**目 录**

[1. 概述 2](#_Toc441262585)

[1.1 实训项目简介 2](#_Toc441262586)

[1.2 实训功能说明 2](#_Toc441262587)

[1.2.1 基本功能 2](#_Toc441262588)

[1.2.2 附加功能 2](#_Toc441262589)

[2. 相关技术 3](#_Toc441262590)

[2.1 Windows定时器技术 3](#_Toc441262591)

[2.2 透明贴图实现技术 3](#_Toc441262592)

[2.3 CObList链表 4](#_Toc441262593)

[2.4获取矩形区域 5](#_Toc441262594)

[2.5使用AfxMessageBox显示游戏过程中的提示信息 5](#_Toc441262595)

[2.6内存释放 5](#_Toc441262596)

[2.7 CImageList处理爆炸效果 5](#_Toc441262597)

[2.8对话框的应用 5](#_Toc441262598)

[3. 总体设计与详细设计 6](#_Toc441262599)

[3.1 系统模块划分 6](#_Toc441262600)

[3.2 主要功能模块 6](#_Toc441262601)

[3.2.1 系统对象类图 6](#_Toc441262602)

[3.2.2 系统主程序活动图 7](#_Toc441262603)

[3.2.3 系统部分流程图 8](#_Toc441262604)

[4. 项目程序测试 9](#_Toc441262605)

[4.1战机移动及子弹发射模块测试 9](#_Toc441262606)

[4.2敌机及子弹模块测试 10](#_Toc441262607)

[4.3爆炸模块测试 10](#_Toc441262608)

[5. 实训中遇到的主要问题及解决方法 10](#_Toc441262609)

# 1. 概述

## 1.1 实训项目简介

本次实训项目是做一个飞机大战的游戏，应用MFC编程，完成一个界面简洁流畅、游戏方式简单，玩起来易于上手的桌面游戏。该飞机大战项目运用的主要技术即是MFC编程中的一些函数、链表思想以及贴图技术。

## 1.2 实训功能说明

### 1.2.1 基本功能

（1）设置一个战机具有一定的速度，通过键盘，方向键可控制战机的位置，空格键发射子弹。

（2）界面中敌机出现的位置，以及敌机子弹的发射均为随机的，敌机与敌机子弹均具有一定的速度，且随着关卡难度的增大，数量和速度均增加。

（3）对于随机产生的敌机和敌机子弹，若超过矩形区域，则释放该对象。

（4）添加爆炸效果，包括战机子弹打中敌机爆炸、敌机子弹打中战机爆炸、战机与敌机相撞爆炸以及战机子弹与敌机子弹相撞爆炸四种爆炸效果。且爆炸发生后敌机、子弹、子弹均消失，战机生命值减一。

### 1.2.2 附加功能

（1） 为游戏界面添加了背景图片，并在战机发射子弹、战机击中敌机、敌机击中战机、以及战机敌机相撞时均添加了背景音效。

（2）为游戏设置了不同的关卡，每个关卡难度不同，敌机与敌机子弹的速度随着关卡增大而加快，进入第二关以后敌机从上下方均会随机出现，且随机发射子弹。

（3）第一关卡敌机从上方飞出，速度一定，战机每打掉一直敌机则增加一分，每积十分，则为战机增加一个生命值，当战机得分超过50分则可进入下一关；进入第二、三关时敌机速度加快，分别从上下两方飞出，此时战机每得分20、30分，才会增加一个生命值，得分超过100、150分则进入下一关、通关。

（4） 在游戏界面输出当前游戏进行信息，包括当前得分、当前关卡以及击中敌机数量。

（5）增加了鼠标控制战机位置这一效果，战绩的位置随着鼠标的移动而移动，并且点击鼠标左键可使得战机发射子弹。

（6）实现了暂停游戏的功能，玩家可通过键盘上的‘Z’键，对游戏进行暂停。

（7）通过对话框的弹出可提示玩家是否查看游戏说明、是否进入下一关、是否重新开始等消息，使得玩家可自己选择。

# 2. 相关技术

## 2.1 Windows定时器技术

Windows定时器是一种输入设备，它周期性地在每经过一个指定的时间间隔后就通知应用程序一次。程序将时间间隔告诉Windows，然后Windows给您的程序发送周期性发生的WM\_TIMER消息以表示时间到了。本程序中使用多个定时器，分别控制不同的功能。在MFC的API函数中使用SetTimer()函数设置定时器，设置系统间隔时间，在OnTimer()函数中实现响应定时器的程序。

