

“C++程序设计与训练”课程大作业

项目报告

项目名称：图片管理系统设计与开发

姓名： 夏 斐

学号： 2012011417

班级： 自 22

日期： 2013 年 9 月

目 录

1 系统功能设计	5
1.1 图片绘制模块	5
1.1.1 图片绘制	5
1.1.2 图片的载入与保存	5
1.1.3 图象的对象管理	5
1.1.4 命令行输入	5
1.2 多用户模块	5
1.2.1 多用户的注册和登录	5
1.2.2 多用户的权限管理	6
1.2.3 用户授权管理	6
1.3 文件管理模块	6
1.3.1 文件搜索	6
1.3.2 图片组管理	6
2 系统总体结构	7
2.1 概要设计	7
2.1.1 框架-文档-视图的架构	7
2.1.2 管理绘制对象的类	8
2.1.3 系统与数据库的交互	9
2.2 小组分工	9
3 本人工作内容	11
3.1 主体框架的搭建	11
3.1.1 主框架的构建	11
3.1.2 主框架和工具栏及对话框之间的交互	12
3.1.3 文档与视图的创建和维护	14
3.2 绘图模块的编写	14
3.2.1 绘图对象的设计	14
3.2.2 文档与视图的交互	15
3.2.3 对象的选定与橡皮擦工具	16
3.3 权限管理模块的编写	17
3.3.1 用户登录与注册	17
3.3.2 文档存储与打开界面	18
3.3.3 授权管理	19
4 项目总结	20
4.1 文件的版本管理	20
4.2 关于数据成员存放的位置	20
4.3 工具条与控件	21
4.4 改善用户体验是永无止境的	21
5 相关问题的说明	22
5.1 数据库的配置	22
5.2 更多拓展功能的说明（操作实例）	22
5.2.1 对象橡皮擦与对象选定	23
5.2.2 命令行的其他功能	24

5.2.3 贝塞尔曲线的绘制	25
5.2.5 对象列表功能	25
5.2.6 图片的载入	26
5.2.7 绘图过程中的快捷键	26
5.2.8 文件的删除	27

1 系统功能设计

1.1 图片绘制模块

1.1.1 图片绘制

图片绘制功能是这个软件的主体，也是能够创造内容的部分。该模块需要程序能够让用户方便地在工作区中绘制各种几何体，如圆、矩形、直线、线段、曲线等，除了规范的几何体，程序还需要能够让用户随意绘制图线。如果需要提升用户体验，还需要能够让用户自行选择绘图的颜色，线的样式，以及选择背景色等。这部分可以参考 Windows 自带的画图，以及 Adobe Photoshop 等。

1.1.2 图片的载入与保存

用户可以从一张已经存在的图像中读取图像并显示，可以在这张图片上进行绘图，并且进行保存。载入图片有两种方式，一是打开文件进行浏览，二是在空白文档中载入一张图片。

1.1.3 图象的对象管理

由于在绘制过程中需要能够整体移动或者移除一个绘制出的对象，所以用户进行的每一次绘制都应该是以对象存储的。进行移动时，只需要改变对象的坐标，擦除时，只需要将对象删除即可。

1.1.4 命令行输入

这部分需要程序有一个能够输入命令行的地方，在编辑框内输入命令并且按回车之后，系统就要在画布上画出相应地图形。

1.2 多用户模块

1.2.1 多用户的注册和登录

系统需要支持多用户，则首先需要让用户注册和登录，每一次登录时，用户或者使用已有账号登录，或者注册新账号。注册新账号时，系统能够将其保存在

数据库中，进行登录时，系统会将输入数据和数据库内数据比较决定是否允许登录。

1.2.2 多用户的权限管理

每个用户可以创建、查看、修改、删除文件，这一部分是和绘图部分紧密相连的，每个用户只能操作自己创建的文件，不能操作别人创建的文件。这部分内容也是通过数据库实现的，在进行文件的保存时，系统维护一个数据库，为里面加入数据，亦即文件的创建者拥有了对于该文件的所有权限，在进行其他操作时，系统将当前用户信息以及文件信息与数据库内数据进行比对，决定允许或者拒绝用户的操作。

1.2.3 用户授权管理

这部分需要允许一个用户将某种权限赋予另外一个用户，在进行操作时，当前登录用户能够看到系统中所有注册用户的列表，并且可以选择文件以及权限，进行授权，这时系统也会维护关于权限的数据库，判断是否能够授权，如果能够则会为数据库增加条目。

在以上两个模块中，我们认为存在两种权限，一种是读权限，一种是写权限，如果某个用户拥有了写权限，则自动拥有了读权限。

1.3 文件管理模块

1.3.1 文件搜索

用户能够依据文件名进行搜索或者依据权限进行搜索，当依据文件名进行搜索时，用户输入关键字，所有含有这个关键字的数据库中的文件都会被显示出来。当依据权限进行搜索时，用户可以看到当前所有具有读权限的文件，并且可以打开以及进行进一步操作。

1.3.2 图片组管理

用户可以新建图片组并且将一些图片放入图片组中进行图片的分组管理。

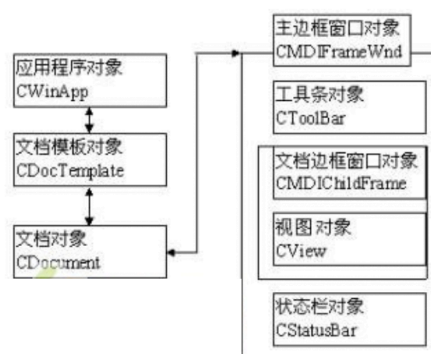
2 系统总体结构

2.1 概要设计

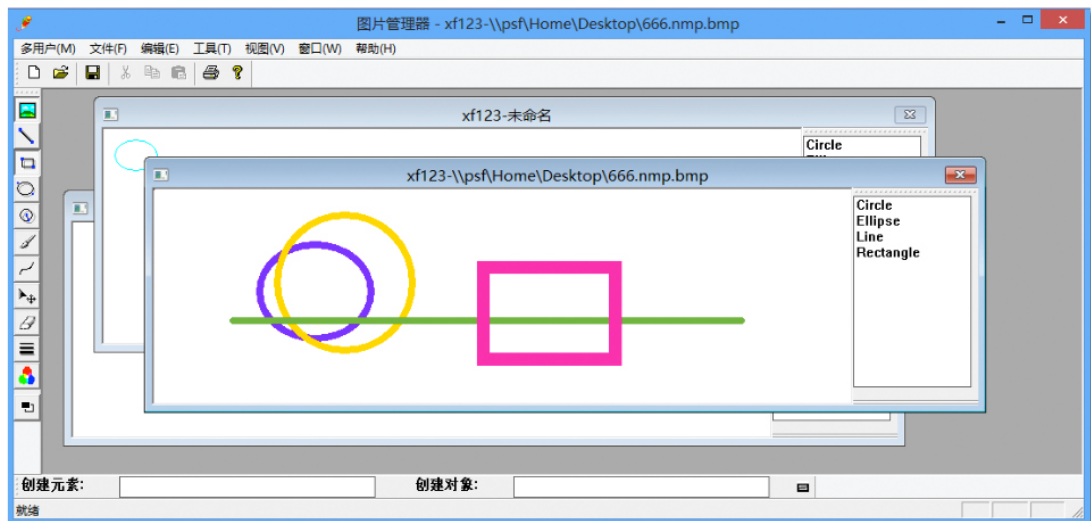
对于这样一个系统，我们想基于多文档来实现，这样用户可以同时操作多个文档。所以对于这样一个系统，主要有三个值得注意的架构，即框架-文档-视图的架构，管理绘制出的对象的一系列类，以及系统与数据库的交互。

