的位置（图片放到res文件里面）
        CImage main;
        main.Load(_T("res\\main.jpg"));
        //这里让图片绘制区域和窗口大小一致
        main.Draw(GetDC()->GetSafeHdc(), CRect(0, 0, m_client.Width(), m_client.Height()));
    }

    if (start == 1) {

        CDC* cDC = this->GetDC(); //获取窗口DC指针
        GetClientRect(&m_client);

        //创建缓冲DC
        m_cacheDC.CreateCompatibleDC(NULL);
        m_cacheCBitmap.CreateCompatibleBitmap(cDC, m_client.Width(), m_client.Height());
        m_cacheDC.SelectObject(&m_cacheCBitmap);

        //画背景
    }
}

```

```

        background.Draw(m_cacheDC, CRect(0, 0, m_client.Width(), m_client.Height()));
        m_plane.Draw(m_cacheDC, m_planepos); //贴飞机

        for (int i = 0; i < max_stonenum; i++) {
            if ( is_stone[i] == 1) { //如果石头存在
                m_stone[i].Draw(m_cacheDC, m_stonepos[i]); //贴石头
            }
        }

        CString b;
        //注意加_T
        b.Format(_T("当前得分: %d"), score);
        cDC->SetBkMode(TRANSPARENT); //透明, 否则新的遮住旧的
        cDC->SetTextColor( RGB(27, 161, 194) ); //设置控件字体颜色
        //设置标签字体以及大小
        (https://jingyan.baidu.com/article/0a52e3f4f18eb5bf62ed72ea.html)
        CFont font;
        font.CreatePointFont(150, _T("方正舒体"));
        cDC->SelectObject(&font);

        cDC->TextOut(m_client.Width() - 160, 10, b);

        //最后将缓冲DC内容输出到窗口DC中
        cDC->BitBlt(0, 0, m_client.Width(), m_client.Height(), &m_cacheDC, 0, 0, SRCCOPY);

        //绘制结束

        //在绘制完图后,使窗口区有效
        ValidateRect(&m_client);
        //释放缓冲DC
        m_cacheDC.DeleteDC();
        //释放对象
        m_cacheCBitmap.DeleteObject();
        //释放窗口DC
        ReleaseDC(cDC);
    }
}

//当用户拖动最小化窗口时系统调用此函数取得光标
//显示。
HCURSOR CplaneDlg::OnQueryDragIcon()
{
    return static_cast<HCURSOR>(m_hIcon);
}

```

```

void CplaneDlg::OnBnClickedButton1()           //开始游戏
{
    // TODO: 在此添加控件通知处理程序代码
    //隐藏控件
    CWnd* pWnd = GetDlgItem(IDC_STATIC);
    pWnd->ShowWindow(FALSE);
    pWnd = GetDlgItem(IDC_BUTTON1);
    pWnd->ShowWindow(FALSE);
    pWnd = GetDlgItem(IDC_BUTTON2);
    pWnd->ShowWindow(FALSE);
    pWnd = GetDlgItem(IDC_BUTTON3);
    pWnd->ShowWindow(FALSE);

    start = 1;

    //设置背景（参考链接: https://blog.csdn.net/qq321772514/article/details/96478790）
    GetClientRect(&m_client);
    //注意这里的位置（把plane.png放到res文件里面）
    background.Load(_T("res\\background.jpg"));
    //这里让图片绘制区域和窗口大小一致
    background.Draw(GetDC()->GetSafeHdc(), CRect(0, 0, m_client.Width(),
m_client.Height()));

    //加载飞机图片
    m_plane.Load(_T("res\\plane.png"));
    TransparentPNG(&m_plane);           //使图片透明化

    //加载石头图片
    for (int i = 0; i < max_stonenum; i++) {
        m_stone[i].Load(_T("res\\stone.png"));
        TransparentPNG(&m_stone[i]);           //使图片透明化
    }

    //设置飞机初始位置
    m_planepos.left = m_client.Width() / 2 - 20;           //飞机左边
    m_planepos.right = m_planepos.left + 40;           //飞机的右边等于左边加上飞机的宽度
    m_planepos.top = m_client.Height() - 40;
    m_planepos.bottom = m_planepos.top + 40;

    OnPaint();

    //设置石头初始位置（在顶部随机出现）           【660 430】

```

```

        srand(unsigned(time(NULL)));
        m_stonepos[0].left = rand() % (m_client.Width() - 40);
        m_stonepos[0].right = m_stonepos[0].left + 40;
        m_stonepos[0].top = 0;
        m_stonepos[0].bottom = 40;

        is_stone[0] = 1;

        if (m_pause == 1) {
            CDC *cDC = this->GetDC();
            CString c;
            //注意加_T
            c.Format(_T("按空格键开始游戏"));
            cDC->SetBkMode(TRANSPARENT); //透明，否则新的遮住旧的
            cDC->SetTextColor(RGB(27, 161, 194)); //设置控件字体颜色
            //设置标签字体以及大小
            (https://jingyan.baidu.com/article/0a52e3f4f18eb5bf62ed72ea.html)
            CFont font;
            font.CreatePointFont(180, _T("方正舒体"));
            cDC->SelectObject(&font);

            cDC->TextOut(m_client.Width()/2 - 120, m_client.Height() / 2 - 50, c);
        }
    }

void CplaneDlg::OnBnClickedButton3() //退出
{
    // TODO: 在此添加控件通知处理程序代码
    exit(1);
}

void CplaneDlg::OnBnClickedButton2() //操作说明
{
    // TODO: 在此添加控件通知处理程序代码
    MessageBox(TEXT("● 操作说明：使用方向键【上下左右】操控飞机来躲避陨石，躲避时间越多得到的分数越高，难度也会随着改变哦，而且越往高处飞分数增长的越多哦！"), TEXT("★ 提示"), MB_OK);
}

//监听键盘按下（参考链接：https://blog.csdn.net/hust\_xy/article/details/9327377）
void CplaneDlg::OnKeyDown(UINT nChar, UINT nRepCnt, UINT nFlags)
{
}

```



```

//监听键盘按下（参考链接：https://blog.csdn.net/weixin\_42899088/article/details/101308231）
BOOL CplaneDlg::PreTranslateMessage(MSG* pMsg)
{
    // TODO: 在此添加专用代码和/或调用基类
    // 键盘按键检测
    if (pMsg->message == WM_KEYDOWN) // 按键按下检测
    {
        if (pMsg->wParam == VK_SPACE) { //按空格键开始
            if (m_pause == 1) {
                m_pause = 0;
                //循环播放背景音乐
                PlaySound(_T("res\\background.wav"), NULL, SND_ASYNC | SND_FILENAME |
SND_LOOP);

                SetTimer(1, down_speed, NULL); //设置定时器1(陨石下落)
                SetTimer(2, new_speed, NULL); //设置定时器2(产生新陨石)
                SetTimer(3, 100, NULL); //设置定时器3(检测飞机移动情况)
                SetTimer(4, 2000, NULL); //设置定时器4(检测飞机存活情况，加分数)
            }

        }

        if (pMsg->wParam == VK_LEFT && m_planepos.left - plane_speed >= 0 ) {
            moving_left = true;
            OnPaint(); //更新
        }
        else if (pMsg->wParam == VK_RIGHT && m_planepos.right + plane_speed <=
m_client.Width() ) {
            moving_right = true;
            OnPaint(); //更新
        }
        else if (pMsg->wParam == VK_UP && m_planepos.top - plane_speed >= 0 ) {
            moving_up = true;
            OnPaint(); //更新
        }
        else if (pMsg->wParam == VK_DOWN && m_planepos.bottom + plane_speed <=
m_client.Height() ) {
            moving_down = true;
            OnPaint(); //更新
        }

    }

    else if (pMsg->message == WM_KEYUP) {

        if (pMsg->wParam == VK_SPACE) {

```

```

        m_pause = 0;
        //return CDialogEx::PreTranslateMessage(pMsg);
    }

    if(pMsg->wParam == VK_LEFT ) {
        moving_left = false;
    }
    else if (pMsg->wParam == VK_RIGHT ) {
        moving_right = false;
    }
    else if (pMsg->wParam == VK_UP ) {
        moving_up = false;
    }
    else if (pMsg->wParam == VK_DOWN ) {
        moving_down = false;
    }
    }
    return CDialogEx::PreTranslateMessage(pMsg);
}

//教程: https://blog.csdn.net/hust\_xy/article/details/9332377
void CplaneDlg::OnTimer(UINT_PTR nIDEvent)
{
    // TODO: 在此添加消息处理程序代码和/或调用默认值
    switch (nIDEvent)
    {
    case 1:
        for (int i = 0; i < max_stonenum; i++) {
            if (is_stone[i] == 1) {
                if (m_stonepos[i].bottom <= m_client.Height() + 100 ) {
                    m_stonepos[i].top += down_speed;
                    m_stonepos[i].bottom += down_speed;
                }
                if (m_stonepos[i].top > m_client.Height() ) {
                    is_stone[i] = 0;
                }
            }
        }

        OnPaint();
        //爆炸
        if (( abs(m_planepos.top - m_stonepos[i].top) < 25 ) && abs(m_stonepos[i].left
- m_planepos.left) < 25 && is_stone[i] == 1) {
            start = 0;          //游戏结束

```

```

for (int i = 0; i < max_stonenum; i++) {
    is_stone[i] = 0;
}

CRect tmp = m_planepos;

//让他们隐藏起来
m_stonepos[i].top = -100;
m_stonepos[i].bottom = -100 + 40;

m_planepos.top = -100;
m_planepos.bottom = -100 + 40;

OnPaint();

//记得加头文件而且必须是wav格式（在线MP3转wav: https://onlineconvertfree.com）
PlaySound(_T("res\\bombmusic.wav"), NULL, SND_ASYNC );//播放爆炸音响

//加载爆炸图片
CImage bomb;
bomb.Load(_T("res\\bomb.png"));
TransparentPNG(&bomb); //使图片透明化
bomb.Draw(GetDC()->GetSafeHdc(), tmp);
KillTimer(1);
KillTimer(2);
KillTimer(3);
KillTimer(4);

CString ts;
ts.Format(_T("游戏结束，你的最终得分是%d,是否重新开始一局新的游戏? "),score);

if (MessageBox(ts, TEXT("★ 提示"), MB_YESNO) == IDYES ) {
    //初始化
    for (int i = 0; i < max_stonenum; i++) {
        is_stone[i] = 0;
    }
    //重新开始一局游戏
    m_pause = 1;
    start = 1;
    score = 0;
    down_speed = 10; //石头下落的速度
    new_speed = 500; //石头产生的速度
    moving_right = false;
    moving_left = false;
}

```

度

```
moving_up = false;
moving_down = false;

//设置飞机初始位置
m_planepos.left = m_client.Width() / 2 - 20;           //飞机左边
m_planepos.right = m_planepos.left + 40;             //飞机的右边等于左边加上飞机的宽

m_planepos.top = m_client.Height() - 40;
m_planepos.bottom = m_planepos.top + 40;

OnPaint();

//设置石头初始位置（在顶部随机出现）           【660 430】
m_stonepos[0].left = rand() % (m_client.Width() - 40);
m_stonepos[0].right = m_stonepos[0].left + 40;
m_stonepos[0].top = 0;
m_stonepos[0].bottom = 40;

is_stone[0] = 1;

if ( m_pause == 1 ) {
    CDC* cDC = this->GetDC();
    CString c;
    //注意加_T
    c.Format(_T("按空格键开始游戏"));
    cDC->SetBkMode(TRANSPARENT); //透明，否则新的遮住旧的
    cDC->SetTextColor(RGB(27, 161, 194)); //设置控件字体颜色
    //设置标签字体以及大小
    (https://jingyan.baidu.com/article/0a52e3f4f18eb5bf62ed72ea.html)
    CFont font;
    font.CreatePointFont(180, _T("方正舒体"));
    cDC->SelectObject(&font);

    cDC->TextOut(m_client.Width() / 2 - 120, m_client.Height() / 2 - 50,
c);
}

}

else {
    exit(1);
}

;
```

```

    }
}

break;

case 2:
{
    for (int i = 0; i < max_stonenum; i++) {
        if (this->is_stone[i] == 0) {           //没有石头的话
            is_stone[i] = 1;                   //标记为有石头
            //srand(unsigned(time(NULL)));      不能调用两次，否则不会随机
            m_stonepos[i].left = rand()%(m_client.Width() - 40);
            m_stonepos[i].right = m_stonepos[i].left + 40;
            m_stonepos[i].top = 0;
            m_stonepos[i].bottom = 40;
            break;
        }
    }
}

break;

case 3:           //允许不断移动（即按住键也可以移动）
    if ( moving_left && m_planepos.left - plane_speed >= 0 ) {
        m_planepos.left -= 10;
        m_planepos.right -= 10;
    }
    else if (moving_right && m_planepos.right + plane_speed <= m_client.Width() ) {
        m_planepos.left += 10;
        m_planepos.right += 10;
    }
    else if (moving_up && m_planepos.top - plane_speed >= 0 ) {
        m_planepos.top -= 10;
        m_planepos.bottom -= 10;
    }
    else if (moving_down && m_planepos.bottom + plane_speed <= m_client.Height() ) {
        m_planepos.top += 10;
        m_planepos.bottom += 10;
    }

    break;

case 4 :
{
    //每过2秒加5分，高度得分
    if (start == 1) {
        score += 10 + (m_client.Height() - m_planepos.Height()) /100;

        //初始速度

```

```

        //int down_speed = 10;          //石头下落的速度
        ///int new_speed = 500;        //石头产生的速度

        //越往后速度越快
        down_speed = down_speed + score / 50 * 0.1;
        new_speed = new_speed - score / 50 * 0.2;
    }

}

break;
default:
    break;
}

CDialogEx::OnTimer(nIDEvent);
}

//把标签那成透明（参考链接：https://blog.csdn.net/zhangchao772329724/article/details/83182463?
utm_medium=distribute.pc_relevant_t0.none-task-blog-2%7Edefault%7EBlogCommendFromMachineLearnPai2%7Edefault-1.control&depth_1-utm_source=distribute.pc_relevant_t0.none-task-blog-2%7Edefault%7EBlogCommendFromMachineLearnPai2%7Edefault-1.control)
HBRUSH CplaneDlg::OnCtlColor(CDC* pDC, CWnd* pWnd, UINT nCtlColor)
{
    HBRUSH hbr = CDialogEx::OnCtlColor(pDC, pWnd, nCtlColor);

    // TODO: 在此更改 DC 的任何特性
    if (pWnd->GetDlgCtrlID() == IDC_STATIC )      //IDC_DEVICE_STAT为Static控件的ID
    {
        //27,161,194
        //52,36,227
        //pDC->SetTextColor(RGB(27, 161, 194));    //设置控件字体颜色
        pDC->SetTextColor(RGB(123, 104, 238));    //设置控件字体颜色
        pDC->SetBkMode(TRANSPARENT);    //设置透明属性

        //设置标签字体以及大小
        (https://jingyan.baidu.com/article/0a52e3f4f18eb5bf62ed72ea.html)
        CFont font;
        font.CreatePointFont(250, _T("方正舒体"));
        pDC->SelectObject(&font);

        return (HBRUSH)GetStockObject(NULL_BRUSH);    //返回空画刷
    }

    // TODO: 如果默认的不是所需画笔，则返回另一个画笔

```

```
    return hbr;  
}
```

```
// planeDlg.h: 头文件  
//  
  
#pragma once  
  
#define MAX 30                //石头出现的最大数量  
  
// CplaneDlg 对话框  
class CplaneDlg : public CDialogEx  
{  
// 构造  
public:  
    CplaneDlg(CWnd* pParent = nullptr); // 标准构造函数  
  
    //添加的数据  
    CImage background;  
    CImage m_plane;  
    CImage m_stone[MAX];  
    CRect m_planepos;  
    CRect m_stonepos[MAX];  
  
    int max_stonenum = MAX;        //最大的石头数量（最多出现10个）  
    int is_stone[MAX] = {0};      //石头是否存在  
    int down_speed = 10;          //石头下落的速度  
    int new_speed = 500;          //石头产生的速度  
    int plane_speed = 10;         //飞机移动速度  
  
    //移动标志，允许飞机不断移动（即按住键可继续移动）  
    bool moving_right = false;  
    bool moving_left = false;  
    bool moving_up = false;  
    bool moving_down = false;  
  
    int score = 0;                //分数  
  
    //解决图片闪烁  
    CRect m_client;               //保存客户区大小  
    CDC m_cacheDC;               //缓冲DC  
    CBitmap m_cacheCBitmap;      //缓冲位图
```