## 2.2 透明贴图实现技术

绘制透明位图的关键就是创建一个“掩码”位图(mask bitmap)，这个“掩码”位图是一个单色位图，它是位图中图像的一个单色剪影。

在详细介绍实现过程之前先介绍下所使用的画图函数以及函数参数所代表的功能；整个绘制过程需要使用到BitBlt()函数。整个功能的实现过程如下：

(1) 创建一张大小与需要绘制图像相同的位图作为“掩码”位图；

(2) 将新创建的“掩码”位图存储至掩码位图的设备描述表中；

(3) 把位图设备描述表的背景设置成“透明色”，不需要显示的颜色；

(4) 复制粘贴位图到“掩码”位图的设备描述表中，这个时候“掩码”位图设备描述表中存放的位图与位图设备描述表中的位图一样；

(5) 把需要透明绘制的位图与对话框绘图相应区域的背景进行逻辑异或操作绘制到对话框上；

(6) 把“掩码”位图与这个时候对话框相应区域的背景进行逻辑与的操作；

(7) 重复步骤5的操作，把需要透明绘制的位图与对话框绘图相应区域的背景进行逻辑异或操作绘制到对话框上；

(8) 最后把系统的画笔还给系统，删除使用过的GDIObject，释放非空的指针，最后把新建的设备描述表也删除。

## 2.3 CObList链表

MFC类库中提供了丰富的CObList类的成员函数，此程序主要用到的成员函数如下：  
(1) 构造函数，为CObject指针构造一个空的列表。

(2) [GetHead](mk:@MSITStore:C:\Documents%20and%20Settings\k01\桌面\MFC类库详解.chm::/Class/CObList_GetHead.htm)()，访问链表首部，返回列表中的首元素（列表不能为空）。  
(3) [AddTail](mk:@MSITStore:C:\Documents%20and%20Settings\k01\桌面\MFC类库详解.chm::/Class/CObList_AddTail.htm)()，在列表尾增加一个元素或另一个列表的所有元素。

(4) [RemoveAll](mk:@MSITStore:C:\Documents%20and%20Settings\k01\桌面\MFC类库详解.chm::/Class/CObList_RemoveAll.htm)(),删除列表中所有的元素。

(5) [GetNext](mk:@MSITStore:C:\Documents%20and%20Settings\k01\桌面\MFC类库详解.chm::/Class/CObList_GetNext.htm)(),返回列表中尾元素的位置。

(6) [GetHeadPosition](mk:@MSITStore:C:\Documents%20and%20Settings\k01\桌面\MFC类库详解.chm::/Class/CObList_GetHeadPosition.htm)()，返回列表中首元素的位置。

(7) [RemoveAt](mk:@MSITStore:C:\Documents%20and%20Settings\k01\桌面\MFC类库详解.chm::/Class/CObList_RemoveAt.htm)()，从列表中删除指定位置的元素。

(8) [GetCount](mk:@MSITStore:C:\Documents%20and%20Settings\k01\桌面\MFC类库详解.chm::/Class/CObList_GetCount.htm)()，返回列表中的元素数。

在CPlaneGameView.h文件中声明各游戏对象与游戏对象链表：

（1）创建各个游戏对象

MyPlane \*myPlane;

EnemyPlane \*enemyPlane;

MyBullet \*myBullet;

EnemyBullet \*enemyBullet;

Explosion \*explosion;

（2）创建游戏对象存储链表

CObList myPlaneList;

CObList enemyPlaneList;

CObList myBulletList;

CObList enemyBulletList;

CObList explosionList;