2.1.1 框架-文档-视图的架构

由于我们采用了多文档视图结构，所以共有五个主类，应用程序类、主框架窗口类，子框架窗口类，视图类，文档类，它们之间的相互关系如图所示：



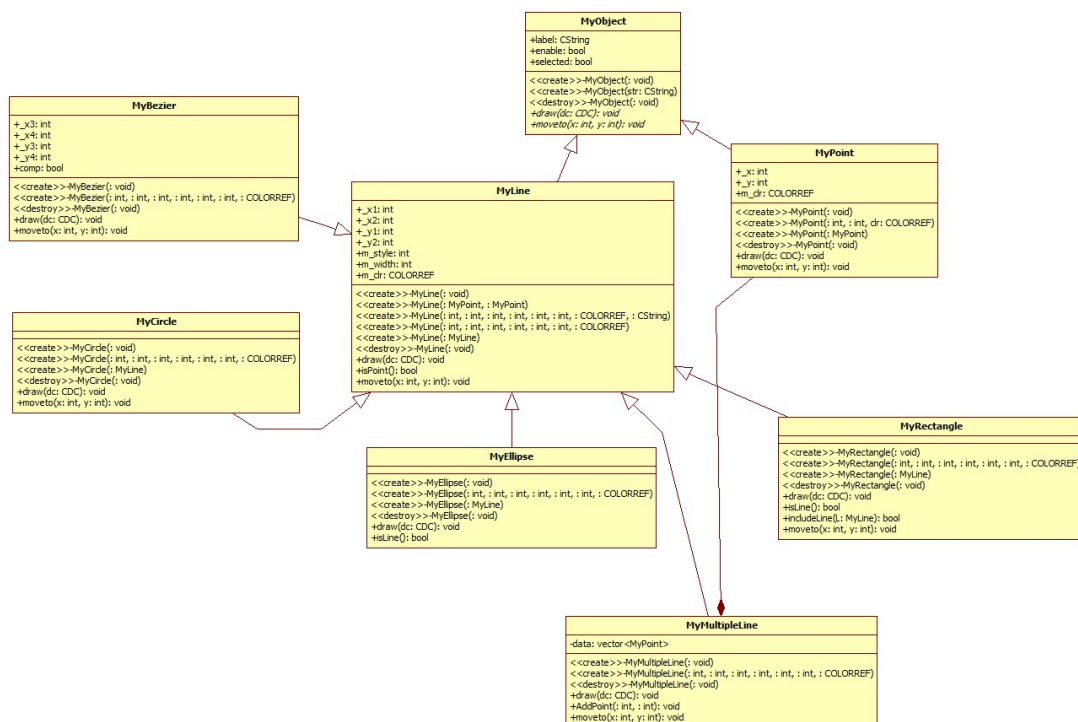
以我们的作业成品为例解释一下这个架构：



其中有一个主框架管理若干个子框架，同时也管理两个工具条，并且将工具条的状态储存在主框架中。

另外，主框架还管理文档，文档的新建由主框架负责，而关闭则由子框架来负责。文档和视图之间还存在密切的交互，视图类负责添加或更改文档类的数据，并且将这些数据显示出来。

2.1.2 管理绘制对象的类



本程序中，管理绘制对象的类的设计是一个重点，我使用了一个称为 MyObject 的虚基类，MyObject 有一个虚方法 Draw，在 MyObject 中并没有给出

实现，而是在其每个派生类中有不同的实现。

MyObject 有一个派生类 **MyLine**，而在 CDC 绘图的过程中，矩形，椭圆和圆都是通过存储两个点的坐标来实现的，所以我们用 **MyLine** 类派生出了 **MyRectangle**，**MyEllipse**，**MyCircle** 等类，此外还从 **MyLine** 派生出了多段线（**MyMultipleLine**）和贝塞尔样条曲线（**MyBeizer**）。他们都有不同的 **draw** 虚方法以及虚析构函数，所以我只需要以一个存储 **MyObject*** 的 **vector** 容器就可以完成它们的储存、显示和删除。

2.1.3 系统与数据库的交互

Adomdb
+m_pConnection: _ConnectionPtr +m_pRecordset: _RecordsetPtr +m_pCommand: _CommandPtr
<<create>>-Adomdb(: void) <<destroy>>-Adomdb(: void) +OnInitADOConn(: CString): void +ExitConnect(: void): void +NewUser(name: CString, password: CString): bool +NewFile(username: CString, filename: CString, sudo: CString, group: CString): bool +CheckFileState(username: CString, filepath: CString): int +NewFileToGroup(username: CString, filename: CString, sudo: CString, group: CString): bool

系统和数据库的交互我们通过一个自己设计的 **Adomdb** 类实现，这个类有三个数据成员，可以完成数据的各类操作，我们封装了一些常用的功能，例如 **OnInitADOConn(CString)**，可以打开数据库中的一个表。**ExitConnect** 则是断开连接，**NewUser** 和 **NewFile** 是向数据库中新加用户数据和文件数据。然而为了更好的拓展性以及使这个类看上去不太冗杂，我们也没有完全把数据成员封装为 **private**，其他类也可以调用，对列表进行遍历等。

2.2 小组分工

小组成员姓名	小组成员班级	小组成员学号	小组成员分工
夏斐	自22	2012011417	<ul style="list-style-type: none"> ● 各类对象的绘制与管理，类的设计 ● 图片的载入与保存 ● 用户登录注册界面的编写 ● 绘制工具栏的编写 ● 命令行界面的编写，带有“编辑框重载功能”的工具条的编写 ● 主体框架的搭建，主框架、子框架、应用程序间的数据交互，文档类与视图类的设计与维护 ● 用户权限和授权管理模块的编写 ● 其他拓展功能的编写
王婉秋	自22	2012011412	<ul style="list-style-type: none"> ● 系统和数据库的连接 ● 图片的保存时与数据库的交互 ● 图片打开时与数据库的交互 ● 方便易用的数据库交互类的编写 ● 文件管理模块，文件搜索，按照权限和按照文件名搜索文件，图片分组功能的设计和实现 ● 命令行翻译工具的编写

3 本人工作内容

3.1 主体框架的搭建

3.1.1 主框架的构建

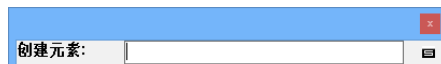
主框架需要提供很多功能，首先需要提供一个能够进行绘图的绘图工具栏，其次还需要提供能够进行文件管理的相应菜单。我们在主框架中存储了当前绘图的状态，包括当前绘图使用的工具，线宽，线型，颜色等。