```

// 对话框数据
#ifdef AFX_DESIGN_TIME
    enum { IDD = IDD_PLANE_DIALOG };
#endif

protected:
    virtual void DoDataExchange(CDataExchange* pDX);    // DDX/DDV 支持

// 实现
protected:
    HICON m_hIcon;

    // 生成的消息映射函数
    virtual BOOL OnInitDialog();
    afx_msg void OnSysCommand(UINT nID, LPARAM lParam);
    afx_msg void OnPaint();
    afx_msg HCURSOR OnQueryDragIcon();
    DECLARE_MESSAGE_MAP()
public:
    afx_msg void OnBnClickedButton1();
    afx_msg void OnBnClickedButton3();
    afx_msg void OnBnClickedButton2();
    afx_msg void OnKeyDown(UINT nChar, UINT nRepCnt, UINT nFlags);
    virtual BOOL PreTranslateMessage(MSG* pMsg);
    afx_msg void OnTimer(UINT_PTR nIDEvent);
    afx_msg HBRUSH OnCtlColor(CDC* pDC, CWnd* pWnd, UINT nCtlColor);
    afx_msg void OnStnClickedStatic3();
};

```

(2021.6.16) ——版本3.0

```

/* 版本:陨石撞飞机3.0
* 时间: 2021.6.16
* 作者: orall
* 完善了: 1.增加声音(爆炸声音和开始声音)
*          2.界面闪烁
*          3.改变游戏的高度和宽度, 增强游戏的可玩性
*          4.增加难度
*          5.得分机制和胜利机制
*          6.完善界面(开始的界面)
* 待完善: 1.得分标签的闪烁
*          2.飞机和陨石碰撞判定的准确度

```



```

*          3.重新开局飞机会炸
*          4.飞机可以移动到边界
*/

// planeDlg.cpp: 实现文件
//

#include "pch.h"
#include "framework.h"
#include "plane.h"
#include "planeDlg.h"
#include "afxdialogex.h"

//game相关的头文件
#include<vector>
#include <iostream>
#include <stdlib.h>
#include <time.h>
#include <conio.h>
#include <windows.h>

//playsound的头文件
#include <mmSystem.h>
#pragma comment(lib,"winmm.lib")

int start = -1;          //判断游戏是否开始（-1表示在主菜单）

//使图片透明化（参考地址：https://blog.csdn.net/hust\_xy/article/details/9313239）
void TransparentPNG(CImage* png)
{
    for (int i = 0; i < png->GetWidth(); i++)
    {
        for (int j = 0; j < png->GetHeight(); j++)
        {
            unsigned char* pucColor = reinterpret_cast<unsigned char*>(png->GetPixelAddress(i, j));
            pucColor[0] = pucColor[0] * pucColor[3] / 255;
            pucColor[1] = pucColor[1] * pucColor[3] / 255;
            pucColor[2] = pucColor[2] * pucColor[3] / 255;
        }
    }
}

#ifdef _DEBUG
#define new DEBUG_NEW
#endif

```

```

// 用于应用程序“关于”菜单项的 CAboutDlg 对话框

class CAboutDlg : public CDialogEx
{
public:
    CAboutDlg();

// 对话框数据
#ifdef AFX_DESIGN_TIME
    enum { IDD = IDD_ABOUTBOX };
#endif

protected:
    virtual void DoDataExchange(CDataExchange* pDX);    // DDX/DDV 支持

// 实现
protected:
    DECLARE_MESSAGE_MAP()
};

CAboutDlg::CAboutDlg() : CDialogEx(IDD_ABOUTBOX)
{
}

void CAboutDlg::DoDataExchange(CDataExchange* pDX)
{
    CDialogEx::DoDataExchange(pDX);
}

BEGIN_MESSAGE_MAP(CAboutDlg, CDialogEx)
END_MESSAGE_MAP()

// CplaneDlg 对话框

CplaneDlg::CplaneDlg(CWnd* pParent /*=nullptr*/)
    : CDialogEx(IDD_PLANE_DIALOG, pParent)
{
    m_hIcon = AfxGetApp()->LoadIcon(IDR_MAINFRAME);
}

void CplaneDlg::DoDataExchange(CDataExchange* pDX)

```

```

{
    CDialogEx::DoDataExchange(pDX);
}

BEGIN_MESSAGE_MAP(CplaneDlg, CDialogEx)
    ON_WM_SYSCOMMAND()
    ON_WM_PAINT()
    ON_WM_QUERYDRAGICON()
    ON_BN_CLICKED(IDC_BUTTON1, &CplaneDlg::OnBnClickedButton1)
    ON_BN_CLICKED(IDC_BUTTON3, &CplaneDlg::OnBnClickedButton3)
    ON_BN_CLICKED(IDC_BUTTON2, &CplaneDlg::OnBnClickedButton2)
    ON_WM_KEYDOWN()
    ON_WM_TIMER()
    ON_WM_CTLCOLOR()
END_MESSAGE_MAP()

// CplaneDlg 消息处理程序

BOOL CplaneDlg::OnInitDialog()
{
    CDialogEx::OnInitDialog();

    // 将“关于...”菜单项添加到系统菜单中。

    // IDM_ABOUTBOX 必须在系统命令范围内。
    ASSERT((IDM_ABOUTBOX & 0xFFF0) == IDM_ABOUTBOX);
    ASSERT(IDM_ABOUTBOX < 0xF000);

    CMenu* pSysMenu = GetSystemMenu(FALSE);
    if (pSysMenu != nullptr)
    {
        BOOL bNameValid;
        CString strAboutMenu;
        bNameValid = strAboutMenu.LoadString(IDS_ABOUTBOX);
        ASSERT(bNameValid);
        if (!strAboutMenu.IsEmpty())
        {
            pSysMenu->AppendMenu(MF_SEPARATOR);
            pSysMenu->AppendMenu(MF_STRING, IDM_ABOUTBOX, strAboutMenu);
        }
    }

    // 设置此对话框的图标。 当应用程序主窗口不是对话框时，框架将自动
    // 执行此操作
    SetIcon(m_hIcon, TRUE);        // 设置大图标

```

```

        SetIcon(m_hIcon, FALSE);           // 设置小图标

// TODO: 在此添加额外的初始化代码
//显示控件(注意静态文本框需要改一个ID名才可以进行相关操作,改成其他控件的别名就行)
CWnd* pWnd = GetDlgItem(IDOK);
pWnd->ShowWindow(FALSE);

//设置窗口大小( 其中SetWindowPos函数的第四和第五个参数分别表示窗口的宽度,和高度,大家可以调整这两个参数
来改变自己窗口的大小。)
//SetWindowPos(NULL, 0, 0, 700, 500, SWP_NOMOVE);

return TRUE; // 除非将焦点设置到控件, 否则返回 TRUE
}

void CplaneDlg::OnSysCommand(UINT nID, LPARAM lParam)
{
    if ((nID & 0xFFF0) == IDM_ABOUTBOX)
    {
        CAboutDlg dlgAbout;
        dlgAbout.DoModal();
    }
    else
    {
        CDialogEx::OnSysCommand(nID, lParam);
    }
}

// 如果向对话框添加最小化按钮, 则需要下面的代码
// 来绘制该图标。 对于使用文档/视图模型的 MFC 应用程序,
// 这将由框架自动完成。

void CplaneDlg::OnPaint()
{
    if (IsIconic())
    {
        CPaintDC dc(this); // 用于绘制的设备上下文

        SendMessage(WM_ICONERASEBKGND, reinterpret_cast<WPARAM>(dc.GetSafeHdc()), 0);

        // 使图标在工作区矩形中居中
        int cxIcon = GetSystemMetrics(SM_CXICON);
        int cyIcon = GetSystemMetrics(SM_CYICON);
        CRect rect;
        GetClientRect(&rect);
        int x = (rect.Width() - cxIcon + 1) / 2;
        int y = (rect.Height() - cyIcon + 1) / 2;
    }
}

```

```

        // 绘制图标
        dc.DrawIcon(x, y, m_hIcon);
    }
    else
    {
        CDialogEx::OnPaint();
    }
    if (start == -1) {

        //设置背景（参考链接: https://blog.csdn.net/qq321772514/article/details/96478790）
        GetClientRect(&m_client);
        //注意这里的位置（图片放到res文件里面）
        CImage main;
        main.Load(_T("res\\main.jpg"));
        //这里让图片绘制区域和窗口大小一致
        main.Draw(GetDC()->GetSafeHdc(), CRect(0, 0, m_client.Width(), m_client.Height()));

    }

    if (start == 1) {
        CDC* cDC = this->GetDC();          //获取窗口DC指针
        GetClientRect(&m_client);

        //创建缓冲DC
        m_cacheDC.CreateCompatibleDC(NULL);
        m_cacheCBitmap.CreateCompatibleBitmap(cDC, m_client.Width(), m_client.Height());
        m_cacheDC.SelectObject(&m_cacheCBitmap);

        //画背景
        background.Draw(m_cacheDC, CRect(0, 0, m_client.Width(), m_client.Height()));
        m_plane.Draw(m_cacheDC, m_planepos);//贴飞机

        for (int i = 0; i < this->max_stonenum; i++) {
            if (this->is_stone[i] == 1) {          //如果石头存在
                m_stone[i].Draw(m_cacheDC, m_stonepos[i]);      //贴石头
            }
        }

        CString tmp;
        //注意加_T
        tmp.Format(_T("当前得分: %d"), score);
        GetDlgItem(IDOK)->SetWindowText(tmp);
    }
}

```

```

        //最后将缓冲DC内容输出到窗口DC中
        cDC->BitBlt(0, 0, m_client.Width(), m_client.Height(), &m_cacheDC, 0, 0, SRCCOPY);

        //—————绘制结束—————

        //在绘制完图后,使窗口区有效
        ValidateRect(&m_client);
        //释放缓冲DC
        m_cacheDC.DeleteDC();
        //释放对象
        m_cacheCBitmap.DeleteObject();
        //释放窗口DC
        ReleaseDC(cDC);

        //显示控件

        CWnd* pWnd = GetDlgItem(IDOK);
        pWnd->SetWindowPos(&CWnd::wndTopMost, 0, 0, 0, 0, SWP_NOSIZE | SWP_NOMOVE);
        //GetDlgItem(IDOK)->ShowWindow(SW_HIDE);

    }

}

//当用户拖动最小化窗口时系统调用此函数取得光标
//显示。
HCURSOR CplaneDlg::OnQueryDragIcon()
{
    return static_cast<HCURSOR>(m_hIcon);
}

void CplaneDlg::OnBnClickedButton1() //开始游戏
{
    // TODO: 在此添加控件通知处理程序代码

```

```
//隐藏控件
CWnd* pWnd = GetDlgItem(IDC_STATIC);
pWnd->ShowWindow(FALSE);
pWnd = GetDlgItem(IDC_BUTTON1);
pWnd->ShowWindow(FALSE);
pWnd = GetDlgItem(IDC_BUTTON2);
pWnd->ShowWindow(FALSE);
pWnd = GetDlgItem(IDC_BUTTON3);
pWnd->ShowWindow(FALSE);
//显示控件
pWnd = GetDlgItem(IDOK);
pWnd->ShowWindow(TRUE);

CString b;
//注意加_T
b.Format(_T("当前得分: %d"), score);
GetDlgItem(IDOK)->SetWindowText(b);

//显示控件

// pWnd = GetDlgItem(IDOK);
//pWnd->SetWindowPos(&CWnd::wndTopMost, 0, 0, 0, 0, SWP_NOSIZE | SWP_NOMOVE);

//pWnd->BringWindowToTop();

start = 1;

//循环播放背景音乐
PlaySound(_T("res\\background.wav"), NULL, SND_ASYNC | SND_FILENAME | SND_LOOP);

//设置背景（参考链接: https://blog.csdn.net/qq321772514/article/details/96478790）
GetClientRect(&m_client);
//注意这里的位置（把plane.png放到res文件里面）
background.Load(_T("res\\background.jpg"));
//这里让图片绘制区域和窗口大小一致
background.Draw(GetDC()->GetSafeHdc(), CRect(0, 0, m_client.Width(),
m_client.Height()));

//加载飞机图片
m_plane.Load(_T("res\\plane.png"));
TransparentPNG(&m_plane); //使图片透明化

//加载石头图片
for (int i = 0; i < this->max_stonenum; i++) {
    m_stone[i].Load(_T("res\\stone.png"));
}
```

```

        TransparentPNG(&m_stone[i]);          //使图片透明化
    }

    //设置飞机初始位置
    m_planepos.left = m_client.Width() / 2 - 20;          //飞机左边
    m_planepos.right = m_planepos.left + 40;          //飞机的右边等于左边加上飞机的宽度
    m_planepos.top = m_client.Height() - 40;
    m_planepos.bottom = m_planepos.top + 40;

    //设置石头初始位置（在顶部随机出现）          【660 430】
    srand(unsigned(time(NULL)));
    m_stonepos[0].left = rand() % (m_client.Width() - 40);
    m_stonepos[0].right = m_stonepos[0].left + 40;
    m_stonepos[0].top = 0;
    m_stonepos[0].bottom = 40;

    is_stone[0] = 1;

    SetTimer(1, down_speed, NULL);          //设置定时器1(陨石下落)
    SetTimer(2, new_speed, NULL);          //设置定时器2(产生新陨石)
    SetTimer(3, 100, NULL);          //设置定时器3(检测飞机移动情况)
    SetTimer(4, 2000, NULL);          //设置定时器4(检测飞机存活情况，加分数)

}

void CplaneDlg::OnBnClickedButton3()          //退出
{
    // TODO: 在此添加控件通知处理程序代码
    exit(1);
}

void CplaneDlg::OnBnClickedButton2()          //操作说明
{
    // TODO: 在此添加控件通知处理程序代码
    MessageBox(TEXT("● 操作说明：使用方向键【上下左右】操控飞机来躲避陨石，躲避时间越多得到的分数越高，难度也会随着改变哦，而且越往高处飞分数增长的越多哦！"), TEXT("★ 提示"), MB_OK);
}

//监听键盘按下（参考链接：https://blog.csdn.net/hust\_xy/article/details/9327377）
void CplaneDlg::OnKeyDown(UINT nChar, UINT nRepCnt, UINT nFlags)
{

```



//监听键盘按下（参考链接：[https://blog.csdn.net/weixin\\_42899088/article/details/101308231](https://blog.csdn.net/weixin_42899088/article/details/101308231)）

BOOL CplaneDlg::PreTranslateMessage(MSG\* pMsg)

{

    // TODO: 在此添加专用代码和/或调用基类

    // 键盘按键检测

    if (pMsg->message == WM\_KEYDOWN)// 按键按下检测

    {

        if (pMsg->wParam == VK\_LEFT && m\_planepos.left - plane\_speed >= 0 ) {

            moving\_left = true;

        }

        else if (pMsg->wParam == VK\_RIGHT && m\_planepos.right + plane\_speed <= m\_client.Width() ) {

            moving\_right = true;

        }

        else if (pMsg->wParam == VK\_UP && m\_planepos.top - plane\_speed >= 0 ) {

            moving\_up = true;

        }

        else if (pMsg->wParam == VK\_DOWN && m\_planepos.bottom + plane\_speed <= m\_client.Height() ) {

            moving\_down = true;

        }

        OnPaint();        //更新

    }

    else if (pMsg->message == WM\_KEYUP) {

        if(pMsg->wParam == VK\_LEFT ) {

            moving\_left = false;

        }

        else if (pMsg->wParam == VK\_RIGHT ) {

            moving\_right = false;

        }

        else if (pMsg->wParam == VK\_UP ) {

            moving\_up = false;

        }

        else if (pMsg->wParam == VK\_DOWN ) {

            moving\_down = false;

        }

    }

    return CDialogEx::PreTranslateMessage(pMsg);

}

//教程: [https://blog.csdn.net/hust\\_xy/article/details/9332377](https://blog.csdn.net/hust_xy/article/details/9332377)

```
void CplaneDlg::OnTimer(UINT_PTR nIDEvent)
```

```
{
```

```
    // TODO: 在此添加消息处理程序代码和/或调用默认值
```

```
    switch (nIDEvent)
```

```
    {
```

```
    case 1:
```

```
        for (int i = 0; i < max_stonenum; i++) {
```

```
            if (is_stone[i] == 1) {
```

```
                if (m_stonepos[i].bottom <= m_client.Height() + 100 ) {
```

```
                    m_stonepos[i].top += down_speed;
```

```
                    m_stonepos[i].bottom += down_speed;
```

```
                }
```

```
                if (m_stonepos[i].top > m_client.Height() ) {
```

```
                    is_stone[i] = 0;
```

```
                }
```

```
            }
```

```
        OnPaint();
```

```
        //爆炸
```

```
        if (( abs(m_planepos.top - m_stonepos[i].top) < 30 ) && abs(m_stonepos[i].left - m_planepos.left) < 35) {
```