## 2.4获取矩形区域

首先，使用CRect定义一个对象，然后使用GetClientRect(&对象名)函数，获取界面的矩形区域rect.Width() 为矩形区域的宽度，rect.Height()为矩形区域的高度。

使用IntersectRect(&,&))函数来判断两个源矩形是否有重合的部分。如果有不为空，则返回非零值；否则，返回0。

## 2.5使用AfxMessageBox显示游戏过程中的提示信息

AfxMessageBox()是[模态对话框](http://baike.baidu.com/view/3148035.htm" \t "_blank)，你不进行确认时程序是否往下运行时，它会阻塞你当前的线程，除非你程序是[多线程](http://baike.baidu.com/view/65706.htm)的程序，否则只有等待模态对话框被确认。

在MFC中，afxmessagebox是全局的对话框最安全，也最方便。

## 2.6内存释放

在VC/MFC用CDC绘图时，频繁的刷新，屏幕会出现闪烁的现象，CPU时间占用率相当高，绘图效率极低，很容易出现程序崩溃。及时的释放程序所占用的内存资源是非常重要的。

在程序中使用到的链表、刷子等占用内存资源的对象都要及时的删除。Delete Brush，

List.removeall()等。

## 2.7 CImageList处理爆炸效果

爆炸效果是连续的显示一系列的图片。如果把每一张图片都显示出来的话，占用的时间是非常多的，必然后导致程序的可行性下降。CImageList是一个“图象列表”是相同大小图象的集合，每个图象都可由其基于零的索引来参考。可以用来存放爆炸效果的一张图片，使用Draw()函数来绘制在某拖拉操作中正被拖动的图象，即可连续绘制出多张图片做成的爆炸效果。

## 2.8对话框的应用

在设置游戏难度、子弹的速度等，使用对话框进行设置非常方便，又体现出界面的友好。

对话框的应用过程如下：

(1). 资源视图下，添加Dialog对话框。然后添加使用到的控件，并修改控件的ID以便于后面的使用。

(2). 为对话框添加类，在对话框模式下，点击项目，添加类。

(3). 在类视图中，为对话框类添加成员变量（控件变量）。设置变量的名称、类型、最值等信息。

(4). 在资源视图菜单中，选择相应的菜单项，右击添加时间监听程序，设置函数处理程序名称。

(5). 在处理程序函数中添加相应的信息。

# 3. 总体设计与详细设计

## 3.1 系统模块划分

该飞机大战游戏程序分为游戏背景位图绘制模块、各游戏对象绘制模块、游戏对象之间的碰撞模块、爆炸效果产生模块、游戏界面输出玩家得分关卡信息模块。

其中在游戏对象绘制模块中，战机是唯一对象，在游戏开始时产生该对象，赋予其固定的生命值，当其与敌机对象、敌机子弹碰撞时使其生命值减一，直至生命值为零，便删除战机对象。敌机对象与敌机子弹对象的绘制中采用定时器技术，定时产生。爆炸对象初始化为空，当游戏过程中即时发生碰撞时，在碰撞位置产生爆炸对象，添加到爆炸链表中。