如上图所示的工具条可以实现各种功能的切换以及线型和颜色的选择。

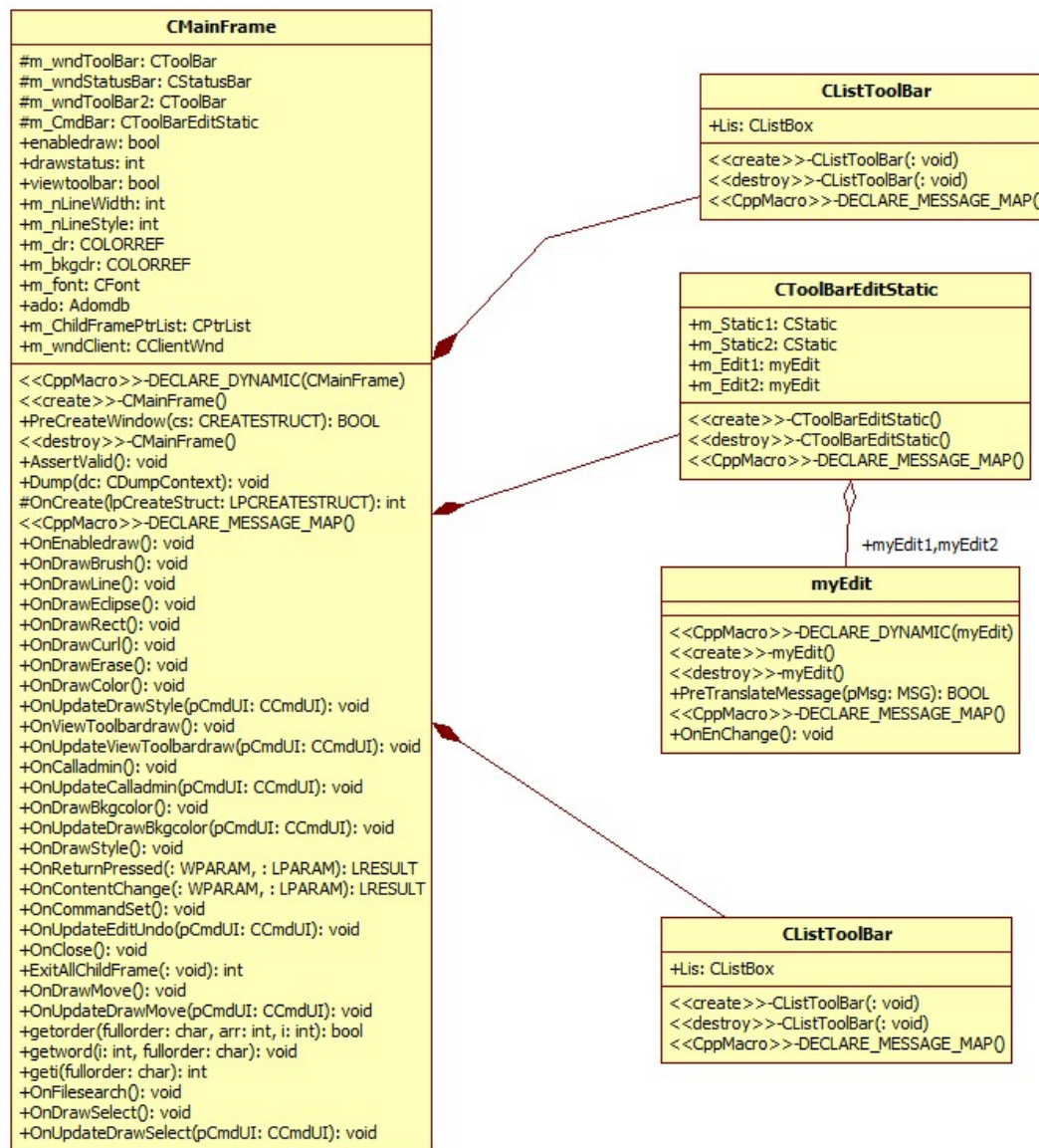


系统默认的文件工具条可以实现文件的新建，打开载入和保存。



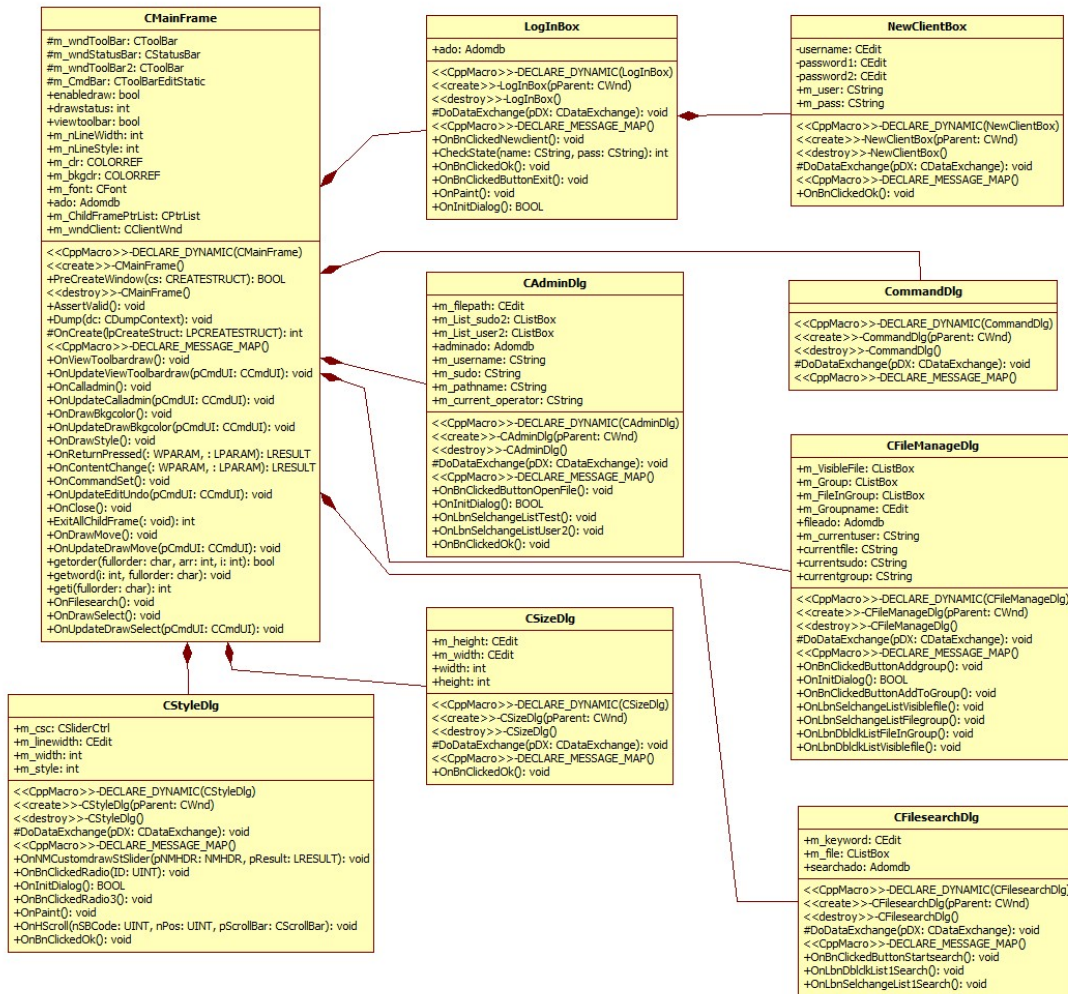
绘制状态的各个变量都保存在 CMainFrm 里，是否开启绘图保存在 enabledraw 里，绘图状态保存在 m_drawstatus 里，颜色存在 m_clr 里，线型线宽分别保存在 m_nLineStyle 里和 m_nLinewidth 里。