```
            start = 0;                //游戏结束
```

```
            CRect tmp = m_planepos;
```

```
            //让他们隐藏起来
```

```
            m_stonepos[i].top = -100;
```

```
            m_stonepos[i].bottom = -100 + 40;
```

```
            m_planepos.top = -100;
```

```
            m_planepos.bottom = -100 + 40;
```

```
            OnPaint();
```

```
            //记得加头文件而且必须是wav格式（在线MP3转wav: https://onlineconvertfree.com）
```

```
            PlaySound(_T("res\\bombmusic.wav"), NULL, SND_ASYNC );//播放爆炸音响
```

```
            //加载爆炸图片
```

```
            CImage bomb;
```

```
            bomb.Load(_T("res\\bomb.png"));
```

```
            TransparentPNG(&bomb);        //使图片透明化
```

```
            bomb.Draw(GetDC()->GetSafeHdc(), tmp);
```

```

KillTimer(1);
KillTimer(2);
//Sleep(2000);

CString ts;
ts.Format(_T("游戏结束，你的最终得分是%d,是否重新开始一局新的游戏? "),score);

if (MessageBox(ts, TEXT("★ 提示"), MB_YESNO) == IDYES ) {
    //初始化
    for (int i = 0; i < max_stonenum; i++) {
        is_stone[i] = 0;
    }

    //重新开始一局游戏

    start = 1;
    score = 0;

    down_speed = 10;           //石头下落的速度
    new_speed = 500;           //石头产生的速度

    moving_right = false;
    moving_left = false;
    moving_up = false;
    moving_down = false;

    //设置飞机初始位置
    m_planepos.left = m_client.Width()/2 - 20;           //飞机左边
    m_planepos.right = m_planepos.left + 40;           //飞机的右边等于左边加上飞机的宽
    度

    m_planepos.top = m_client.Height()-40;
    m_planepos.bottom = m_planepos.top + 40;

    //设置石头初始位置（在顶部随机出现）           【660 430】
    srand(unsigned(time(NULL)));
    m_stonepos[0].left = rand() % (m_client.Width()-40);
    m_stonepos[0].right = m_stonepos[0].left + 40;
    m_stonepos[0].top = 0;
    m_stonepos[0].bottom = 40;

    is_stone[0] = 1;

    //循环播放背景音乐
    PlaySound(_T("res\\background.wav"), NULL, SND_ASYNC | SND_FILENAME |
SND_LOOP);

```

```

        SetTimer(1, down_speed, NULL);        //设置定时器1(陨石下落)
        SetTimer(2, new_speed, NULL);        //设置定时器2(产生新陨石)

    }
    else {
        exit(1);
    }
    ;

}

}

break;

case 2:
{
    for (int i = 0; i < max_stonenum; i++) {
        if (this->is_stone[i] == 0) {          //没有石头的话
            is_stone[i] = 1;                  //标记为有石头
            //srand(unsigned(time(NULL)));      不能调用两次，否则不会随机
            m_stonepos[i].left = rand()%(m_client.Width() - 40);
            m_stonepos[i].right = m_stonepos[i].left + 40;
            m_stonepos[i].top = 0;
            m_stonepos[i].bottom = 40;
            break;
        }
    }
}

break;

case 3:          //允许不断移动（即按住键也可以移动）
    if (moving_left) {
        m_planepos.left -= 10;
        m_planepos.right -= 10;
    }
    else if (moving_right) {
        m_planepos.left += 10;
        m_planepos.right += 10;
    }
    else if (moving_up) {
        m_planepos.top -= 10;
        m_planepos.bottom -= 10;
    }
    else if (moving_down) {
        m_planepos.top += 10;

```

```

        m_planepos.bottom += 10;
    }

    break;
case 4 :
{
    //每过2秒加5分，高度得分
    if (start == 1) {
        score += 10 + (m_client.Height() - m_planepos.Height()) / 100;
    }
    CString tmp;
    //注意加_T
    tmp.Format(_T("当前得分: %d"), score);
    GetDlgItem(IDOK)->SetWindowText(tmp);

    //初始速度
    //int down_speed = 10;           //石头下落的速度
    //int new_speed = 500;           //石头产生的速度

    //越往后速度越快
    down_speed = down_speed + score / 100 * 1;
    new_speed = new_speed - score / 100 * 2;

}

    break;
default:
    break;
}

CDialogEx::OnTimer(nIDEvent);
}

//把标签那成透明（参考链接：https://blog.csdn.net/zhangchao772329724/article/details/83182463?
utm_medium=distribute.pc_relevant_t0.none-task-blog-2%7Edefault%7EBlogCommendFromMachineLearnPai2%7Edefault-1.control&depth_1-utm_source=distribute.pc_relevant_t0.none-task-blog-2%7Edefault%7EBlogCommendFromMachineLearnPai2%7Edefault-1.control)
HBRUSH CplaneDlg::OnCtlColor(CDC* pDC, CWnd* pWnd, UINT nCtlColor)
{
    HBRUSH hbr = CDialogEx::OnCtlColor(pDC, pWnd, nCtlColor);

    // TODO: 在此更改 DC 的任何特性

```

```

if (pWnd->GetDlgCtrlID() == IDC_STATIC )    //IDC_DEVICE_STAT为Static控件的ID
{
    //27,161,194
    //52,36,227
    //pDC->SetTextColor(RGB(27, 161, 194));    //设置控件字体颜色
    pDC->SetTextColor(RGB(52, 36, 227));    //设置控件字体颜色
    pDC->SetBkMode(TRANSPARENT);    //设置透明属性

    //设置标签字体以及大小
    (https://jingyan.baidu.com/article/0a52e3f4f18eb5bf62ed72ea.html)
    CFont font;
    font.CreatePointFont(250, _T("方正舒体"));
    pDC->SelectObject(&font);

    return (HBRUSH)GetStockObject(NULL_BRUSH);    //返回空画刷
}

if ( pWnd->GetDlgCtrlID() == IDOK )    //Static控件的ID
{
    //27,161,194
    //52,36,227
    pDC->SetTextColor(RGB(27, 161, 194));    //设置控件字体颜色
    //pDC->SetTextColor(RGB(52, 36, 227));    //设置控件字体颜色
    pDC->SetBkMode(TRANSPARENT);    //设置透明属性

    //设置标签字体以及大小
    (https://jingyan.baidu.com/article/0a52e3f4f18eb5bf62ed72ea.html)
    CFont font;
    font.CreatePointFont(2, _T("方正舒体"));
    pDC->SelectObject(&font);

    return (HBRUSH)GetStockObject(NULL_BRUSH);    //返回空画刷
}

// TODO: 如果默认的不是所需画笔, 则返回另一个画笔
return hbr;
}

void CplaneDlg::OnStnClickedStatic3()
{
    // TODO: 在此添加控件通知处理程序代码
}

// planeDlg.h: 头文件
//

```

```

#pragma once

#define MAX 30                //石头出现的最大数量

// CplaneDlg 对话框
class CplaneDlg : public CDialogEx
{
// 构造
public:
    CplaneDlg(CWnd* pParent = nullptr); // 标准构造函数

    //添加的数据
    CImage background;
    CImage m_plane;
    CImage m_stone[MAX];
    CRect m_planepos;
    CRect m_stonepos[MAX];

    int max_stonenum = MAX;        //最大的石头数量（最多出现10个）
    int is_stone[MAX] = {0};      //石头是否存在
    int down_speed = 10;          //石头下落的速度
    int new_speed = 500;          //石头产生的速度
    int plane_speed = 10;         //飞机移动速度

    //移动标志，允许飞机不断移动（即按住键可继续移动）
    bool moving_right = false;
    bool moving_left = false;
    bool moving_up = false;
    bool moving_down = false;

    int score = 0;                //分数

    //解决图片闪烁
    CRect m_client;               //保存客户区大小
    CDC m_cacheDC;                //缓冲DC
    CBitmap m_cacheCBitmap;       //缓冲位图

// 对话框数据
#ifdef AFX_DESIGN_TIME
    enum { IDD = IDD_PLANE_DIALOG };

```

```

#endif

protected:
    virtual void DoDataExchange(CDataExchange* pDX);    // DDX/DDV 支持

// 实现
protected:
    HICON m_hIcon;

    // 生成的消息映射函数
    virtual BOOL OnInitDialog();
    afx_msg void OnSysCommand(UINT nID, LPARAM lParam);
    afx_msg void OnPaint();
    afx_msg HCURSOR OnQueryDragIcon();
    DECLARE_MESSAGE_MAP()
public:
    afx_msg void OnBnClickedButton1();
    afx_msg void OnBnClickedButton3();
    afx_msg void OnBnClickedButton2();
    afx_msg void OnKeyDown(UINT nChar, UINT nRepCnt, UINT nFlags);
    virtual BOOL PreTranslateMessage(MSG* pMsg);
    afx_msg void OnTimer(UINT_PTR nIDEvent);
    afx_msg HBRUSH OnCtlColor(CDC* pDC, CWnd* pWnd, UINT nCtlColor);
    afx_msg void OnStnClickedStatic3();
};

```

(2021.6.15) ——版本2.0.1

```

/* 版本:陨石撞飞机2.0.1
 * 时间: 2021.6.15
 * 作者: orall
 * 完善了: 1.增加声音（爆炸声音和开始声音）
 *          2.界面闪烁
 *          3.改变游戏的高度和宽度，增强游戏的可玩性
 * 待完善: 1.增加难度
 *          2.完善界面（开始的界面）
 *          3.得分机制和胜利机制
 */

// planeDlg.cpp: 实现文件
//

#include "pch.h"

```



```

#include "framework.h"
#include "plane.h"
#include "planeDlg.h"
#include "afxdialogex.h"

//game相关的头文件
#include<vector>
#include <iostream>
#include <stdlib.h>
#include <time.h>
#include <conio.h>
#include <windows.h>

//playsound的头文件
#include <mmSystem.h>
#pragma comment(lib,"winmm.lib")

int start = 0;           //判断游戏是否开始

//使图片透明化（参考地址：https://blog.csdn.net/hust\_xy/article/details/9313239）
void TransparentPNG(CImage* png)
{
    for (int i = 0; i < png->GetWidth(); i++)
    {
        for (int j = 0; j < png->GetHeight(); j++)
        {
            unsigned char* pucColor = reinterpret_cast<unsigned char*>(png->GetPixelAddress(i, j));
            pucColor[0] = pucColor[0] * pucColor[3] / 255;
            pucColor[1] = pucColor[1] * pucColor[3] / 255;
            pucColor[2] = pucColor[2] * pucColor[3] / 255;
        }
    }
}

#ifdef _DEBUG
#define new DEBUG_NEW
#endif

// 用于应用程序“关于”菜单项的 CAboutDlg 对话框

class CAboutDlg : public CDialogEx
{
public:
    CAboutDlg();

```

```

// 对话框数据
#ifdef AFX_DESIGN_TIME
    enum { IDD = IDD_ABOUTBOX };
#endif

protected:
    virtual void DoDataExchange(CDataExchange* pDX);    // DDX/DDV 支持

// 实现
protected:
    DECLARE_MESSAGE_MAP()
};

CAboutDlg::CAboutDlg() : CDialogEx(IDD_ABOUTBOX)
{
}

void CAboutDlg::DoDataExchange(CDataExchange* pDX)
{
    CDialogEx::DoDataExchange(pDX);
}

BEGIN_MESSAGE_MAP(CAboutDlg, CDialogEx)
END_MESSAGE_MAP()

// CplaneDlg 对话框

CplaneDlg::CplaneDlg(CWnd* pParent /*=nullptr*/)
    : CDialogEx(IDD_PLANE_DIALOG, pParent)
{
    m_hIcon = AfxGetApp()->LoadIcon(IDR_MAINFRAME);
}

void CplaneDlg::DoDataExchange(CDataExchange* pDX)
{
    CDialogEx::DoDataExchange(pDX);
}

BEGIN_MESSAGE_MAP(CplaneDlg, CDialogEx)
    ON_WM_SYSCOMMAND()
    ON_WM_PAINT()
    ON_WM_QUERYDRAGICON()

```

```

ON_BN_CLICKED(IDC_BUTTON1, &CplaneDlg::OnBnClickedButton1)
ON_BN_CLICKED(IDC_BUTTON3, &CplaneDlg::OnBnClickedButton3)
ON_BN_CLICKED(IDC_BUTTON2, &CplaneDlg::OnBnClickedButton2)
ON_WM_KEYDOWN()
ON_WM_TIMER()
END_MESSAGE_MAP()

// CplaneDlg 消息处理程序

BOOL CplaneDlg::OnInitDialog()
{
    CDialogEx::OnInitDialog();

    // 将“关于...”菜单项添加到系统菜单中。

    // IDM_ABOUTBOX 必须在系统命令范围内。
    ASSERT((IDM_ABOUTBOX & 0xFFF0) == IDM_ABOUTBOX);
    ASSERT(IDM_ABOUTBOX < 0xF000);

    CMenu* pSysMenu = GetSystemMenu(FALSE);
    if (pSysMenu != nullptr)
    {
        BOOL bNameValid;
        CString strAboutMenu;
        bNameValid = strAboutMenu.LoadString(IDS_ABOUTBOX);
        ASSERT(bNameValid);
        if (!strAboutMenu.IsEmpty())
        {
            pSysMenu->AppendMenu(MF_SEPARATOR);
            pSysMenu->AppendMenu(MF_STRING, IDM_ABOUTBOX, strAboutMenu);
        }
    }

    // 设置此对话框的图标。当应用程序主窗口不是对话框时，框架将自动
    // 执行此操作
    SetIcon(m_hIcon, TRUE);        // 设置大图标
    SetIcon(m_hIcon, FALSE);       // 设置小图标

    // TODO: 在此添加额外的初始化代码

    //设置窗口大小(其中SetWindowPos函数的第四和第五个参数分别表示窗口的宽度,和高度,大家可以调整这两个参数
    //来改变自己窗口的大小。)
    //SetWindowPos(NULL, 0, 0, 700, 500, SWP_NOMOVE);

    return TRUE;  // 除非将焦点设置到控件，否则返回 TRUE

```

```

}

void CplaneDlg::OnSysCommand(UINT nID, LPARAM lParam)
{
    if ((nID & 0xFFF0) == IDM_ABOUTBOX)
    {
        CAboutDlg dlgAbout;
        dlgAbout.DoModal();
    }
    else
    {
        CDialogEx::OnSysCommand(nID, lParam);
    }
}

// 如果向对话框添加最小化按钮，则需要下面的代码
// 来绘制该图标。 对于使用文档/视图模型的 MFC 应用程序，
// 这将由框架自动完成。

void CplaneDlg::OnPaint()
{
    if (IsIconic())
    {
        CPaintDC dc(this); // 用于绘制的设备上下文

        SendMessage(WM_ICONERASEBKGND, reinterpret_cast<WPARAM>(dc.GetSafeHdc()), 0);

        // 使图标在工作区矩形中居中
        int cxIcon = GetSystemMetrics(SM_CXICON);
        int cyIcon = GetSystemMetrics(SM_CYICON);
        CRect rect;
        GetClientRect(&rect);
        int x = (rect.Width() - cxIcon + 1) / 2;
        int y = (rect.Height() - cyIcon + 1) / 2;

        // 绘制图标
        dc.DrawIcon(x, y, m_hIcon);
    }
    else
    {
        CDialogEx::OnPaint();
    }
}

```

```

if (start == 1) {
    CDC* cDC = this->GetDC();          //获取窗口DC指针
    GetClientRect(&m_client);

    //创建缓冲DC
    m_cacheDC.CreateCompatibleDC(NULL);
    m_cacheCBitmap.CreateCompatibleBitmap(cDC, m_client.Width(), m_client.Height());
    m_cacheDC.SelectObject(&m_cacheCBitmap);

    //画背景
    background.Draw(m_cacheDC, CRect(0, 0, m_client.Width(), m_client.Height()));
    m_plane.Draw(m_cacheDC, m_planepos); //贴飞机

    for (int i = 0; i < this->max_stonenum; i++) {
        if (this->is_stone[i] == 1) {          //如果石头存在
            m_stone[i].Draw(m_cacheDC, m_stonepos[i]);      //贴石头
        }
    }

    //最后将缓冲DC内容输出到窗口DC中
    cDC->BitBlt(0, 0, m_client.Width(), m_client.Height(), &m_cacheDC, 0, 0, SRCCOPY);

    //—————绘制结束—————

    //在绘制完图后,使窗口区有效
    ValidateRect(&m_client);
    //释放缓冲DC
    m_cacheDC.DeleteDC();
    //释放对象
    m_cacheCBitmap.DeleteObject();
    //释放窗口DC
    ReleaseDC(cDC);
}

```