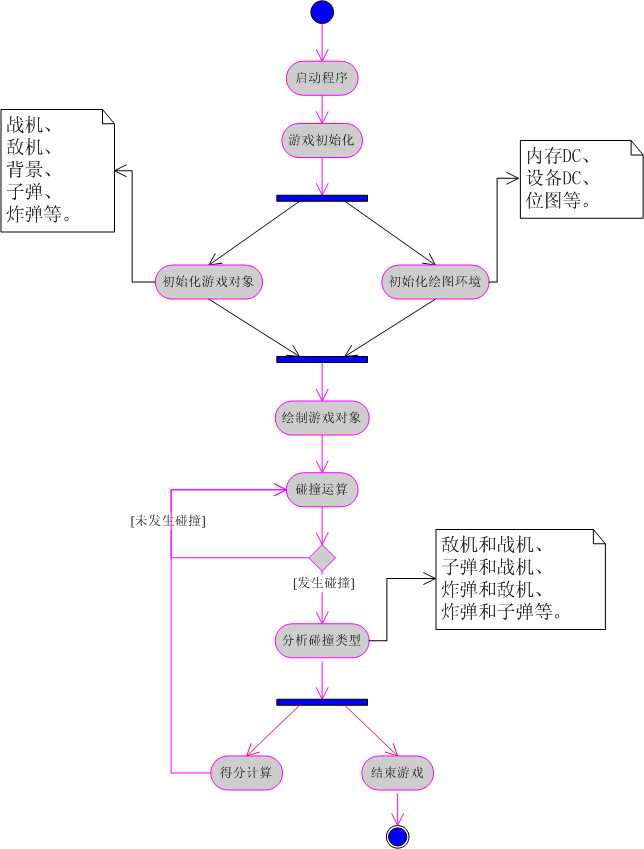
## 3.2 主要功能模块

## 3.2.1 系统对象类图

CGameObject是各个游戏对象的抽象父类，继承自CObject类，其他的类：战机类、敌机类、爆炸类、子弹类、子弹类、文字类都继承了此类。

每个游戏对象类中既继承了来自父类CGameObject的属性，又有自己的特有属性和方法。

### 3.2.2 系统主程序活动图



### 3.2.3 系统部分流程图

(1) 该飞机大战游戏执行流程图：



(2) 利用定时器定时产生敌机并绘制敌机流程图



# 4. 项目程序测试

## 4.1战机移动及子弹发射模块测试

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例 | 预期结果 | 实际结果 | 问题描述 | 修改方案 |
| 点击A键或鼠标左移 | 战机向左移动 | 战机向左移动 |  |  |
| 点击D键或鼠标右移 | 战机向右移动 | 战机向右移动 |  |  |
| 点击W键或鼠标上移 | 战机向上移动 | 战机向上移动 |  |  |
| 点击S键或鼠标上移 | 战机向下移动 | 战机向下移动 |  |  |

## 4.2敌机及子弹模块测试

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例 | 预期结果 | 实际结果 | 问题描述 | 修改方案 |
| 玩家得分<50(未通过第一关) | 敌机只从上方飞出，速度固定 | 敌机只从上方飞出，速度固定 |  |  |
| 玩家得分>50(通过第一关后) | 敌机从上下两方向均可飞出，且速度不断增加 | 敌机从上下两方向均可飞出，且速度不断增加 |  |  |

## 4.3爆炸模块测试

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例 | 预期结果 | 实际结果 | 问题描述 | 修改方案 |
| 战机子弹打中敌机 | 敌机位置处爆炸，敌机消失，战机生命-1 | 敌机位置处爆炸，敌机消失，战机生命-1 |  |  |
| 敌机子弹打中战机 | 战机位置处爆炸，战机生命-1 | 战机位置处爆炸，战机生命-1 |  |  |
| 敌机战机相撞 | 敌机位置处爆炸，敌机消失，战机生命-1 | 敌机位置处爆炸，敌机消失，战机生命-1 |  |  |
| 战机子弹与敌机子弹相撞 | 敌机子弹处爆炸，子弹与子弹均消失消失 | 敌机子弹处爆炸，子弹与子弹均消失消失 |  |  |
| 战机生命值==0 | 战机消失，GameOver或者过关 | 战机消失，GameOver或者过关 |  |  |

# 5. 实训中遇到的主要问题及解决方法

（1）由于对C++的面向对象的思想和逻辑思路不熟悉，不明白其中的封装之类的以及多态的思想，致使开始真正的进入实训接触到项目时没有开发思路，通过逐步查询书籍整理C++面向对象编程思路，才逐步理清项目的开发步骤。

（2）本飞机大战的游戏要求使用链表实现各游戏对象的存储和释放，由于链表知识掌握的不牢固，使用起来总是出现这样那样的错误，给整个游戏开发带来了很大的障碍，通过不断的调试修改，最终使程序正确运行。

（3）在绘制各种游戏对象—敌机和敌机子弹时，开始使用随机函数，画出敌机时而很少，总是打不到预定的效果，后来经过修改使用定时器产生敌机和敌机子弹，使整个游戏更加人性化。