3.1.2 主框架和工具栏及对话框之间的交互

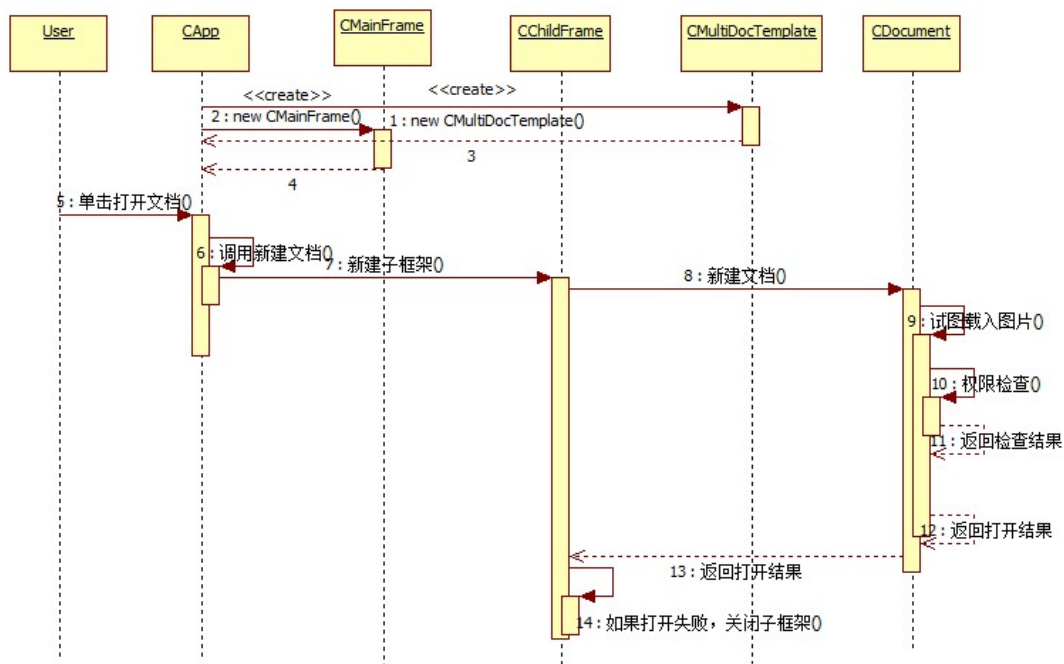


主框架还负责维护状态栏和工具栏，负责更新工具栏的状态并且响应工具栏的消息。主框架中存有当前绘图状态的变量，是否允许绘图，选择的是什么工具，当前的线型和颜色是什么。

各个子功能大都是通过对话框来实现的，对话框为主框架或者应用类的成员，其关系如图所示。



3.1.3 文档与视图的创建和维护

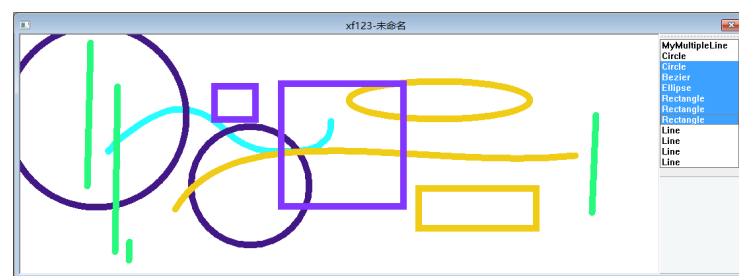


文档类和视图类是程序的重要部分，文档类完成文档的打开和保存，数据的存储，数据的串行化（由于这次作业是要求保存 bmp，所以没有用到这个功能，保存直接在视图类中实现了），视图类负责从文档类中读取数据并显示，并且接收用户输入，将数据存储到文档里。图示了一个文件打开的典型流程，打开时先创建一个临时的文档，并且利用文档对图片进行载入，如果载入成功，则保留文档，如果载入失败，则将文档关闭。

3.2 绘图模块的编写

3.2.1 绘图对象的设计

如图所示是我们实现的绘图效果。



户移动鼠标时，改变 tmp 的状态，并且刷新重绘，用户抬起鼠标时，将 tmp 存到对应的文档类里，并且将 tmp 置为 NULL。在重绘函数里，我们从文档类读出当前的容器，依次将每个元素画出来，并且将 tmp 画出来。

3.2.3 对象的选定与橡皮擦工具

上述所指的绘图功能仅限于直线、矩形、圆和椭圆，有另外五种绘图工具和它们的原理有比较大的差别。

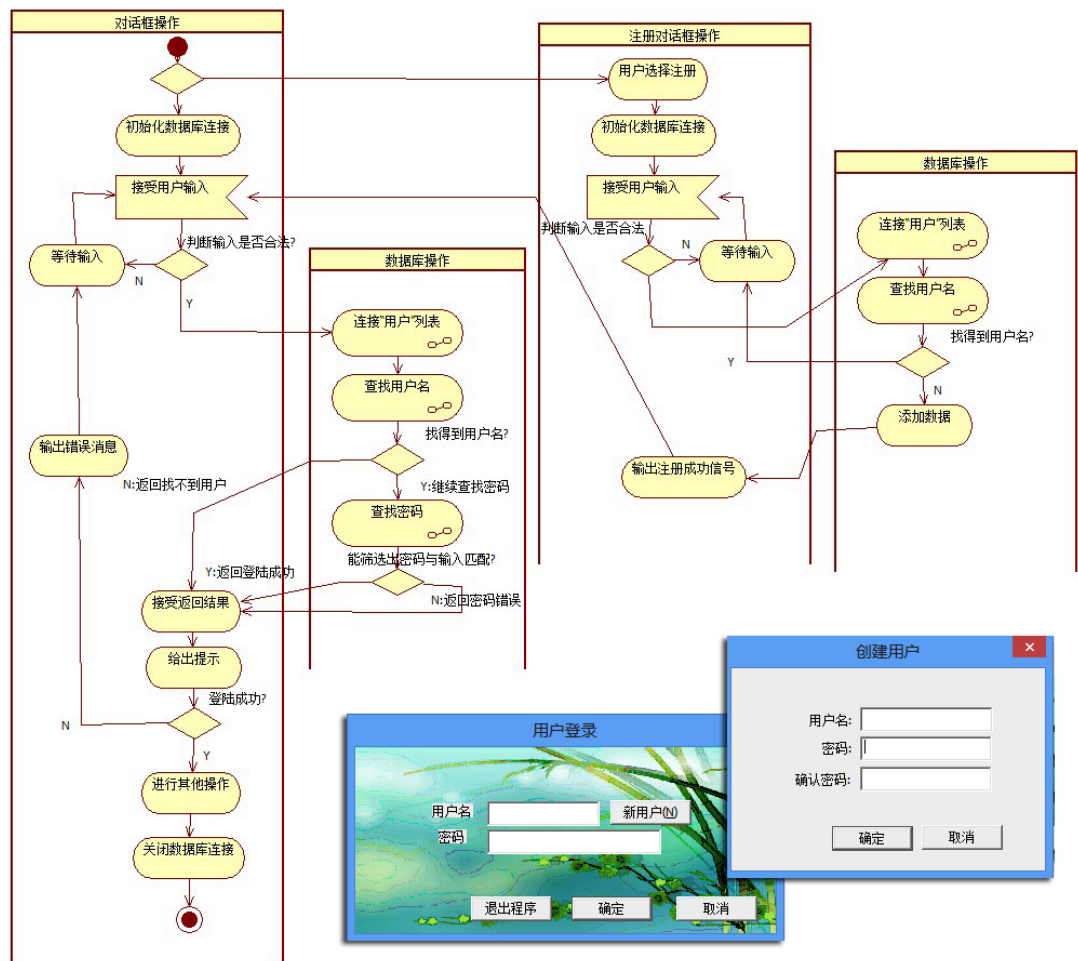
对象的选定工具和橡皮擦工具比较特殊，但原理相近，当鼠标按下时，就创建一个虚线组成的矩形，当鼠标移动时更新矩形的状态，鼠标抬起时，判断对象容器中有多少对象被矩形框包裹，就把它们进行选定或者删除，并且将矩形框删除。