```

}

```

```

//当用户拖动最小化窗口时系统调用此函数取得光标
//显示。

```

```

HCURSOR CplaneDlg::OnQueryDragIcon()
{
    return static_cast<HCURSOR>(m_hIcon);
}

```

```

void CplaneDlg::OnBnClickedButton1()           //开始游戏
{
    // TODO: 在此添加控件通知处理程序代码
    //隐藏控件
    CWnd* pWnd = GetDlgItem(IDC_STATIC);
    pWnd->ShowWindow(FALSE);
    pWnd = GetDlgItem(IDC_BUTTON1);
    pWnd->ShowWindow(FALSE);
    pWnd = GetDlgItem(IDC_BUTTON2);
    pWnd->ShowWindow(FALSE);
    pWnd = GetDlgItem(IDC_BUTTON3);
    pWnd->ShowWindow(FALSE);

    start = 1;

    //循环播放背景音乐
    PlaySound(_T("res\\background.wav"), NULL, SND_ASYNC | SND_FILENAME | SND_LOOP);

    //设置背景（参考链接：https://blog.csdn.net/qq321772514/article/details/96478790）
    GetClientRect(&m_client);
    //注意这里的位置（把plane.png放到res文件里面）
    background.Load(_T("res\\background.jpg"));
    //这里让图片绘制区域和窗口大小一致
    background.Draw(GetDC()->GetSafeHdc(), CRect(0, 0, m_client.Width(),
m_client.Height()));

    //加载飞机图片
    m_plane.Load(_T("res\\plane.png"));
    TransparentPNG(&m_plane);           //使图片透明化

    //加载石头图片
    for (int i = 0; i < this->max_stonenum; i++) {
        m_stone[i].Load(_T("res\\stone.png"));
        TransparentPNG(&m_stone[i]);    //使图片透明化
    }

    //设置飞机初始位置
    m_planepos.left = m_client.Width() / 2 - 20;           //飞机左边
    m_planepos.right = m_planepos.left + 40;           //飞机的右边等于左边加上飞机的宽度
    m_planepos.top = m_client.Height() - 40;
    m_planepos.bottom = m_planepos.top + 40;
}

```

```

//设置石头初始位置（在顶部随机出现）      【660 430】
srand(unsigned(time(NULL)));
m_stonepos[0].left = rand() % (m_client.Width() - 40);
m_stonepos[0].right = m_stonepos[0].left + 40;
m_stonepos[0].top = 0;
m_stonepos[0].bottom = 40;

is_stone[0] = 1;

SetTimer(1, down_speed, NULL);      //设置定时器1(陨石下落)
SetTimer(2, new_speed, NULL);      //设置定时器2(产生新陨石)

}

void CplaneDlg::OnBnClickedButton3()      //退出
{
    // TODO: 在此添加控件通知处理程序代码
    exit(1);
}

void CplaneDlg::OnBnClickedButton2()      //操作说明
{
    // TODO: 在此添加控件通知处理程序代码
    MessageBox(TEXT("● 操作说明：使用方向键【上下左右】操控飞机来躲避陨石。"), TEXT("★ 提示"),
    MB_OK);
}

//监听键盘按下（参考链接：https://blog.csdn.net/hust\_xy/article/details/9327377）
void CplaneDlg::OnKeyDown(UINT nChar, UINT nRepCnt, UINT nFlags)
{
}

//监听键盘按下（参考链接：https://blog.csdn.net/weixin\_42899088/article/details/101308231）
BOOL CplaneDlg::PreTranslateMessage(MSG* pMsg)
{
    // TODO: 在此添加专用代码和/或调用基类
    // 键盘按键检测
    if (pMsg->message == WM_KEYDOWN) // 按键按下检测

```

```

{
    if (pMsg->wParam == VK_LEFT && m_planepos.left - 10 >= 0 ) {
        m_planepos.left -= 10;
        m_planepos.right -= 10;
    }
    if (pMsg->wParam == VK_RIGHT && m_planepos.right + 10 <= m_client.Width() ) {
        m_planepos.left += 10;
        m_planepos.right += 10;
    }
    if (pMsg->wParam == VK_UP && m_planepos.top - 10 >= 0 ) {
        m_planepos.top -= 10;
        m_planepos.bottom -= 10;
    }
    if (pMsg->wParam == VK_DOWN && m_planepos.bottom + 10 <= m_client.Height() ) {
        m_planepos.top += 10;
        m_planepos.bottom += 10;
    }
    OnPaint();        //更新

}
return CDialogEx::PreTranslateMessage(pMsg);
}

//教程: https://blog.csdn.net/hust\_xy/article/details/9332377
void CplaneDlg::OnTimer(UINT_PTR nIDEvent)
{
    // TODO: 在此添加消息处理程序代码和/或调用默认值
    switch (nIDEvent)
    {
    case 1:
        for (int i = 0; i < max_stonenum; i++) {
            if (is_stone[i] == 1) {
                if (m_stonepos[i].bottom <= m_client.Height() + 100 ) {
                    m_stonepos[i].top += down_speed;
                    m_stonepos[i].bottom += down_speed;
                }
                if (m_stonepos[i].top > m_client.Height() ) {
                    is_stone[i] = 0;
                }
            }

        }

        OnPaint();
        //爆炸
        if (( abs(m_planepos.top - m_stonepos[i].top) < 30 ) && abs(m_stonepos[i].left

```



```

- m_planepos.left) < 35) {
    CRect tmp = m_planepos;

    //让他们隐藏起来
    m_stonepos[i].top = -100;
    m_stonepos[i].bottom = -100 + 40;

    m_planepos.top = -100;
    m_planepos.bottom = -100 + 40;

    OnPaint();

    //记得加头文件而且必须是wav格式（在线MP3转wav: https://onlineconvertfree.com）
    PlaySound(_T("res\\bombmusic.wav"), NULL, SND_ASYNC );//播放爆炸音响

    //加载爆炸图片
    CImage bomb;
    bomb.Load(_T("res\\bomb.png"));
    TransparentPNG(&bomb);          //使图片透明化
    bomb.Draw(GetDC()->GetSafeHdc(), tmp);
    KillTimer(1);
    KillTimer(2);
    Sleep(2000);

    if (MessageBox(TEXT("很遗憾，游戏结束,是否重新开始一局新的游戏? "), TEXT("★ 提示"),
MB_YESNO) == IDYES ) {
        //初始化
        for (int i = 0; i < max_stonenum; i++) {
            is_stone[i] = 0;
        }

        //重新开始一局游戏

        //设置飞机初始位置
        m_planepos.left = m_client.Width()/2 - 20;          //飞机左边
        m_planepos.right = m_planepos.left + 40;          //飞机的右边等于左边加上飞机的宽
度

        m_planepos.top = m_client.Height()-40;
        m_planepos.bottom = m_planepos.top + 40;

        //设置石头初始位置（在顶部随机出现）          【660 430】
        srand(unsigned(time(NULL)));
        m_stonepos[0].left = rand() % (m_client.Width()-40);
        m_stonepos[0].right = m_stonepos[0].left + 40;
        m_stonepos[0].top = 0;

```

```

        m_stonepos[0].bottom = 40;

        is_stone[0] = 1;

        //循环播放背景音乐
        PlaySound(_T("res\\background.wav"), NULL, SND_ASYNC | SND_FILENAME |
SND_LOOP);

        SetTimer(1, down_speed, NULL);          //设置定时器1(陨石下落)
        SetTimer(2, new_speed, NULL);           //设置定时器2(产生新陨石)

    }
    else {
        exit(1);
    }
    ;

}

}

break;

case 2:
{
    for (int i = 0; i < max_stonenum; i++) {
        if (this->is_stone[i] == 0) {           //没有石头的话
            is_stone[i] = 1;                    //标记为有石头
            //srand(unsigned(time(NULL)));      不能调用两次，否则不会随机
            m_stonepos[i].left = rand()%(m_client.Width() - 40);
            m_stonepos[i].right = m_stonepos[i].left + 40;
            m_stonepos[i].top = 0;
            m_stonepos[i].bottom = 40;
            break;
        }
    }
}

break;

default:
    break;
}

```

```
    CDialogEx::OnTimer(nIDEvent);  
}
```

(2021.6.15)——版本2.0

```
/* 版本:陨石撞飞机2.0  
* 时间: 2021.6.15  
* 作者: orall  
* 完善了: 1.增加声音(爆炸声音和开始声音)  
*          2.界面闪烁  
* 待完善: 1.增加难度  
*          2.完善界面(开始的界面)  
*          3.得分机制和胜利机制  
*          4.游戏的可玩性  
*/  
  
// planeDlg.cpp: 实现文件  
//  
  
#include "pch.h"  
#include "framework.h"  
#include "plane.h"  
#include "planeDlg.h"  
#include "afxdialogex.h"  
  
//game相关的头文件  
#include<vector>  
#include <iostream>  
#include <stdlib.h>  
#include <time.h>  
#include <conio.h>  
#include <windows.h>  
  
//playsound的头文件  
#include <mmSystem.h>  
#pragma comment(lib,"winmm.lib")  
  
int start = 0;           //判断游戏是否开始  
  
//使图片透明化(参考地址: https://blog.csdn.net/hust\_xy/article/details/9313239)  
void TransparentPNG(CImage* png)  
{  
    for (int i = 0; i < png->GetWidth(); i++)  
    {  
        for (int j = 0; j < png->GetHeight(); j++)
```

```

    {
        unsigned char* pucColor = reinterpret_cast<unsigned char*>(png->GetPixelAddress(i,
j));

        pucColor[0] = pucColor[0] * pucColor[3] / 255;
        pucColor[1] = pucColor[1] * pucColor[3] / 255;
        pucColor[2] = pucColor[2] * pucColor[3] / 255;
    }
}
}

#ifdef _DEBUG
#define new DEBUG_NEW
#endif

// 用于应用程序“关于”菜单项的 CAboutDlg 对话框

class CAboutDlg : public CDialogEx
{
public:
    CAboutDlg();

// 对话框数据
#ifdef AFX_DESIGN_TIME
    enum { IDD = IDD_ABOUTBOX };
#endif

protected:
    virtual void DoDataExchange(CDataExchange* pDX);    // DDX/DDV 支持

// 实现
protected:
    DECLARE_MESSAGE_MAP()
};

CAboutDlg::CAboutDlg() : CDialogEx(IDD_ABOUTBOX)
{
}

void CAboutDlg::DoDataExchange(CDataExchange* pDX)
{
    CDialogEx::DoDataExchange(pDX);
}

BEGIN_MESSAGE_MAP(CAboutDlg, CDialogEx)
END_MESSAGE_MAP()

```

```

// CplaneDlg 对话框

CplaneDlg::CplaneDlg(CWnd* pParent /*=nullptr*/)
    : CDialogEx(IDD_PLANE_DIALOG, pParent)
{
    m_hIcon = AfxGetApp()->LoadIcon(IDR_MAINFRAME);
}

void CplaneDlg::DoDataExchange(CDataExchange* pDX)
{
    CDialogEx::DoDataExchange(pDX);
}

BEGIN_MESSAGE_MAP(CplaneDlg, CDialogEx)
    ON_WM_SYSCOMMAND()
    ON_WM_PAINT()
    ON_WM_QUERYDRAGICON()
    ON_BN_CLICKED(IDC_BUTTON1, &CplaneDlg::OnBnClickedButton1)
    ON_BN_CLICKED(IDC_BUTTON3, &CplaneDlg::OnBnClickedButton3)
    ON_BN_CLICKED(IDC_BUTTON2, &CplaneDlg::OnBnClickedButton2)
    ON_WM_KEYDOWN()
    ON_WM_TIMER()
END_MESSAGE_MAP()

// CplaneDlg 消息处理程序

BOOL CplaneDlg::OnInitDialog()
{
    CDialogEx::OnInitDialog();

    // 将“关于...”菜单项添加到系统菜单中。

    // IDM_ABOUTBOX 必须在系统命令范围内。
    ASSERT((IDM_ABOUTBOX & 0xFFF0) == IDM_ABOUTBOX);
    ASSERT(IDM_ABOUTBOX < 0xF000);

    CMenu* pSysMenu = GetSystemMenu(FALSE);
    if (pSysMenu != nullptr)
    {
        BOOL bNameValid;
        CString strAboutMenu;

```

```

        bNameValid = strAboutMenu.LoadString(IDS_ABOUTBOX);
        ASSERT(bNameValid);
        if (!strAboutMenu.IsEmpty())
        {
            pSysMenu->AppendMenu(MF_SEPARATOR);
            pSysMenu->AppendMenu(MF_STRING, IDM_ABOUTBOX, strAboutMenu);
        }
    }

    // 设置此对话框的图标。 当应用程序主窗口不是对话框时，框架将自动
    // 执行此操作
    SetIcon(m_hIcon, TRUE);          // 设置大图标
    SetIcon(m_hIcon, FALSE);         // 设置小图标

    // TODO: 在此添加额外的初始化代码

    //设置窗口大小( 其中SetWindowPos函数的第四和第五个参数分别表示窗口的宽度,和高度,大家可以调整这两个参数
    //来改变自己窗口的大小。)
    SetWindowPos(NULL, 0, 0, 700, 500, SWP_NOMOVE);

    return TRUE;  // 除非将焦点设置到控件，否则返回 TRUE
}

void CplaneDlg::OnSysCommand(UINT nID, LPARAM lParam)
{
    if ((nID & 0xFFF0) == IDM_ABOUTBOX)
    {
        CAboutDlg dlgAbout;
        dlgAbout.DoModal();
    }
    else
    {
        CDialogEx::OnSysCommand(nID, lParam);
    }
}

// 如果向对话框添加最小化按钮，则需要下面的代码
// 来绘制该图标。 对于使用文档/视图模型的 MFC 应用程序，
// 这将由框架自动完成。

void CplaneDlg::OnPaint()
{
    if (IsIconic())
    {
        CPaintDC dc(this); // 用于绘制的设备上下文
    }
}

```

```

SendMessage(WM_ICONERASEBKGND, reinterpret_cast<WPARAM>(dc.GetSafeHdc()), 0);

// 使图标在工作区矩形中居中
int cxIcon = GetSystemMetrics(SM_CXICON);
int cyIcon = GetSystemMetrics(SM_CYICON);
CRect rect;
GetClientRect(&rect);
int x = (rect.Width() - cxIcon + 1) / 2;
int y = (rect.Height() - cyIcon + 1) / 2;

// 绘制图标
dc.DrawIcon(x, y, m_hIcon);
}
else
{
    CDialogEx::OnPaint();
}

//解决界面闪烁问题(参考链接: https://blog.csdn.net/hust\_xy/article/details/9334121)

if (start == 1) {
    CDC* cDC = this->GetDC();           //获取窗口DC指针
    CRect m_client;
    GetClientRect(&m_client);

    //创建缓冲DC
    m_cacheDC.CreateCompatibleDC(NULL);
    m_cacheCBitmap.CreateCompatibleBitmap(cDC, m_client.Width(), m_client.Height());
    m_cacheDC.SelectObject(&m_cacheCBitmap);

    //画背景
    CRect rc;
    GetWindowRect(&rc);
    background.Draw(m_cacheDC, CRect(0, 0, rc.Width(), rc.Height()));
    m_plane.Draw(m_cacheDC, m_planepos); //贴飞机

    for (int i = 0; i < this->max_stonenum; i++) {
        if (this->is_stone[i] == 1) {           //如果石头存在
            m_stone[i].Draw(m_cacheDC, m_stonepos[i]); //贴石头
        }
    }
    //ReleaseDC(cDC);           //释放DC
    //最后将缓冲DC内容输出到窗口DC中
    cDC->BitBlt(0, 0, m_client.Width(), m_client.Height(), &m_cacheDC, 0, 0, SRCCOPY);
}

```

```

//—————绘制结束—————

//在绘制完图后,使窗口区有效
ValidateRect(&m_client);
//释放缓冲DC
m_cacheDC.DeleteDC();
//释放对象
m_cacheCBitmap.DeleteObject();
//释放窗口DC
ReleaseDC(cDC);
}

}

//当用户拖动最小化窗口时系统调用此函数取得光标
//显示。
HCURSOR CplaneDlg::OnQueryDragIcon()
{
    return static_cast<HCURSOR>(m_hIcon);
}

void CplaneDlg::OnBnClickedButton1() //开始游戏
{
    // TODO: 在此添加控件通知处理程序代码
    //隐藏控件
    CWnd* pWnd = GetDlgItem(IDC_STATIC);
    pWnd->ShowWindow(FALSE);
    pWnd = GetDlgItem(IDC_BUTTON1);
    pWnd->ShowWindow(FALSE);
    pWnd = GetDlgItem(IDC_BUTTON2);
    pWnd->ShowWindow(FALSE);
    pWnd = GetDlgItem(IDC_BUTTON3);
    pWnd->ShowWindow(FALSE);

    start = 1;