多段线的实现原理是在点击时创建一个多段线对象，在鼠标移动时将路径上的点都加入这个对象中，鼠标抬起时将对象放入容器。

贝塞尔曲线是相当于画两条线段，则构成了两个端点两个控制点，可以画出优美的样条曲线。

3.3 权限管理模块的编写

3.3.1 用户登录与注册



用户登录和注册的逻辑图如上所示，我通过两个对话框实现，在输入后进行数据检查并且利用另一位同学写好的数据库的类进行数据库的维护。新建用户时，直接调用 `NewUser(CString,CString)`，检查用户时，则是手动遍历用户列表。

LogInBox	NewClientBox
<code>+ado: Adomdb</code>	<code>-username: CEdit</code>
<code><<CppMacro>>-DECLARE_DYNAMIC(LogInBox)</code>	<code>-password1: CEdit</code>
<code><<create>>-LogInBox(pParent: CWnd)</code>	<code>-password2: CEdit</code>
<code><<destroy>>-LogInBox()</code>	<code>+m_user: CString</code>
<code>#DataExchange(pDX: CDataExchange): void</code>	<code>+m_pass: CString</code>
<code><<CppMacro>>-DECLARE_MESSAGE_MAP()</code>	<code><<CppMacro>>-DECLARE_DYNAMIC(NewClientBox)</code>
<code>+OnBnClickedNewclient(): void</code>	<code><<create>>-NewClientBox(pParent: CWnd)</code>
<code>+CheckState(name: CString, pass: CString): int</code>	<code><<destroy>>-NewClientBox()</code>
<code>+OnBnClickedOk(): void</code>	<code>#DataExchange(pDX: CDataExchange): void</code>
<code>+OnBnClickedButtonExit(): void</code>	<code><<CppMacro>>-DECLARE_MESSAGE_MAP()</code>
<code>+OnPaint(): void</code>	<code>+OnBnClickedOk(): void</code>
<code>+OnInitDialog(): BOOL</code>	

图中所示的是处理登录和注册界面的对话框的类。这里我们做了很多应对错

误输入的方法，例如注册新用户时会检查这个用户是否存在，避免创建同名用户，提高了程序的健壮性。

除此之外，我们还提供了注销登录的功能，注销登录要求关闭当前所有文档，方可注销成功。

3.3.2 文档存储与打开界面

文件储存和打开时与数据库进行交互，存文件时将该文件的创建者连同文件路径存入数据库，打开时也进行校验，如果能够在表中匹配到用户名和文件路径则可以打开，打开方式有只读和可写，如果匹配不到用户名和文件路径，则不允许打开该文件。

权限:表			
	用户名	文件路径	访问权限
▶	xf123	\\psf\Home\Desktop\1234. bmp	写
	xf123	\\psf\Home\Desktop\IMG_0901. jpg	写
	xf123	\\psf\Home\Desktop\8888. bmp	读
	xf123	\\psf\Home\Desktop\4567. bmp	写
	xf123	\\psf\Home\Desktop\4567. bmp	写
	xf123	\\psf\Home\Desktop\4567. bmp	写
	annaufife	\\psf\Home\Desktop\555. bmp	写
	xf123	\\psf\Home\Desktop\12354. bmp	写
	123	\\psf\Home\Desktop\43645. bmp	写
	xf123	\\psf\Home\Desktop\8888. bmp	写
	xf123	\\psf\Home\Desktop\8888. bmp	写
	xf123	\\psf\Home\Desktop\8888. bmp	写
	xf123	\\psf\Home\Desktop\8888. bmp	写
	xf123	\\psf\Home\Desktop\8888. bmp	写

除了选择打开，还有另外一种方式，单击打开旁边的按钮，是向一个空白文档中载入一张图片，在进行载入时同样会进行权限检查，不可以将某一张库中的图片载入，也就是说，不能将任何一张由用户创建的图片载入工作区，如果当前文档不为空，则不允许载入。

```
if(m_srcimg==NULL) m_srcimg=new CImage();
else m_srcimg->Destroy(); //释放以前的图片
m_srcimg->Load(str);

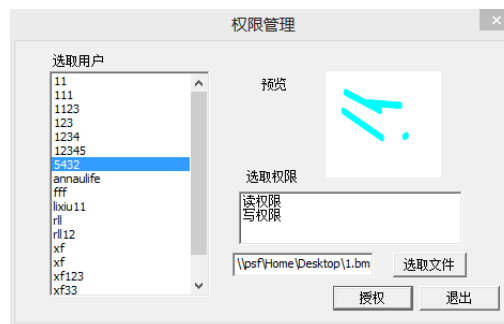
if(m_img==NULL) m_img=new CImage();
else m_img->Destroy(); //释放以前的图片

if ((m_srcimg->GetHeight()>800)|| (m_srcimg->GetWidth()>1200))
{
    AfxMessageBox("您载入的图片太大，已经做了相应缩放");
    int tempheight = 600;
    int tempwidth = tempheight*m_srcimg->GetWidth()/m_srcimg->GetHeight();
    if (tempwidth>1200)
    {
        tempheight = 600*1200/tempwidth;
        tempwidth =1200;
    }
    CreateStretchedImage(m_srcimg,m_img,tempheight,tempwidth);
}
else
    ImageCopy(*m_srcimg,*m_img);
```

无论是图片的打开还是图片的保存都分两步实现，一是将图片载入 m_srcimg，代表原图片，第二是检查 m_srcimg 的尺寸，如果图片过大，则进行压缩并且将缩放后的图片载入 m_img。m_img 是在视图类中绘制在界面上的图片。

文件的存储步骤大致如下，保存时会先检查文档类的一个数据成员 `m_path_name` 是否已经有值，如果已经有值说明文件已经被保存过，就直接保存到这个路径，如果没有值就说明是第一次保存，需要用户选择路径，为了实现多个用户可以存同名文件，我们在用户的文件名之后加入了用户名作为后缀。

3.3.3 授权管理



在权限管理对话框中，对话框和数据库交互，获得当前所有用户的名单，并且可以点击选取文件按钮来选择一个文件，并且选取一种权限。点击授权则会完成授权。这里做了一些容错性处理，例如如果用户选择了自己也没有写权限的图片则无法完成授权，如果有某一项没有选则也会发出警告。并且，改对话框还很人性化地提供了预览功能。这里我们也采取了很多提高程序健壮性的措施，例如一个用户不可以给自己授权。

3.4 其他拓展功能的实现

3.4.1 选定闪烁功能

在选定是我能够提供闪烁效果，这是通过定时器来实现的。我设置了一个布尔变量，每隔 500ms 翻转一次。如果这个布尔值为真，就显示选定对象，否则不显示，这样就实现了选定对象闪烁的功能

```
afx_msg void OnTimer(UINT_PTR nIDEvent);
```

3.4.2 shift 正交功能

使用了响应键盘事件，如果 shift 键按下，就将 `normalize` 设置为真，在绘制时，在移动鼠标时对坐标的修改有限制，只能画出正方形，圆形以及水平竖直直线。具体参见代码。

3.4.2 右侧列表功能

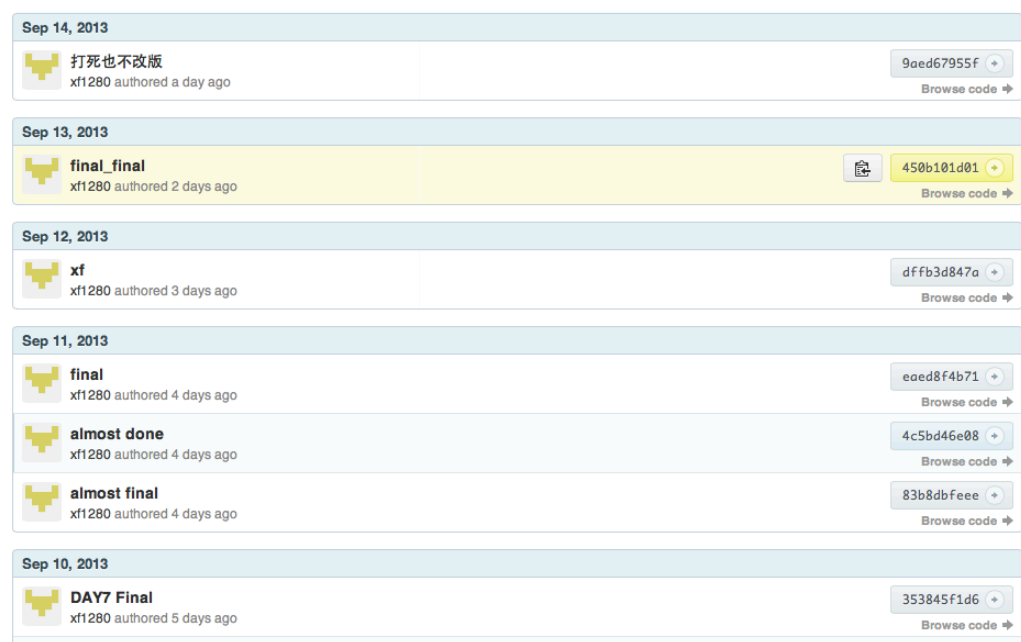
我在 ChildFrame 中加入了嵌入 listbox 的工具栏，用来显示对象列表，从而实现了非常难以实现的多选对象功能，实现起来非常复杂，从略。

4 项目总结

4.1 文件的版本管理

为了多人协作共同编写这个程序，我们使用了 Github 作为工程的版本管理，该软件可以保存历史版本，而且可以做多人开发的融合等，为我们的开发提供了很多方便，也减少了工程融合时遇到的问题。短短一周之内，共有 22 次提交，见证了这个小软件的发展。

此项目在 Github 上的地址是 [Github.com/Eureka22/PicManage](https://github.com/Eureka22/PicManage)



4.2 关于数据成员存放的位置

程序中存在很多数据成员，我仔细考虑了应该将这些数据成员存放的位置。实际上，几乎从每个类中都能获取到任何想要的指针，所以数据存放在哪里并无紧要，甚至有同学直接将数据存在视图类里。但我认为这是不科学的，程序设计应该符合“紧内聚、松耦合”的原则，不应该频繁地从一个类调用另外一个类的方法，或者修改其数据成员。所以我将登录状态和登录操作交给应用类来处理，绘图的工具和状态采用主框架类来处理，绘图采用视图类来处理，数据交给文档

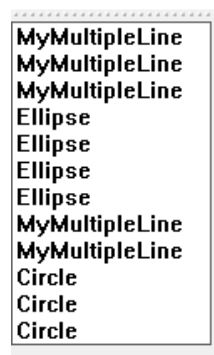
类来处理，这样一来，各个模块之间的相互依赖就变少了，每个类负责各自的功能，根据功能来划分，也符合面向对象程序设计的理念。在完成了整体设计，实现了大部分功能之后我们进行了进一步封装，对于一些只有本类需要用到的数据成员和成员函数我们用了私有类型来处理，这样不会暴露太多接口。

4.3 工具条与控件

为了实现更多功能，我们采用了多文档来实现本次大作业，而多文档用到了文档-视图结构的较多内容，学习和使用起来比较复杂。我也广泛查阅了资料，解决了一些问题，其中一个问题就是如何在工具栏中嵌入控件，这是一个对话框程序不会遇到的问题，因为对话框中可以直接添加控件，而对于多文档的主框架，则比较难在其中加入控件，所以我们采用了一种折衷的办法，我们派生了 `CToolBar` 类，此时我惊喜地发现，`CToolBar` 类是从 `CWnd` 类派生而来的，既然我们能在 `CWnd` 类中增加控件，在 `CToolBar` 类中也一定可以。我做了很多次尝试，用空按钮将工具条挤出空当，在上面创建控件，自己写消息响应函数。这是我创建的包含编辑框的工具条：



这是我创建的带有 `Listbox` 的工具条：



而关于如何在工具栏中增加 `listbox`，我完全没有找到任何资料，这是一项**完全原创的创新**。而在创建了 `listbox` 之后，我还让它能控制对象的选中，并且可以响应对象的变化，例如增加或者删除对象，则列表里的元素则会相应增加或者减少。这里我借鉴了多文档多视图的理念，比起维护一个视图，我觉得维护一个工具条中的一个控件更加简单而且能实现更多功能。这个小创造这让我觉得举一反三很重要，在使用一个现成的工具前深入了解其原理更重要。

4.4 改善用户体验是无止境的

本作业在刚开始做的时候，就决定做成基于多文档的，所以打算实现很多功能，在实现了基本功能和几乎所有拓展功能之后，又开发了许多功能。在这个过程中，我们一方面努力提高用户体验，另一方面减少 `bug`（这也是提高用户体验

的一个方面)，我将我们组编写的程序给一些同学体验，让他们提出意见，然后我们再试图改进，并借此体验复杂软件开发的整个过程。

一开始，我们做的软件实现移动对象非常笨拙，是将最后绘制的对象吸附到鼠标上，并且进行移动，这样一来实现的功能就不够强大。

作业中要求能够将对象进行移动，那么究竟是移动哪个对象呢？选定对象成为了本程序的一个难点。我想到我可以用类似于橡皮擦的形式实现对象的选定，这就又出现了一个问题，如何表示被选定的对象呢？最后，我使用了一个定时器，来实现了被选定对象的闪烁，使选定和移动功能方便易用。

还有一个典型的例子，我们在要绘制圆形和正方形时，用椭圆工具和矩形工具比较难画准，这是我借鉴了大部分软件的方式，即按住 `shift` 开启正交化，响应 `WM_KEYDOWN`，实现了在按住 `shift` 的时候自动绘制圆形、正方形和水平、竖直的线段。