    //循环播放背景音乐
    PlaySound(_T("res\\background.wav"), NULL, SND_ASYNC | SND_FILENAME | SND_LOOP);

```



```

//设置背景（参考链接：https://blog.csdn.net/qq321772514/article/details/96478790）
CRect rc;
GetWindowRect(&rc);
//注意这里的位置（把plane.png放到res文件里面）
background.Load(_T("res\\background.jpg"));
//这里让图片绘制区域和窗口大小一致
background.Draw(GetDC()->GetSafeHdc(), CRect(0, 0, rc.Width(), rc.Height()));

//加载飞机图片
m_plane.Load(_T("res\\plane.png"));
TransparentPNG(&m_plane);           //使图片透明化

//加载石头图片
for (int i = 0; i < this->max_stonenum; i++) {
    m_stone[i].Load(_T("res\\stone.png"));
    TransparentPNG(&m_stone[i]);    //使图片透明化
}

//设置飞机初始位置
m_planepos.left = 700 / 2 - 40;      //飞机左边
m_planepos.right = m_planepos.left + 40; //飞机的右边等于左边加上飞机的宽度
m_planepos.top = 500 - 90 + 10;      //所以高度是420
m_planepos.bottom = m_planepos.top + 40;

//设置石头初始位置（在顶部随机出现）    【660 430】
srand(unsigned(time(NULL)));
m_stonepos[0].left = rand() % 660;
m_stonepos[0].right = m_stonepos[0].left + 40;
m_stonepos[0].top = 0;
m_stonepos[0].bottom = 40;

is_stone[0] = 1;

SetTimer(1, down_speed, NULL);        //设置定时器1(陨石下落)
SetTimer(2, new_speed, NULL);          //设置定时器2(产生新陨石)

}

```

```

void CplaneDlg::OnBnClickedButton3()           //退出
{
    // TODO: 在此添加控件通知处理程序代码
    exit(1);
}

void CplaneDlg::OnBnClickedButton2()           //操作说明
{
    // TODO: 在此添加控件通知处理程序代码
    MessageBox(TEXT("● 操作说明：使用方向键【上下左右】操控飞机来躲避陨石。"), TEXT("★ 提示"),
    MB_OK);
}

//监听键盘按下（参考链接：https://blog.csdn.net/hust\_xy/article/details/9327377）
void CplaneDlg::OnKeyDown(UINT nChar, UINT nRepCnt, UINT nFlags)
{
}

//监听键盘按下（参考链接：https://blog.csdn.net/weixin\_42899088/article/details/101308231）
BOOL CplaneDlg::PreTranslateMessage(MSG* pMsg)
{
    // TODO: 在此添加专用代码和/或调用基类
    // 键盘按键检测
    if (pMsg->message == WM_KEYDOWN) // 按键按下检测
    {
        if (pMsg->wParam == VK_LEFT && m_planepos.left >= 10 ) {
            m_planepos.left -= 10;
            m_planepos.right -= 10;
        }
        else if (pMsg->wParam == VK_RIGHT && m_planepos.right <= 670) {
            m_planepos.left += 10;
            m_planepos.right += 10;
        }
        else if (pMsg->wParam == VK_UP && m_planepos.top >= 10) {
            m_planepos.top -= 10;
            m_planepos.bottom -= 10;
        }
        else if (pMsg->wParam == VK_DOWN && m_planepos.bottom <= 440) {
            m_planepos.top += 10;
            m_planepos.bottom += 10;
        }
        OnPaint();           //更新
        /*
        CDC* cDC = this->GetDC();           //获取窗口DC指针
        if (start == 1) {

```

```

        //重新画背景
        CRect rc;
        GetWindowRect(&rc);
        background.Draw(GetDC()->GetSafeHdc(), CRect(0, 0, rc.Width(), rc.Height()));
        m_plane.Draw(*cDC, m_planepos); //贴飞机
    }
    ReleaseDC(cDC);          //释放DC
    */
}
return CDialogEx::PreTranslateMessage(pMsg);
}

//教程: https://blog.csdn.net/hust\_xy/article/details/9332377
void CplaneDlg::OnTimer(UINT_PTR nIDEvent)
{
    // TODO: 在此添加消息处理程序代码和/或调用默认值
    switch (nIDEvent)
    {
    case 1:
        for (int i = 0; i < max_stonenum; i++) {
            if (is_stone[i] == 1) {
                if (m_stonepos[i].bottom <= 480) {
                    m_stonepos[i].top += this->down_speed;
                    m_stonepos[i].bottom += this->down_speed;
                }
                if (m_stonepos[i].bottom > 470) {
                    is_stone[i] = 0;
                }
            }
        }

        OnPaint();
        //爆炸
        if ((abs(m_planepos.top - m_stonepos[i].top) < 30 ) && abs(m_stonepos[i].left
- m_planepos.left) < 35) {
            CRect tmp = m_planepos;

            //让他们隐藏起来
            m_stonepos[i].top = -100;
            m_stonepos[i].bottom = -100 + 40;

            m_planepos.top = -100;
            m_planepos.bottom = -100 + 40;
        }
    }
}

```

```

OnPaint();

//记得加头文件而且必须是wav格式（在线MP3转wav: https://onlineconvertfree.com）
PlaySound(_T("res\\bombmusic.wav"), NULL, SND_ASYNC );//播放爆炸音响

//加载爆炸图片
CImage bomb;
bomb.Load(_T("res\\bomb.png"));
TransparentPNG(&bomb);          //使图片透明化
bomb.Draw(GetDC()->GetSafeHdc(), tmp);
KillTimer(1);
KillTimer(2);
Sleep(2000);

if (MessageBox(TEXT("很遗憾，游戏结束,是否重新开始一局新的游戏? "), TEXT("★ 提示"),
MB_YESNO) == IDYES ) {
    //初始化
    for (int i = 0; i < max_stonenum; i++) {
        is_stone[i] = 0;
    }

    //重新开始一局游戏

    //设置飞机初始位置
    m_planepos.left = 700 / 2 - 40;          //飞机左边
    m_planepos.right = m_planepos.left + 40; //飞机的右边等于左边加上飞机的宽
度

    m_planepos.top = 500 - 90 + 10;          //所以高度是420
    m_planepos.bottom = m_planepos.top + 40;

    //设置石头初始位置（在顶部随机出现）    【660 430】
    srand(unsigned(time(NULL)));
    m_stonepos[0].left = rand() % 660;
    m_stonepos[0].right = m_stonepos[0].left + 40;
    m_stonepos[0].top = 0;
    m_stonepos[0].bottom = 40;

    this->is_stone[0] = 1;

    //循环播放背景音乐
    PlaySound(_T("res\\background.wav"), NULL, SND_ASYNC | SND_FILENAME |
SND_LOOP);

    SetTimer(1, this->down_speed, NULL);      //设置定时器1(陨石下落)
    SetTimer(2, this->new_speed, NULL);       //设置定时器2(产生新陨石)

```

```

        }
        else {
            exit(1);
        }
        ;
    }

}

break;

case 2:
{
    for (int i = 0; i < this->max_stonenum; i++) {
        if (this->is_stone[i] == 0) {           //没有石头的话
            is_stone[i] = 1;                   //标记为有石头
            //srand(unsigned(time(NULL)));      不能调用两次，否则不会随机
            m_stonepos[i].left = rand() % 660;
            m_stonepos[i].right = m_stonepos[i].left + 40;
            m_stonepos[i].top = 0;
            m_stonepos[i].bottom = 40;
            break;
        }
    }
}

break;

default:
    break;
}

CDialogEx::OnTimer(nIDEvent);
}

```

```

// planeDlg.h: 头文件
//

```

```

#pragma once

```

```

#define MAX 30                                //石头出现的最大数量

// CplaneDlg 对话框
class CplaneDlg : public CDialogEx
{
// 构造
public:
    CplaneDlg(CWnd* pParent = nullptr); // 标准构造函数

    //添加的数据
    CImage background;
    CImage m_plane;
    CImage m_stone[MAX];
    CRect m_planepos;
    CRect m_stonepos[MAX];

    int max_stonenum = MAX;                    //最大的石头数量（最多出现10个）
    int is_stone[MAX] = {0};                  //石头是否存在
    int down_speed = 10;                      //石头下落的速度
    int new_speed = 500;                      //石头产生的速度

    //解决图片闪烁
    CDC m_cacheDC;    //缓冲DC
    CBitmap m_cacheCBitmap; //缓冲位图

// 对话框数据
#ifdef AFX_DESIGN_TIME
    enum { IDD = IDD_PLANE_DIALOG };
#endif

protected:
    virtual void DoDataExchange(CDataExchange* pDX);    // DDX/DDV 支持

// 实现
protected:
    HICON m_hIcon;

    // 生成的消息映射函数
    virtual BOOL OnInitDialog();
    afx_msg void OnSysCommand(UINT nID, LPARAM lParam);
    afx_msg void OnPaint();
    afx_msg HCURSOR OnQueryDragIcon();

```

```

    DECLARE_MESSAGE_MAP()
public:
    afx_msg void OnBnClickedButton1();
    afx_msg void OnBnClickedButton3();
    afx_msg void OnBnClickedButton2();
    afx_msg void OnKeyDown(UINT nChar, UINT nRepCnt, UINT nFlags);
    virtual BOOL PreTranslateMessage(MSG* pMsg);
    afx_msg void OnTimer(UINT_PTR nIDEvent);
};

```

(2021.6.15) ——版本1.0

```

/* 版本:陨石撞飞机1.0
 * 时间: 2021.6.15
 * 作者: orall
 * 版本特点: 实现了基本功能
 * 待完善: 1.增加难度
 *           2.增加声音(爆炸声音和开始声音)
 *           3.完善界面(开始的界面)
 *           4.得分机制和胜利机制
 *           5.界面闪烁
 */

// planeDlg.cpp: 实现文件
//

#include "pch.h"
#include "framework.h"
#include "plane.h"
#include "planeDlg.h"
#include "afxdialogex.h"

//game相关的头文件
#include<vector>
#include <iostream>
#include <stdlib.h>
#include <time.h>
#include <conio.h>
#include <windows.h>

int start = 0;                //判断游戏是否开始

//使图片透明化(参考地址: https://blog.csdn.net/hust\_xy/article/details/9313239)
void TransparentPNG(CImage* png)

```

```

{
for (int i = 0; i < png->GetWidth(); i++)
{
    for (int j = 0; j < png->GetHeight(); j++)
    {
        unsigned char* pucColor = reinterpret_cast<unsigned char*>(png->GetPixelAddress(i,
j));

        pucColor[0] = pucColor[0] * pucColor[3] / 255;
        pucColor[1] = pucColor[1] * pucColor[3] / 255;
        pucColor[2] = pucColor[2] * pucColor[3] / 255;
    }
}
}

#ifdef _DEBUG
#define new DEBUG_NEW
#endif

// 用于应用程序“关于”菜单项的 CAboutDlg 对话框

class CAboutDlg : public CDialogEx
{
public:
    CAboutDlg();

// 对话框数据
#ifdef AFX_DESIGN_TIME
    enum { IDD = IDD_ABOUTBOX };
#endif

protected:
    virtual void DoDataExchange(CDataExchange* pDX);    // DDX/DDV 支持

// 实现
protected:
    DECLARE_MESSAGE_MAP()
};

CAboutDlg::CAboutDlg() : CDialogEx(IDD_ABOUTBOX)
{
}

void CAboutDlg::DoDataExchange(CDataExchange* pDX)
{
    CDialogEx::DoDataExchange(pDX);
}

```



```

}

BEGIN_MESSAGE_MAP(CAboutDlg, CDialogEx)
END_MESSAGE_MAP()

// CplaneDlg 对话框

CplaneDlg::CplaneDlg(CWnd* pParent /*=nullptr*/)
    : CDialogEx(IDD_PLANE_DIALOG, pParent)
{
    m_hIcon = AfxGetApp()->LoadIcon(IDR_MAINFRAME);
}

void CplaneDlg::DoDataExchange(CDataExchange* pDX)
{
    CDialogEx::DoDataExchange(pDX);
}

BEGIN_MESSAGE_MAP(CplaneDlg, CDialogEx)
    ON_WM_SYSCOMMAND()
    ON_WM_PAINT()
    ON_WM_QUERYDRAGICON()
    ON_BN_CLICKED(IDC_BUTTON1, &CplaneDlg::OnBnClickedButton1)
    ON_BN_CLICKED(IDC_BUTTON3, &CplaneDlg::OnBnClickedButton3)
    ON_BN_CLICKED(IDC_BUTTON2, &CplaneDlg::OnBnClickedButton2)
    ON_WM_KEYDOWN()
    ON_WM_TIMER()
END_MESSAGE_MAP()

// CplaneDlg 消息处理程序

BOOL CplaneDlg::OnInitDialog()
{
    CDialogEx::OnInitDialog();

    // 将“关于...”菜单项添加到系统菜单中。

    // IDM_ABOUTBOX 必须在系统命令范围内。
    ASSERT((IDM_ABOUTBOX & 0xFFF0) == IDM_ABOUTBOX);
    ASSERT(IDM_ABOUTBOX < 0xF000);

    CMenu* pSysMenu = GetSystemMenu(FALSE);

```

```

    if (pSysMenu != nullptr)
    {
        BOOL bNameValid;
        CString strAboutMenu;
        bNameValid = strAboutMenu.LoadString(IDS_ABOUTBOX);
        ASSERT(bNameValid);
        if (!strAboutMenu.IsEmpty())
        {
            pSysMenu->AppendMenu(MF_SEPARATOR);
            pSysMenu->AppendMenu(MF_STRING, IDM_ABOUTBOX, strAboutMenu);
        }
    }

    // 设置此对话框的图标。 当应用程序主窗口不是对话框时，框架将自动
    // 执行此操作
    SetIcon(m_hIcon, TRUE);          // 设置大图标
    SetIcon(m_hIcon, FALSE);         // 设置小图标

    // TODO: 在此添加额外的初始化代码

    //设置窗口大小( 其中SetWindowPos函数的第四和第五个参数分别表示窗口的宽度,和高度,大家可以调整这两个参数
    //来改变自己窗口的大小。)
    SetWindowPos(NULL, 0, 0, 700, 500, SWP_NOMOVE);

    return TRUE;  // 除非将焦点设置到控件，否则返回 TRUE
}

void CplaneDlg::OnSysCommand(UINT nID, LPARAM lParam)
{
    if ((nID & 0xFFF0) == IDM_ABOUTBOX)
    {
        CAboutDlg dlgAbout;
        dlgAbout.DoModal();
    }
    else
    {
        CDialogEx::OnSysCommand(nID, lParam);
    }
}

// 如果向对话框添加最小化按钮，则需要下面的代码
// 来绘制该图标。 对于使用文档/视图模型的 MFC 应用程序，
// 这将由框架自动完成。

void CplaneDlg::OnPaint()
{

```

```

if (IsIconic())
{
    CPaintDC dc(this); // 用于绘制的设备上下文

    SendMessage(WM_ICONERASEBKGND, reinterpret_cast<WPARAM>(dc.GetSafeHdc()), 0);

    // 使图标在工作区矩形中居中
    int cxIcon = GetSystemMetrics(SM_CXICON);
    int cyIcon = GetSystemMetrics(SM_CYICON);
    CRect rect;
    GetClientRect(&rect);
    int x = (rect.Width() - cxIcon + 1) / 2;
    int y = (rect.Height() - cyIcon + 1) / 2;

    // 绘制图标
    dc.DrawIcon(x, y, m_hIcon);
}
else
{
    CDialogEx::OnPaint();
}

CDC* cDC = this->GetDC(); //获取窗口DC指针
if (start == 1) {
    //重新画背景
    CRect rc;
    GetWindowRect(&rc);
    background.Draw(GetDC()->GetSafeHdc(), CRect(0, 0, rc.Width(), rc.Height()));
    m_plane.Draw(*cDC, m_planepos); //贴飞机

    for (int i = 0; i < this->max_stonenum; i++) {
        if (this->is_stone[i] == 1) { //如果石头存在
            m_stone[i].Draw(*cDC, m_stonepos[i]); //贴石头
        }
    }
}
ReleaseDC(cDC); //释放DC
}

//当用户拖动最小化窗口时系统调用此函数取得光标
//显示。
HCURSOR CplaneDlg::OnQueryDragIcon()
{
    return static_cast<HCURSOR>(m_hIcon);
}

```

```

void CplaneDlg::OnBnClickedButton1()           //开始游戏
{
    // TODO: 在此添加控件通知处理程序代码
    //隐藏控件
    CWnd* pWnd = GetDlgItem(IDC_STATIC);
    pWnd->ShowWindow(FALSE);
    pWnd = GetDlgItem(IDC_BUTTON1);
    pWnd->ShowWindow(FALSE);
    pWnd = GetDlgItem(IDC_BUTTON2);
    pWnd->ShowWindow(FALSE);
    pWnd = GetDlgItem(IDC_BUTTON3);
    pWnd->ShowWindow(FALSE);

    start = 1;

    //设置背景（参考链接：https://blog.csdn.net/qq321772514/article/details/96478790）
    CRect rc;
    GetWindowRect(&rc);
    //注意这里的位置（把plane.png放到res文件里面）
    background.Load(_T("res\\background.jpg"));
    //这里让图片绘制区域和窗口大小一致
    background.Draw(GetDC()->GetSafeHdc(), CRect(0, 0, rc.Width(), rc.Height()));

    //加载飞机图片
    m_plane.Load(_T("res\\plane.png"));
    TransparentPNG(&m_plane);           //使图片透明化

    //加载石头图片
    for (int i = 0; i < this->max_stonenum; i++) {
        m_stone[i].Load(_T("res\\stone.png"));
        TransparentPNG(&m_stone[i]);    //使图片透明化
    }

    //设置飞机初始位置
    m_planepos.left = 700 / 2 - 40;           //飞机左边
    m_planepos.right = m_planepos.left + 40;    //飞机的右边等于左边加上飞机的宽度
    m_planepos.top = 500 - 90 + 10 ;           //所以高度是420
    m_planepos.bottom = m_planepos.top + 40;

    //设置石头初始位置（在顶部随机出现）      【660 430】
    srand(unsigned(time(NULL)));
    m_stonepos[0].left = rand() % 660;

```

```

        m_stonepos[0].right = m_stonepos[0].left + 40;
        m_stonepos[0].top = 0;
        m_stonepos[0].bottom = 40;

        this->is_stone[0] = 1;

        SetTimer(1, this->down_speed, NULL);           //设置定时器1(陨石下落)
        SetTimer(2, this->new_speed, NULL);             //设置定时器2(产生新陨石)

    }

void CplaneDlg::OnBnClickedButton3()                  //退出
{
    // TODO: 在此添加控件通知处理程序代码
    exit(1);
}

void CplaneDlg::OnBnClickedButton2()                  //操作说明
{
    // TODO: 在此添加控件通知处理程序代码
    MessageBox(TEXT("● 操作说明：使用方向键【上下左右】操控飞机来躲避陨石。"), TEXT("★ 提示"),
    MB_OK);
}

//监听键盘按下（参考链接：https://blog.csdn.net/hust\_xy/article/details/9327377）
void CplaneDlg::OnKeyDown(UINT nChar, UINT nRepCnt, UINT nFlags)
{
}

//监听键盘按下（参考链接：https://blog.csdn.net/weixin\_42899088/article/details/101308231）
BOOL CplaneDlg::PreTranslateMessage(MSG* pMsg)
{
    // TODO: 在此添加专用代码和/或调用基类
    // 键盘按键检测
    if (pMsg->message == WM_KEYDOWN) // 按键按下检测
    {
        if (pMsg->wParam == VK_LEFT && m_planepos.left >= 10 ) {
            m_planepos.left -= 10;
            m_planepos.right -= 10;
        }
    }
}

```

```

        else if (pMsg->wParam == VK_RIGHT && m_planepos.right <= 670) {
            m_planepos.left += 10;
            m_planepos.right += 10;
        }
        else if (pMsg->wParam == VK_UP && m_planepos.top >= 10) {
            m_planepos.top -= 10;
            m_planepos.bottom -= 10;
        }
        else if (pMsg->wParam == VK_DOWN && m_planepos.bottom <= 440) {
            m_planepos.top += 10;
            m_planepos.bottom += 10;
        }
        OnPaint();          //更新
        /*
        CDC* cDC = this->GetDC();          //获取窗口DC指针
        if (start == 1) {
            //重新画背景
            CRect rc;
            GetWindowRect(&rc);
            background.Draw(GetDC()->GetSafeHdc(), CRect(0, 0, rc.Width(), rc.Height()));
            m_plane.Draw(*cDC, m_planepos); //贴飞机
        }
        ReleaseDC(cDC);          //释放DC
        */
    }
    return CDialogEx::PreTranslateMessage(pMsg);
}

```

//教程: [https://blog.csdn.net/hust\\_xy/article/details/9332377](https://blog.csdn.net/hust_xy/article/details/9332377)

```

void CplaneDlg::OnTimer(UINT_PTR nIDEvent)
{
    // TODO: 在此添加消息处理程序代码和/或调用默认值

    CImage tmp1;
    switch (nIDEvent)
    {
    case 1:
        for (int i = 0; i < this->max_stonenum; i++) {
            if (this->is_stone[i] == 1) {
                if (m_stonepos[i].bottom <= 480) {
                    m_stonepos[i].top += this->down_speed;
                    m_stonepos[i].bottom += this->down_speed;
                }
                if (m_stonepos[i].bottom > 470) {
                    this->is_stone[i] = 0;
                }
            }
        }
    }
}

```

```

    }

    OnPaint();
    //爆炸
    if (m_stonepos[i].bottom >= m_planepos.top && m_stonepos[i].bottom <=
m_planepos.bottom && abs(m_stonepos[i].left - m_planepos.left) < 40) {
        CRect tmp = m_planepos;

        //让他们隐藏起来
        m_stonepos[i].top = -100;
        m_stonepos[i].bottom = -100 + 40;

        m_planepos.top = -100;
        m_planepos.bottom = -100 + 40;

        OnPaint();

        //加载爆炸图片
        CImage bomb;
        bomb.Load(_T("res\\bomb.png"));
        TransparentPNG(&bomb);        //使图片透明化
        bomb.Draw(GetDC()->GetSafeHdc(), tmp);
        KillTimer(1);
        KillTimer(2);
        Sleep(2000);

        if (MessageBox(TEXT("很遗憾，游戏结束,是否重新开始一局新的游戏? "), TEXT("★ 提示"),
MB_YESNO) == IDYES ) {
            //初始化
            for (int i = 0; i < this->max_stonenum; i++) {
                is_stone[i] = 0;
            }

            //重新开始一局游戏

            //设置飞机初始位置
            m_planepos.left = 700 / 2 - 40;                //飞机左边
            m_planepos.right = m_planepos.left + 40;        //飞机的右边等于左边加上飞机的宽
度

            m_planepos.top = 500 - 90 + 10;                //所以高度是420
            m_planepos.bottom = m_planepos.top + 40;

            //设置石头初始位置（在顶部随机出现）

```

```

        srand(unsigned(time(NULL)));
        m_stonepos[0].left = rand() % 660;
        m_stonepos[0].right = m_stonepos[0].left + 40;
        m_stonepos[0].top = 0;
        m_stonepos[0].bottom = 40;

        this->is_stone[0] = 1;

        SetTimer(1, this->down_speed, NULL);           //设置定时器1(陨石下落)
        SetTimer(2, this->new_speed, NULL);             //设置定时器2(产生新陨石)

        //重新开始一局新游戏
        //OnBnClickedButton1();
    }
    else {
        exit(1);
    }
    ;

}

}

break;

case 2:
{
    for (int i = 0; i < this->max_stonenum; i++) {
        if (this->is_stone[i] == 0) {                    //没有石头的话
            is_stone[i] = 1;                            //标记为有石头
            //srand(unsigned(time(NULL)));              不能调用两次，否则不会随机
            m_stonepos[i].left = rand() % 660;
            m_stonepos[i].right = m_stonepos[i].left + 40;
            m_stonepos[i].top = 0;
            m_stonepos[i].bottom = 40;
            break;
        }
    }
}

break;

default:
    break;
}

```



```
    CDialogEx::OnTimer(nIDEvent);  
}
```

```
// planeDlg.h: 头文件  
//  
  
#pragma once  
  
#define MAX 30  
  
// CplaneDlg 对话框  
class CplaneDlg : public CDialogEx  
{  
// 构造  
public:  
    CplaneDlg(CWnd* pParent = nullptr); // 标准构造函数  
  
    //添加的数据  
    CImage background;  
    CImage m_plane;  
    CImage m_stone[MAX];  
    CRect m_planepos;  
    CRect m_stonepos[MAX];  
    int max_stonenum = MAX;           //最大的石头数量（最多出现10个）  
    int is_stone[MAX] = {0};         //石头是否存在  
    int down_speed = 10;             //石头下落的速度  
    int new_speed = 500;             //石头产生的速度  
  
// 对话框数据  
#ifdef AFX_DESIGN_TIME  
    enum { IDD = IDD_PLANE_DIALOG };  
#endif  
  
    protected:  
        virtual void DoDataExchange(CDataExchange* pDX);    // DDX/DDV 支持  
  
// 实现  
protected:
```

```

        HICON m_hIcon;

        // 生成的消息映射函数
        virtual BOOL OnInitDialog();
        afx_msg void OnSysCommand(UINT nID, LPARAM lParam);
        afx_msg void OnPaint();
        afx_msg HCURSOR OnQueryDragIcon();
        DECLARE_MESSAGE_MAP()
public:
        afx_msg void OnBnClickedButton1();
        afx_msg void OnBnClickedButton3();
        afx_msg void OnBnClickedButton2();
        afx_msg void OnKeyDown(UINT nChar, UINT nRepCnt, UINT nFlags);
        virtual BOOL PreTranslateMessage(MSG* pMsg);
        afx_msg void OnTimer(UINT_PTR nIDEvent);
};

```

(2021.6.15)——雏形3.0

```

// planeDlg.cpp: 实现文件
//

#include "pch.h"
#include "framework.h"
#include "plane.h"
#include "planeDlg.h"
#include "afxdialogex.h"

//game相关的头文件
#include<vector>
#include <iostream>
#include <stdlib.h>
#include <time.h>
#include <conio.h>
#include <windows.h>

int start = 0;           //判断游戏是否开始

//使图片透明化（参考地址：https://blog.csdn.net/hust\_xy/article/details/9313239）
void TransparentPNG(CImage* png)
{
    for (int i = 0; i < png->GetWidth(); i++)
    {
        for (int j = 0; j < png->GetHeight(); j++)

```

```

    {
        unsigned char* pucColor = reinterpret_cast<unsigned char*>(png->GetPixelAddress(i,
j));

        pucColor[0] = pucColor[0] * pucColor[3] / 255;
        pucColor[1] = pucColor[1] * pucColor[3] / 255;
        pucColor[2] = pucColor[2] * pucColor[3] / 255;
    }
}
}

#ifdef _DEBUG
#define new DEBUG_NEW
#endif

// 用于应用程序“关于”菜单项的 CAboutDlg 对话框

class CAboutDlg : public CDialogEx
{
public:
    CAboutDlg();

// 对话框数据
#ifdef AFX_DESIGN_TIME
    enum { IDD = IDD_ABOUTBOX };
#endif

protected:
    virtual void DoDataExchange(CDataExchange* pDX);    // DDX/DDV 支持

// 实现
protected:
    DECLARE_MESSAGE_MAP()
};

CAboutDlg::CAboutDlg() : CDialogEx(IDD_ABOUTBOX)
{
}

void CAboutDlg::DoDataExchange(CDataExchange* pDX)
{
    CDialogEx::DoDataExchange(pDX);
}

BEGIN_MESSAGE_MAP(CAboutDlg, CDialogEx)
END_MESSAGE_MAP()

```

```

// CplaneDlg 对话框

CplaneDlg::CplaneDlg(CWnd* pParent /*=nullptr*/)
    : CDialogEx(IDD_PLANE_DIALOG, pParent)
{
    m_hIcon = AfxGetApp()->LoadIcon(IDR_MAINFRAME);
}

void CplaneDlg::DoDataExchange(CDataExchange* pDX)
{
    CDialogEx::DoDataExchange(pDX);
}

BEGIN_MESSAGE_MAP(CplaneDlg, CDialogEx)
    ON_WM_SYSCOMMAND()
    ON_WM_PAINT()
    ON_WM_QUERYDRAGICON()
    ON_BN_CLICKED(IDC_BUTTON1, &CplaneDlg::OnBnClickedButton1)
    ON_BN_CLICKED(IDC_BUTTON3, &CplaneDlg::OnBnClickedButton3)
    ON_BN_CLICKED(IDC_BUTTON2, &CplaneDlg::OnBnClickedButton2)
    ON_WM_KEYDOWN()
    ON_WM_TIMER()
END_MESSAGE_MAP()

// CplaneDlg 消息处理程序

BOOL CplaneDlg::OnInitDialog()
{
    CDialogEx::OnInitDialog();

    // 将“关于...”菜单项添加到系统菜单中。

    // IDM_ABOUTBOX 必须在系统命令范围内。
    ASSERT((IDM_ABOUTBOX & 0xFFF0) == IDM_ABOUTBOX);
    ASSERT(IDM_ABOUTBOX < 0xF000);

    CMenu* pSysMenu = GetSystemMenu(FALSE);
    if (pSysMenu != nullptr)
    {
        BOOL bNameValid;
        CString strAboutMenu;

```

```

        bNameValid = strAboutMenu.LoadString(IDS_ABOUTBOX);
        ASSERT(bNameValid);
        if (!strAboutMenu.IsEmpty())
        {
            pSysMenu->AppendMenu(MF_SEPARATOR);
            pSysMenu->AppendMenu(MF_STRING, IDM_ABOUTBOX, strAboutMenu);
        }
    }

    // 设置此对话框的图标。 当应用程序主窗口不是对话框时，框架将自动
    // 执行此操作
    SetIcon(m_hIcon, TRUE);          // 设置大图标
    SetIcon(m_hIcon, FALSE);         // 设置小图标

    // TODO: 在此添加额外的初始化代码

    //设置窗口大小( 其中SetWindowPos函数的第四和第五个参数分别表示窗口的宽度,和高度,大家可以调整这两个参数
    //来改变自己窗口的大小。)
    SetWindowPos(NULL, 0, 0, 700, 500, SWP_NOMOVE);

    return TRUE;  // 除非将焦点设置到控件，否则返回 TRUE
}

void CplaneDlg::OnSysCommand(UINT nID, LPARAM lParam)
{
    if ((nID & 0xFFF0) == IDM_ABOUTBOX)
    {
        CAboutDlg dlgAbout;
        dlgAbout.DoModal();
    }
    else
    {
        CDialogEx::OnSysCommand(nID, lParam);
    }
}

// 如果向对话框添加最小化按钮，则需要下面的代码
// 来绘制该图标。 对于使用文档/视图模型的 MFC 应用程序，
// 这将由框架自动完成。

void CplaneDlg::OnPaint()
{
    if (IsIconic())
    {
        CPaintDC dc(this); // 用于绘制的设备上下文
    }
}

```

```

        SendMessage(WM_ICONERASEBKGND, reinterpret_cast<WPARAM>(dc.GetSafeHdc()), 0);

        // 使图标在工作区矩形中居中
        int cxIcon = GetSystemMetrics(SM_CXICON);
        int cyIcon = GetSystemMetrics(SM_CYICON);
        CRect rect;
        GetClientRect(&rect);
        int x = (rect.Width() - cxIcon + 1) / 2;
        int y = (rect.Height() - cyIcon + 1) / 2;

        // 绘制图标
        dc.DrawIcon(x, y, m_hIcon);
    }
    else
    {
        CDialogEx::OnPaint();
    }

    CDC* cDC = this->GetDC();           //获取窗口DC指针
    if (start == 1) {
        //重新画背景
        CRect rc;
        GetWindowRect(&rc);
        background.Draw(GetDC()->GetSafeHdc(), CRect(0, 0, rc.Width(), rc.Height()));
        m_plane.Draw(*cDC, m_planepos); //贴飞机

        for (int i = 0; i < this->max_stonenum; i++) {
            if (this->is_stone[i] == 1) {           //如果石头存在
                m_stone[i].Draw(*cDC, m_stonepos[i]); //贴石头
            }
        }
    }

    ReleaseDC(cDC);           //释放DC
    /**
    //获取窗口DC指针
    CDC* cDC = this->GetDC();
    //贴飞机
    if (start == 1) {
        m_plane.Draw(*cDC, m_planepos);
    }