再比如说在搜索文件对话框里，实现了双击文件目录中的文件就可以直接打开，虽是比较细节的地方，但也能提高用户体验。

以上说的只是简单的几个方面，我们在这些方面还在很多细节上做了改进。在这个开发的过程中，我认识到提升用户体验的过程是永无止境的，永远可以有需要做的东西。

5 相关问题的说明

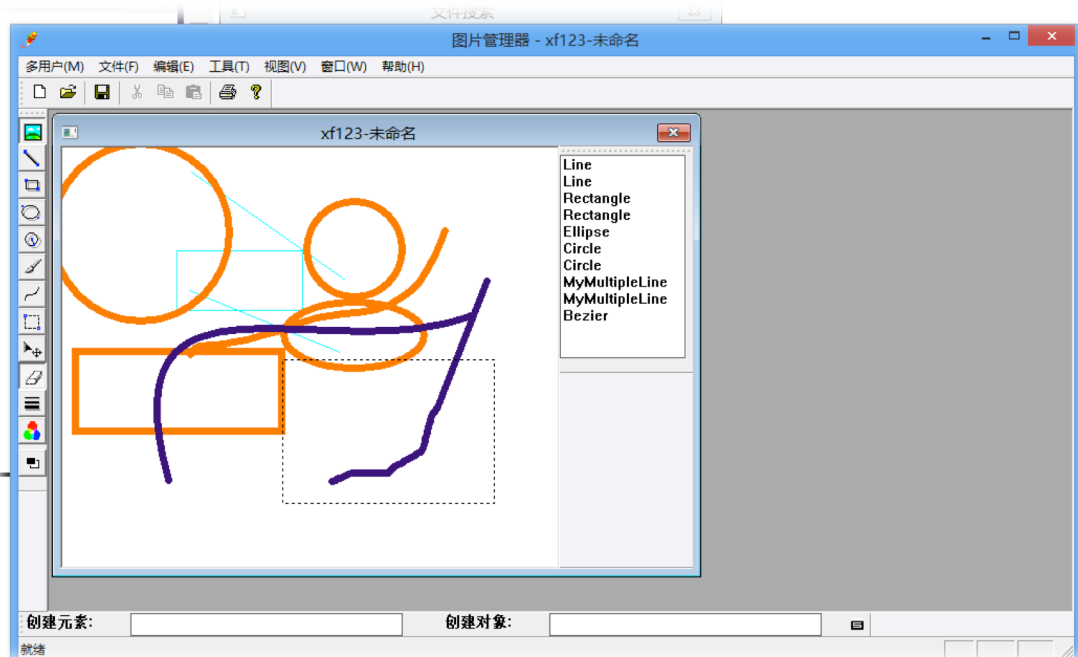
5.1 数据库的配置

文件运行需要提供一个数据库，直接使用 `Release` 文件夹下的数据库即可，数据库中的表已经编辑好，程序运行时将这两个文件存放在一个文件夹下即可。

5.2 更多拓展功能的说明（操作实例）

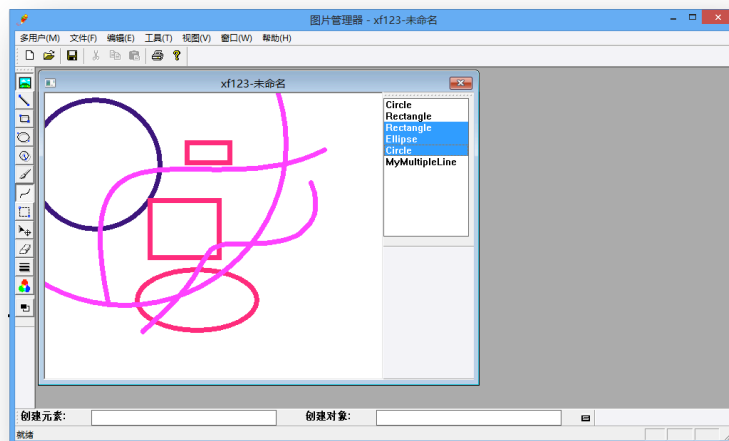
除了大作业要求中的功能，本系统一是实现了很多额外的功能，二是在提高用户体验方面做了很多努力，下面就次做简要说明。

5.2.1 对象橡皮擦与对象选定

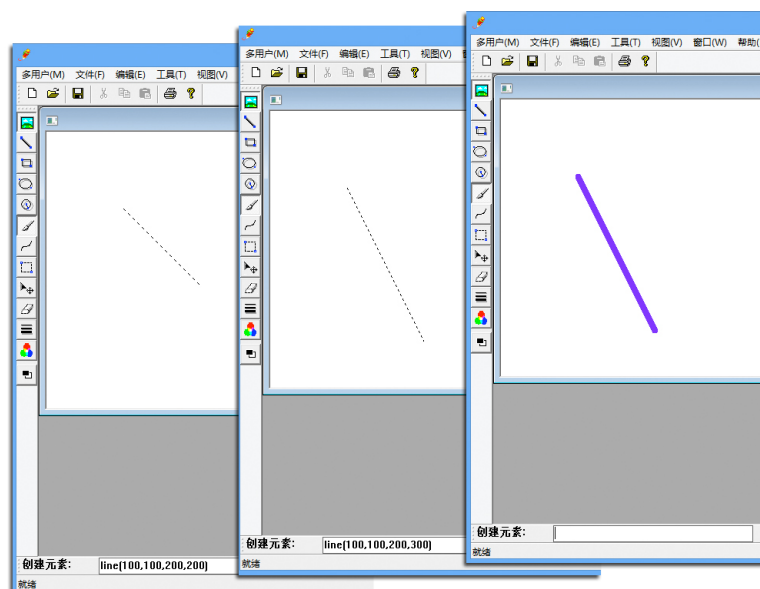


本项目中的橡皮擦为对象橡皮擦，只能擦除整个对象，使用时点选橡皮擦工具，可以拖出一个框，所有在框中额物体将会被删除掉。

对象的选定可以通过两种方法实现。一是选中每个子框架右边的对象栏，可以支持多选，被选中的对象将会在屏幕上闪烁，被选定的对象可以移动。另一种方法就是和橡皮类似，使用选择工具可以在屏幕上拖出一个框，所有框中的物体将会被选中，可以连续选择。完成拖动，或者进行了绘制操作，则会取消选定。