    //释放DC
    ReleaseDC(cDC);
    */

```

```

}

//当用户拖动最小化窗口时系统调用此函数取得光标
//显示。
HCURSOR CplaneDlg::OnQueryDragIcon()
{
    return static_cast<HCURSOR>(m_hIcon);
}

void CplaneDlg::OnBnClickedButton1() //开始游戏
{
    // TODO: 在此添加控件通知处理程序代码
    //隐藏控件
    CWnd* pWnd = GetDlgItem(IDC_STATIC);
    pWnd->ShowWindow(FALSE);
    pWnd = GetDlgItem(IDC_BUTTON1);
    pWnd->ShowWindow(FALSE);
    pWnd = GetDlgItem(IDC_BUTTON2);
    pWnd->ShowWindow(FALSE);
    pWnd = GetDlgItem(IDC_BUTTON3);
    pWnd->ShowWindow(FALSE);

    start = 1;

    //设置背景（参考链接：https://blog.csdn.net/qq321772514/article/details/96478790）
    CRect rc;
    GetWindowRect(&rc);
    //注意这里的位置（把plane.png放到res文件里面）
    background.Load(_T("res\\background.jpg"));
    //这里让图片绘制区域和窗口大小一致
    background.Draw(GetDC()->GetSafeHdc(), CRect(0, 0, rc.Width(), rc.Height()));

    //加载飞机图片
    m_plane.Load(_T("res\\plane.png"));
    TransparentPNG(&m_plane); //使图片透明化

    //加载石头图片
    for (int i = 0; i < 10; i++) {
        m_stone[i].Load(_T("res\\stone.png"));
        TransparentPNG(&m_stone[i]); //使图片透明化
    }
}

```

```

//设置飞机初始位置
m_planepos.left = 700 / 2 - 40;           //飞机左边
m_planepos.right = m_planepos.left + 40;  //飞机的右边等于左边加上飞机的宽度
m_planepos.top = 500 - 90 + 10 ;          //所以高度是420
m_planepos.bottom = m_planepos.top + 40;

//设置石头初始位置（在顶部随机出现）    【660 430】
srand(unsigned(time(NULL)));
m_stonepos[0].left = rand() % 660;
m_stonepos[0].right = m_stonepos[0].left + 40;
m_stonepos[0].top = 0;
m_stonepos[0].bottom = 40;

this->is_stone[0] = 1;

SetTimer(1, 100, NULL);                  //设置定时器1(陨石下落)
SetTimer(2, 1000, NULL);                 //设置定时器2(产生新陨石)

}

void CplaneDlg::OnBnClickedButton3()      //退出
{
    // TODO: 在此添加控件通知处理程序代码
    exit(1);
}

void CplaneDlg::OnBnClickedButton2()      //操作说明
{
    // TODO: 在此添加控件通知处理程序代码
    MessageBox(TEXT("● 操作说明：使用方向键【上下左右】操控飞机来躲避陨石。"), TEXT("★ 提示"),
    MB_OK);
}

//监听键盘按下（参考链接：https://blog.csdn.net/hust\_xy/article/details/9327377）
void CplaneDlg::OnKeyDown(UINT nChar, UINT nRepCnt, UINT nFlags)
{
}

```



//监听键盘按下（参考链接：[https://blog.csdn.net/weixin\\_42899088/article/details/101308231](https://blog.csdn.net/weixin_42899088/article/details/101308231)）

```
BOOL CplaneDlg::PreTranslateMessage(MSG* pMsg)
{
    // TODO: 在此添加专用代码和/或调用基类
    // 键盘按键检测
    if (pMsg->message == WM_KEYDOWN)// 按键按下检测
    {
        if (pMsg->wParam == VK_LEFT && m_planepos.left >= 10 ) {
            m_planepos.left -= 10;
            m_planepos.right -= 10;
        }
        else if (pMsg->wParam == VK_RIGHT && m_planepos.right <= 670) {
            m_planepos.left += 10;
            m_planepos.right += 10;
        }
        else if (pMsg->wParam == VK_UP && m_planepos.top >= 10) {
            m_planepos.top -= 10;
            m_planepos.bottom -= 10;
        }
        else if (pMsg->wParam == VK_DOWN && m_planepos.bottom <= 440) {
            m_planepos.top += 10;
            m_planepos.bottom += 10;
        }
        OnPaint();          //更新
        /*
        CDC* cDC = this->GetDC();          //获取窗口DC指针
        if (start == 1) {
            //重新画背景
            CRect rc;
            GetWindowRect(&rc);
            background.Draw(GetDC()->GetSafeHdc(), CRect(0, 0, rc.Width(), rc.Height()));
            m_plane.Draw(*cDC, m_planepos); //贴飞机
        }
        ReleaseDC(cDC);          //释放DC
        */
    }
    return CDialogEx::PreTranslateMessage(pMsg);
}
```

//教程：[https://blog.csdn.net/hust\\_xy/article/details/9332377](https://blog.csdn.net/hust_xy/article/details/9332377)

```
void CplaneDlg::OnTimer(UINT_PTR nIDEvent)
```

```
{
    // TODO: 在此添加消息处理程序代码和/或调用默认值

    CImage tmp1;
    switch (nIDEvent)
```

```

{
case 1:
    for (int i = 0; i < this->max_stonenum; i++) {
        if (this->is_stone[i] == 1) {
            if (m_stonepos[i].bottom <= 480) {
                m_stonepos[i].top += 10;
                m_stonepos[i].bottom += 10;
            }
            if (m_stonepos[i].bottom > 470) {
                this->is_stone[i] = 0;
            }
        }

        OnPaint();
        //爆炸
        if (m_stonepos[i].bottom >= m_planepos.top && m_stonepos[i].bottom <=
m_planepos.bottom && abs(m_stonepos[i].left - m_planepos.left) < 40) {
            //让他们隐藏起来
            m_stonepos[i].top = -100;
            m_stonepos[i].bottom = -100 + 40;

            m_planepos.top = -100;
            m_planepos.bottom = -100 + 40;
            OnPaint();

            MessageBox(TEXT("游戏结束!!!"), TEXT("★ 提示"), MB_OK);
            KillTimer(1);
        }

    }

    break;

case 2:
{
    for (int i = 0; i < this->max_stonenum; i++) {
        if (this->is_stone[i] == 0) {           //没有石头的话
            is_stone[i] = 1;                   //标记为有石头
            //srand(unsigned(time(NULL)));      不能调用两次，否则不会随机
            m_stonepos[i].left = rand() % 660;
            m_stonepos[i].right = m_stonepos[i].left + 40;
            m_stonepos[i].top = 0;
            m_stonepos[i].bottom = 40;
            break;
        }
    }
}
}

```

```

        }

    }

}

    break;
default:
    break;
}

CDialogEx::OnTimer(nIDEvent);
}

```

```

// planeDlg.h: 头文件
//

#pragma once

#define MAX 20

// CplaneDlg 对话框
class CplaneDlg : public CDialogEx
{
// 构造
public:
    CplaneDlg(CWnd* pParent = nullptr); // 标准构造函数

    //添加的数据
    CImage background;
    CImage m_plane;
    CImage m_stone[MAX];
    CRect m_planepos;
    CRect m_stonepos[MAX];
    int now = 0;
    int stone_num = 0; //石头的数量
    int max_stonenum = MAX; //最大的石头数量（最多出现10个）
    int is_stone[MAX] = {0}; //石头是否存在

// 对话框数据
#ifdef AFX_DESIGN_TIME

```

```

        enum { IDD = IDD_PLANE_DIALOG };
#endif

protected:
    virtual void DoDataExchange(CDataExchange* pDX);    // DDX/DDV 支持

// 实现
protected:
    HICON m_hIcon;

    // 生成的消息映射函数
    virtual BOOL OnInitDialog();
    afx_msg void OnSysCommand(UINT nID, LPARAM lParam);
    afx_msg void OnPaint();
    afx_msg HCURSOR OnQueryDragIcon();
    DECLARE_MESSAGE_MAP()
public:
    afx_msg void OnBnClickedButton1();
    afx_msg void OnBnClickedButton3();
    afx_msg void OnBnClickedButton2();
    afx_msg void OnKeyDown(UINT nChar, UINT nRepCnt, UINT nFlags);
    virtual BOOL PreTranslateMessage(MSG* pMsg);
    afx_msg void OnTimer(UINT_PTR nIDEvent);
};

```

(2021.6.14) ——雏形2.0

```

// planeDlg.cpp: 实现文件
//

#include "pch.h"
#include "framework.h"
#include "plane.h"
#include "planeDlg.h"
#include "afxdialogex.h"

//game相关的头文件
#include <iostream>
#include <stdlib.h>
#include <time.h>
#include <conio.h>
#include <windows.h>

```

```

int start = 0;                //判断游戏是否开始

//使图片透明化（参考地址：https://blog.csdn.net/hust\_xy/article/details/9313239）
void TransparentPNG(CImage* png)
{
for (int i = 0; i < png->GetWidth(); i++)
{
    for (int j = 0; j < png->GetHeight(); j++)
    {
        unsigned char* pucColor = reinterpret_cast<unsigned char*>(png->GetPixelAddress(i,
j));

        pucColor[0] = pucColor[0] * pucColor[3] / 255;
        pucColor[1] = pucColor[1] * pucColor[3] / 255;
        pucColor[2] = pucColor[2] * pucColor[3] / 255;
    }
}
}

/*
//参考链接：https://blog.csdn.net/yunyunyx/article/details/80016585

class plane : public CplaneDlg {                //飞机类
    //飞机的位置（窗口大小是(700, 500)）
public:
    int x,y;
    HDC c;

    plane(HDC _c) {
        x = 700/2 - 40 ;
        y = 500 - 90 ;
        c = _c;

        //设置图片（参考链接：https://blog.csdn.net/q321772514/article/details/96478790）
        CImage mImage;
        //注意这里的位置（把png放到res文件里面）
        if (mImage.Load(_T("res\\plane.png")) == S_OK)
        {
            TransparentPNG(&mImage);                //使图片透明化
            mImage.Draw(c, x,y,40,40);                //底部最中间的位置
        }
    }

    void move(int _x,int _y) {

        CImage mImage;
        if (mImage.Load(_T("res\\plane.png")) == S_OK)
        {

```

```

        TransparentPNG(&mImage);          //使图片透明化
        mImage.Draw(c, x + _x, y + _y, 40, 40);
    }
}

};

class stone: public CplaneDlg {          //石头类
    //石头的位置（窗口大小是(700, 500)）
public:
    int x, y;

    stone(HDC c) {
        x = 700 / 2;
        y = 0;

        //设置背景（参考链接: https://blog.csdn.net/qq321772514/article/details/96478790）
        CRect rc;
        //GetWindowRect(&rc);
        CImage mImage;
        //注意这里的位置（把plane.png放到res文件里面）
        if (mImage.Load(_T("res\\stone.png")) == S_OK)
        {
            TransparentPNG(&mImage);          //使图片透明化
            mImage.Draw(c, x, y, 30, 30);      //画图

        }

    }

}

};
*/

#ifdef _DEBUG
#define new DEBUG_NEW
#endif

// 用于应用程序“关于”菜单项的 CAboutDlg 对话框

class CAboutDlg : public CDialogEx
{
public:
    CAboutDlg();

    // 对话框数据

```

```

#ifdef AFX_DESIGN_TIME
    enum { IDD = IDD_ABOUTBOX };
#endif

protected:
    virtual void DoDataExchange(CDataExchange* pDX);    // DDX/DDV 支持

// 实现
protected:
    DECLARE_MESSAGE_MAP()
};

CAboutDlg::CAboutDlg() : CDialogEx(IDD_ABOUTBOX)
{
}

void CAboutDlg::DoDataExchange(CDataExchange* pDX)
{
    CDialogEx::DoDataExchange(pDX);
}

BEGIN_MESSAGE_MAP(CAboutDlg, CDialogEx)
END_MESSAGE_MAP()

// CplaneDlg 对话框

CplaneDlg::CplaneDlg(CWnd* pParent /*=nullptr*/)
    : CDialogEx(IDD_PLANE_DIALOG, pParent)
{
    m_hIcon = AfxGetApp()->LoadIcon(IDR_MAINFRAME);
}

void CplaneDlg::DoDataExchange(CDataExchange* pDX)
{
    CDialogEx::DoDataExchange(pDX);
}

BEGIN_MESSAGE_MAP(CplaneDlg, CDialogEx)
    ON_WM_SYSCOMMAND()
    ON_WM_PAINT()
    ON_WM_QUERYDRAGICON()
    ON_BN_CLICKED(IDC_BUTTON1, &CplaneDlg::OnBnClickedButton1)
    ON_BN_CLICKED(IDC_BUTTON3, &CplaneDlg::OnBnClickedButton3)

```

```

        ON_BN_CLICKED(IDC_BUTTON2, &CplaneDlg::OnBnClickedButton2)
        ON_WM_KEYDOWN()
        ON_WM_TIMER()
    END_MESSAGE_MAP()

// CplaneDlg 消息处理程序

BOOL CplaneDlg::OnInitDialog()
{
    CDialogEx::OnInitDialog();

    // 将“关于...”菜单项添加到系统菜单中。

    // IDM_ABOUTBOX 必须在系统命令范围内。
    ASSERT((IDM_ABOUTBOX & 0xFFF0) == IDM_ABOUTBOX);
    ASSERT(IDM_ABOUTBOX < 0xF000);

    CMenu* pSysMenu = GetSystemMenu(FALSE);
    if (pSysMenu != nullptr)
    {
        BOOL bNameValid;
        CString strAboutMenu;
        bNameValid = strAboutMenu.LoadString(IDS_ABOUTBOX);
        ASSERT(bNameValid);
        if (!strAboutMenu.IsEmpty())
        {
            pSysMenu->AppendMenu(MF_SEPARATOR);
            pSysMenu->AppendMenu(MF_STRING, IDM_ABOUTBOX, strAboutMenu);
        }
    }

    // 设置此对话框的图标。 当应用程序主窗口不是对话框时，框架将自动
    // 执行此操作
    SetIcon(m_hIcon, TRUE);        // 设置大图标
    SetIcon(m_hIcon, FALSE);       // 设置小图标

    // TODO: 在此添加额外的初始化代码

    //设置窗口大小( 其中SetWindowPos函数的第四和第五个参数分别表示窗口的宽度,和高度,大家可以调整这两个参数
    //来改变自己窗口的大小。)
    SetWindowPos(NULL, 0, 0, 700, 500, SWP_NOMOVE);

    return TRUE;  // 除非将焦点设置到控件，否则返回 TRUE
}

```



```

void CplaneDlg::OnSysCommand(UINT nID, LPARAM lParam)
{
    if ((nID & 0xFFF0) == IDM_ABOUTBOX)
    {
        CAboutDlg dlgAbout;
        dlgAbout.DoModal();
    }
    else
    {
        CDialogEx::OnSysCommand(nID, lParam);
    }
}

// 如果向对话框添加最小化按钮，则需要下面的代码
// 来绘制该图标。 对于使用文档/视图模型的 MFC 应用程序，
// 这将由框架自动完成。

void CplaneDlg::OnPaint()
{
    if (IsIconic())
    {
        CPaintDC dc(this); // 用于绘制的设备上下文

        SendMessage(WM_ICONERASEBKGND, reinterpret_cast<WPARAM>(dc.GetSafeHdc()), 0);

        // 使图标在工作区矩形中居中
        int cxIcon = GetSystemMetrics(SM_CXICON);
        int cyIcon = GetSystemMetrics(SM_CYICON);
        CRect rect;
        GetClientRect(&rect);
        int x = (rect.Width() - cxIcon + 1) / 2;
        int y = (rect.Height() - cyIcon + 1) / 2;

        // 绘制图标
        dc.DrawIcon(x, y, m_hIcon);
    }
    else
    {
        CDialogEx::OnPaint();
    }

    CDC* cDC = this->GetDC(); //获取窗口DC指针
    if (start == 1) {
        //重新画背景
        CRect rc;
        GetWindowRect(&rc);
    }
}

```

```

        background.Draw(GetDC()->GetSafeHdc(), CRect(0, 0, rc.Width(), rc.Height()));
        m_plane.Draw(*cDC, m_planepos); //贴飞机
        m_stone.Draw(*cDC, m_stonepos); //贴石头
    }
    ReleaseDC(cDC);          //释放DC
    /**
    //获取窗口DC指针
    CDC* cDC = this->GetDC();
    //贴飞机
    if (start == 1) {
        m_plane.Draw(*cDC, m_planepos);
    }

    //释放DC
    ReleaseDC(cDC);
    */
}

//当用户拖动最小化窗口时系统调用此函数取得光标
//显示。
HCURSOR CplaneDlg::OnQueryDragIcon()
{
    return static_cast<HCURSOR>(m_hIcon);
}

void CplaneDlg::OnBnClickedButton1()          //开始游戏
{
    // TODO: 在此添加控件通知处理程序代码
    //隐藏控件
    CWnd* pWnd = GetDlgItem(IDC_STATIC);
    pWnd->ShowWindow(FALSE);
    pWnd = GetDlgItem(IDC_BUTTON1);
    pWnd->ShowWindow(FALSE);
    pWnd = GetDlgItem(IDC_BUTTON2);
    pWnd->ShowWindow(FALSE);
    pWnd = GetDlgItem(IDC_BUTTON3);
    pWnd->ShowWindow(FALSE);

    start = 1;
}

```

```

//设置背景（参考链接：https://blog.csdn.net/qq321772514/article/details/96478790）
CRect rc;
GetWindowRect(&rc);
//注意这里的位置（把plane.png放到res文件里面）
background.Load(_T("res\\background.jpg"));
//这里让图片绘制区域和窗口大小一致
background.Draw(GetDC()->GetSafeHdc(), CRect(0, 0, rc.Width(), rc.Height()));

//加载飞机图片
m_plane.Load(_T("res\\plane.png"));
TransparentPNG(&m_plane);          //使图片透明化

//加载石头图片
m_stone.Load(_T("res\\stone.png"));
TransparentPNG(&m_stone);          //使图片透明化

//设置飞机初始位置
m_planepos.left = 700 / 2 - 40;          //飞机左边
m_planepos.right = m_planepos.left + 40; //飞机的右边等于左边加上飞机的宽度
m_planepos.top = 500 - 90 + 10 ;          //所以高度是420
m_planepos.bottom = m_planepos.top + 40;

//设置石头初始位置（在顶部随机出现）    【660 430】
srand(unsigned(time(NULL)));
m_stonepos.left = rand() % 660;
m_stonepos.right = m_stonepos.left + 40;
m_stonepos.top = 0;
m_stonepos.bottom = 40;

SetTimer(1, 100, NULL);          //设置定时器

}

void CplaneDlg::OnBnClickedButton3()          //退出
{
    // TODO: 在此添加控件通知处理程序代码
    exit(1);
}

void CplaneDlg::OnBnClickedButton2()          //操作说明

```

```

{
    // TODO: 在此添加控件通知处理程序代码
    MessageBox(TEXT("● 操作说明：使用方向键【上下左右】操控飞机来躲避陨石。"), TEXT("★ 提示"),
    MB_OK);
}

//监听键盘按下（参考链接：https://blog.csdn.net/hust\_xy/article/details/9327377）
void CplaneDlg::OnKeyDown(UINT nChar, UINT nRepCnt, UINT nFlags)
{
}

//监听键盘按下（参考链接：https://blog.csdn.net/weixin\_42899088/article/details/101308231）
BOOL CplaneDlg::PreTranslateMessage(MSG* pMsg)
{
    // TODO: 在此添加专用代码和/或调用基类
    // 键盘按键检测
    if (pMsg->message == WM_KEYDOWN)// 按键按下检测
    {
        if (pMsg->wParam == VK_LEFT && m_planepos.left >= 10 ) {
            m_planepos.left -= 10;
            m_planepos.right -= 10;
        }
        else if (pMsg->wParam == VK_RIGHT && m_planepos.right <= 670) {
            m_planepos.left += 10;
            m_planepos.right += 10;
        }
        else if (pMsg->wParam == VK_UP && m_planepos.top >= 10) {
            m_planepos.top -= 10;
            m_planepos.bottom -= 10;
        }
        else if (pMsg->wParam == VK_DOWN && m_planepos.bottom <= 440) {
            m_planepos.top += 10;
            m_planepos.bottom += 10;
        }
        OnPaint();          //更新
        /*
        CDC* cDC = this->GetDC();          //获取窗口DC指针
        if (start == 1) {
            //重新画背景
            CRect rc;
            GetWindowRect(&rc);
            background.Draw(GetDC()->GetSafeHdc(), CRect(0, 0, rc.Width(), rc.Height()));
            m_plane.Draw(*cDC, m_planepos); //贴飞机
        }
        ReleaseDC(cDC);          //释放DC
        */
    }
}

```

```

    }

    return CDialogEx::PreTranslateMessage(pMsg);
}

//教程: https://blog.csdn.net/hust\_xy/article/details/9332377
void CplaneDlg::OnTimer(UINT_PTR nIDEvent)
{
    // TODO: 在此添加消息处理程序代码和/或调用默认值
    switch (nIDEvent)
    {
    case 1:
        if (m_stonepos.bottom <= 440) {
            m_stonepos.top += 10;
            m_stonepos.bottom += 10;
        }
        OnPaint();
        if ( m_stonepos.bottom >= m_planepos.top && m_stonepos.bottom <= m_planepos.bottom
        && abs(m_stonepos.left - m_planepos.left) < 40 ) {
            //让他们隐藏起来
            m_stonepos.top = -100 ;
            m_stonepos.bottom = -100 + 40;

            m_planepos.top = -100;
            m_planepos.bottom = -100 + 40;
            OnPaint();

            MessageBox(TEXT("游戏结束!!!"), TEXT("★ 提示"), MB_OK);
            KillTimer(1);
        }

        break;
    default:
        break;
    }

    CDialogEx::OnTimer(nIDEvent);
}

```

(2021.6.14) ——雏形1.0

// planeDlg.cpp: 实现文件

```
//

#include "pch.h"
#include "framework.h"
#include "plane.h"
#include "planeDlg.h"
#include "afxdialogex.h"

//game相关的头文件
#include <iostream>
#include <stdlib.h>
#include <time.h>
#include <conio.h>
#include <windows.h>

int start = 0;           //判断游戏是否开始

//使图片透明化（参考地址：https://blog.csdn.net/hust\_xy/article/details/9313239）
void TransparentPNG(CImage* png)
{
    for (int i = 0; i < png->GetWidth(); i++)
    {
        for (int j = 0; j < png->GetHeight(); j++)
        {
            unsigned char* pucColor = reinterpret_cast<unsigned char*>(png->GetPixelAddress(i, j));

            pucColor[0] = pucColor[0] * pucColor[3] / 255;
            pucColor[1] = pucColor[1] * pucColor[3] / 255;
            pucColor[2] = pucColor[2] * pucColor[3] / 255;
        }
    }
}

/*
//参考链接：https://blog.csdn.net/yunyunyx/article/details/80016585

class plane : public CplaneDlg {           //飞机类
    //飞机的位置（窗口大小是(700, 500)）
public:
    int x,y;
    HDC c;

    plane(HDC _c) {
        x = 700/2 - 40 ;
        y = 500 - 90 ;
        c = _c;
    }
};
*/

```

```

//设置图片（参考链接：https://blog.csdn.net/qq321772514/article/details/96478790）
CImage mImage;
//注意这里的位置（把png放到res文件里面）
if (mImage.Load(_T("res\\plane.png")) == S_OK)
{
    TransparentPNG(&mImage);          //使图片透明化
    mImage.Draw(c, x,y,40,40);        //底部最中间的位置
}
}

void move(int _x,int _y) {

    CImage mImage;
    if (mImage.Load(_T("res\\plane.png")) == S_OK)
    {
        TransparentPNG(&mImage);      //使图片透明化
        mImage.Draw(c, x + _x, y + _y, 40, 40);
    }
}

};

class stone: public CplaneDlg {                //石头类
//石头的位置（窗口大小是(700, 500)）
public:
    int x, y;

    stone(HDC c) {
        x = 700 / 2;
        y = 0;

        //设置背景（参考链接：https://blog.csdn.net/qq321772514/article/details/96478790）
        CRect rc;
        //GetWindowRect(&rc);
        CImage mImage;
        //注意这里的位置（把plane.png放到res文件里面）
        if (mImage.Load(_T("res\\stone.png")) == S_OK)
        {
            TransparentPNG(&mImage);    //使图片透明化
            mImage.Draw(c, x, y, 30, 30); //画图
        }

    }

}

};

```

```

*/

#ifdef _DEBUG
#define new DEBUG_NEW
#endif

// 用于应用程序“关于”菜单项的 CAboutDlg 对话框

class CAboutDlg : public CDialogEx
{
public:
    CAboutDlg();

// 对话框数据
#ifdef AFX_DESIGN_TIME
    enum { IDD = IDD_ABOUTBOX };
#endif

protected:
    virtual void DoDataExchange(CDataExchange* pDX);    // DDX/DDV 支持

// 实现
protected:
    DECLARE_MESSAGE_MAP()
};

CAboutDlg::CAboutDlg() : CDialogEx(IDD_ABOUTBOX)
{
}

void CAboutDlg::DoDataExchange(CDataExchange* pDX)
{
    CDialogEx::DoDataExchange(pDX);
}

BEGIN_MESSAGE_MAP(CAboutDlg, CDialogEx)
END_MESSAGE_MAP()

// CplaneDlg 对话框

CplaneDlg::CplaneDlg(CWnd* pParent /*=nullptr*/)
    : CDialogEx(IDD_PLANE_DIALOG, pParent)

```



```

{
    m_hIcon = AfxGetApp()->LoadIcon(IDR_MAINFRAME);
}

void CplaneDlg::DoDataExchange(CDataExchange* pDX)
{
    CDialogEx::DoDataExchange(pDX);
}

BEGIN_MESSAGE_MAP(CplaneDlg, CDialogEx)
    ON_WM_SYSCOMMAND()
    ON_WM_PAINT()
    ON_WM_QUERYDRAGICON()
    ON_BN_CLICKED(IDC_BUTTON1, &CplaneDlg::OnBnClickedButton1)
    ON_BN_CLICKED(IDC_BUTTON3, &CplaneDlg::OnBnClickedButton3)
    ON_BN_CLICKED(IDC_BUTTON2, &CplaneDlg::OnBnClickedButton2)
    ON_WM_KEYDOWN()
END_MESSAGE_MAP()

// CplaneDlg 消息处理程序

BOOL CplaneDlg::OnInitDialog()
{
    CDialogEx::OnInitDialog();

    // 将“关于...”菜单项添加到系统菜单中。

    // IDM_ABOUTBOX 必须在系统命令范围内。
    ASSERT((IDM_ABOUTBOX & 0xFFF0) == IDM_ABOUTBOX);
    ASSERT(IDM_ABOUTBOX < 0xF000);

    CMenu* pSysMenu = GetSystemMenu(FALSE);
    if (pSysMenu != nullptr)
    {
        BOOL bNameValid;
        CString strAboutMenu;
        bNameValid = strAboutMenu.LoadString(IDS_ABOUTBOX);
        ASSERT(bNameValid);
        if (!strAboutMenu.IsEmpty())
        {
            pSysMenu->AppendMenu(MF_SEPARATOR);
            pSysMenu->AppendMenu(MF_STRING, IDM_ABOUTBOX, strAboutMenu);
        }
    }
}

```

```

// 设置此对话框的图标。 当应用程序主窗口不是对话框时，框架将自动
// 执行此操作
SetIcon(m_hIcon, TRUE);          // 设置大图标
SetIcon(m_hIcon, FALSE);        // 设置小图标

// TODO: 在此添加额外的初始化代码

//设置窗口大小( 其中SetWindowPos函数的第四和第五个参数分别表示窗口的宽度,和高度,大家可以调整这两个参数
来改变自己窗口的大小。)
SetWindowPos(NULL, 0, 0, 700, 500, SWP_NOMOVE);

return TRUE;  // 除非将焦点设置到控件，否则返回 TRUE
}

void CplaneDlg::OnSysCommand(UINT nID, LPARAM lParam)
{
    if ((nID & 0xFFF0) == IDM_ABOUTBOX)
    {
        CAboutDlg dlgAbout;
        dlgAbout.DoModal();
    }
    else
    {
        CDialogEx::OnSysCommand(nID, lParam);
    }
}

// 如果向对话框添加最小化按钮，则需要下面的代码
// 来绘制该图标。 对于使用文档/视图模型的 MFC 应用程序，
// 这将由框架自动完成。

void CplaneDlg::OnPaint()
{
    if (IsIconic())
    {
        CPaintDC dc(this); // 用于绘制的设备上下文

        SendMessage(WM_ICONERASEBKGND, reinterpret_cast<WPARAM>(dc.GetSafeHdc()), 0);

        // 使图标在工作区矩形中居中
        int cxIcon = GetSystemMetrics(SM_CXICON);
        int cyIcon = GetSystemMetrics(SM_CYICON);
        CRect rect;
        GetClientRect(&rect);
        int x = (rect.Width() - cxIcon + 1) / 2;
        int y = (rect.Height() - cyIcon + 1) / 2;
    }
}

```

```

        // 绘制图标
        dc.DrawIcon(x, y, m_hIcon);
    }
    else
    {
        CDialogEx::OnPaint();
    }

    /**
    //获取窗口DC指针
    CDC* cDC = this->GetDC();
    //贴飞机
    if (start == 1) {
        m_plane.Draw(*cDC, m_planepos);
    }

    //释放DC
    ReleaseDC(cDC);
    */

}

//当用户拖动最小化窗口时系统调用此函数取得光标
//显示。
HCURSOR CplaneDlg::OnQueryDragIcon()
{
    return static_cast<HCURSOR>(m_hIcon);
}

void CplaneDlg::OnBnClickedButton1() //开始游戏
{
    // TODO: 在此添加控件通知处理程序代码
    //隐藏控件
    CWnd* pWnd = GetDlgItem(IDC_STATIC);
    pWnd->ShowWindow(FALSE);
    pWnd = GetDlgItem(IDC_BUTTON1);
    pWnd->ShowWindow(FALSE);
    pWnd = GetDlgItem(IDC_BUTTON2);
    pWnd->ShowWindow(FALSE);
    pWnd = GetDlgItem(IDC_BUTTON3);

```

```

pWnd->ShowWindow(FALSE);

start = 1;

//设置背景（参考链接：https://blog.csdn.net/qq321772514/article/details/96478790）
CRect rc;
GetWindowRect(&rc);
//注意这里的位置（把plane.png放到res文件里面）
background.Load(_T("res\\background.jpg"));
//这里让图片绘制区域和窗口大小一致
background.Draw(GetDC()->GetSafeHdc(), CRect(0, 0, rc.Width(), rc.Height()));

//加载飞机图片
m_plane.Load(_T("res\\plane.png"));
TransparentPNG(&m_plane);          //使图片透明化

//加载石头图片
m_stone.Load(_T("res\\stone.png"));
TransparentPNG(&m_stone);          //使图片透明化

//设置飞机初始位置
m_planepos.left = 700 / 2 - 40;          //飞机左边
m_planepos.right = m_planepos.left + 40; //飞机的右边等于左边加上飞机的宽度
m_planepos.top = 500 - 90 + 10;          //所以高度是420
m_planepos.bottom = m_planepos.top + 40;

//设置石头初始位置

}

void CplaneDlg::OnBnClickedButton3()          //退出
{
    // TODO: 在此添加控件通知处理程序代码
    exit(1);
}

void CplaneDlg::OnBnClickedButton2()          //操作说明
{
    // TODO: 在此添加控件通知处理程序代码
    MessageBox(TEXT("● 操作说明：使用方向键【上下左右】操控飞机来躲避陨石。"), TEXT("★ 提示"),

```

```

MB_OK);
}

//监听键盘按下（参考链接：https://blog.csdn.net/hust\_xy/article/details/9327377）
void CplaneDlg::OnKeyDown(UINT nChar, UINT nRepCnt, UINT nFlags)
{
}

//监听键盘按下（参考链接：https://blog.csdn.net/weixin\_42899088/article/details/101308231）
BOOL CplaneDlg::PreTranslateMessage(MSG* pMsg)
{
    // TODO: 在此添加专用代码和/或调用基类
    // 键盘按键检测
    if (pMsg->message == WM_KEYDOWN)// 按键按下检测
    {
        if (pMsg->wParam == VK_LEFT && m_planepos.left >= 10 ) {
            m_planepos.left -= 10;
            m_planepos.right -= 10;
        }
        else if (pMsg->wParam == VK_RIGHT && m_planepos.right <= 670) {
            m_planepos.left += 10;
            m_planepos.right += 10;
        }
        else if (pMsg->wParam == VK_UP && m_planepos.top >= 10) {
            m_planepos.top -= 10;
            m_planepos.bottom -= 10;
        }
        else if (pMsg->wParam == VK_DOWN && m_planepos.bottom <= 440) {
            m_planepos.top += 10;
            m_planepos.bottom += 10;
        }
    }

    //
    CDC* cDC = this->GetDC();          //获取窗口DC指针
    if (start == 1) {
        //重新画背景
        CRect rc;
        GetWindowRect(&rc);
        background.Draw(GetDC()->GetSafeHdc(), CRect(0, 0, rc.Width(), rc.Height()));
        m_plane.Draw(*cDC, m_planepos); //贴飞机
    }
    ReleaseDC(cDC);                    //释放DC
}

```

```
return CDialogEx::PreTranslateMessage(pMsg);  
}
```