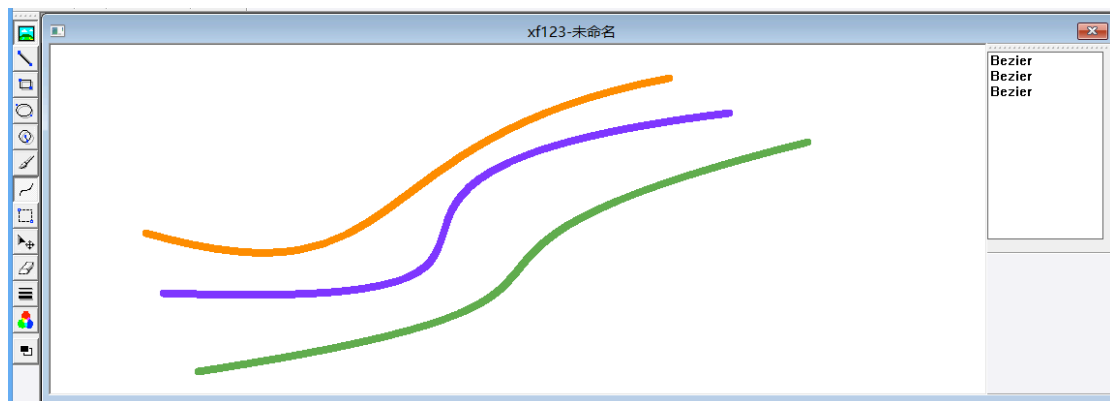


5.2.2 命令行的其他功能



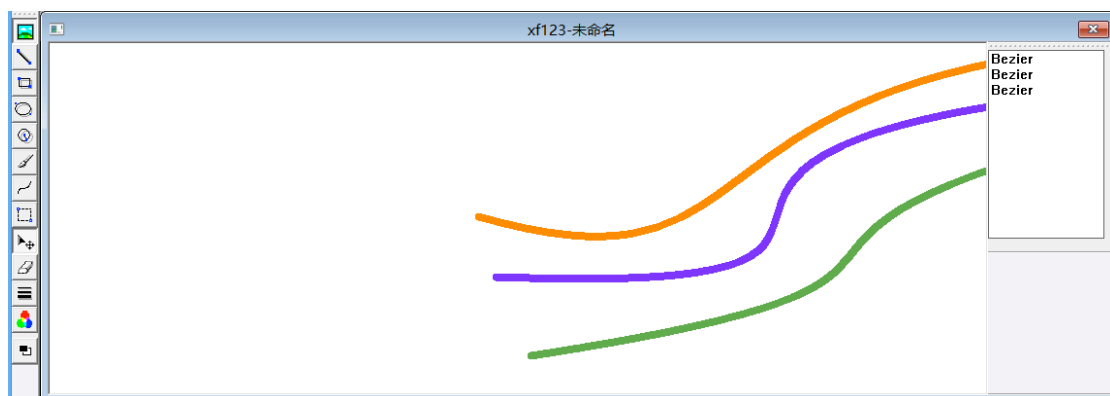
考虑到用户对于坐标不太熟悉，我们在输命令的时候实现了预览功能，即输入命令后并不会在屏幕上画线，而是绘制一条虚线，示意画出来的物体将会出现在这个位置，当用户真正敲击回车时，对象才会被绘制上去。敲击回车之前，用户可以更改坐标，也完全可以取消绘制。实现方面，我们借助了 **tmp** 这个指针，即一开始就把它放在 **tmp** 里，直到敲击回车才真正放入文档里。这里我们发现用 **tmp** 做缓冲是一个很好的机制。

5.2.3 贝塞尔曲线的绘制



用户可以在界面上绘制贝塞尔曲线，第一次画的直线固定了两个端点，第二次绘制的是两个控制点，由于要先固定一个控制点拖动另外一个控制点，所以对不熟悉样条曲线的人来说操作起来稍显复杂而不精确。

5.2.4 对象的移动

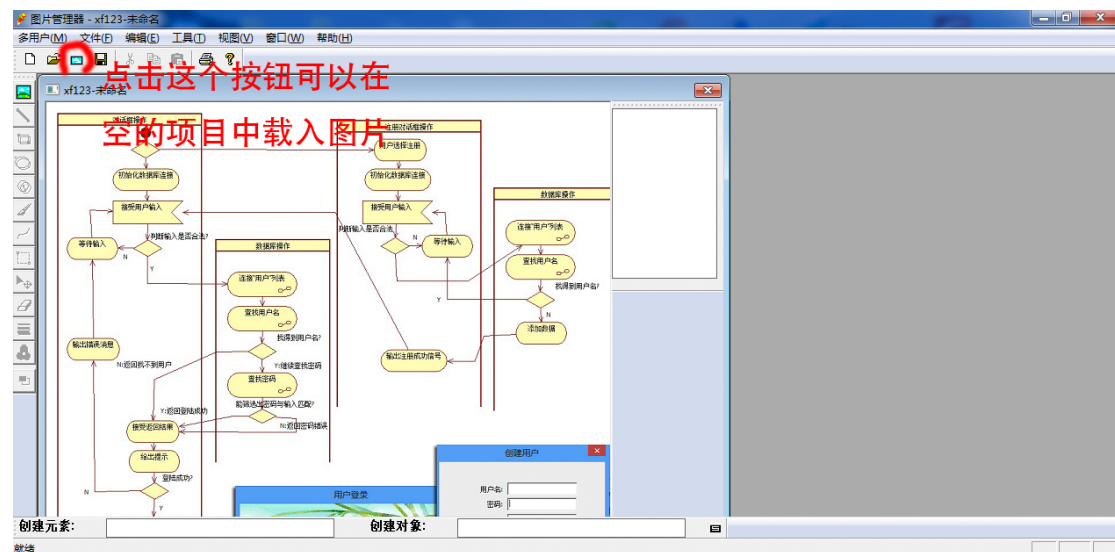


一旦选定了对象，可以使用移动工具进行移动，这里用到了对象的 `moveto` 虚函数，是运行时多态的一个例子。

5.2.5 对象列表功能

本程序提供了对象列表功能，在视图的右侧有一个工具栏，内嵌一个 `listbox`，里面列出了已经画出的所有对象，可以在其中进行选定。这里其实是对拓展功能的一个扩充，作业中有一个要求，用一句命令画出的称为一个对象，可以对对象进行移动，那么在我们的程序中，所有一次性绘出的东西都当成了对象，可以进行选定，删除和移动。

5.2.6 图片的载入



点击载入图片按钮可以在当前项目中载入图片，如果图片过大会进行一些缩放。

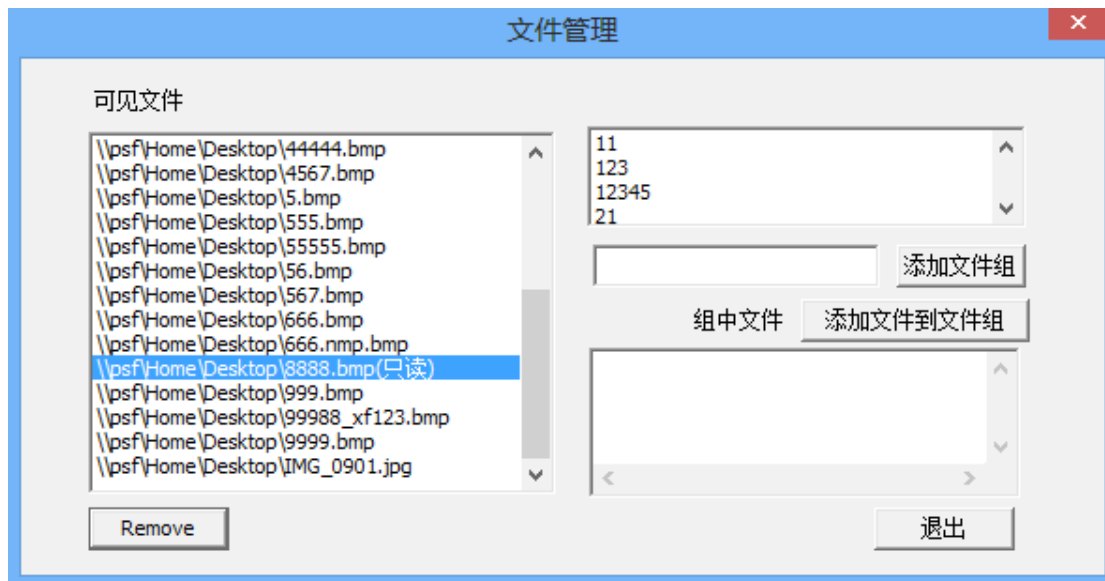
5.2.7 绘图过程中的快捷键

绘图过程中可以采用快捷键，按住 shift 可以开启“正交化”，即绘制椭圆只能绘制出圆，绘制矩形只能绘制出正方形，绘制线段只能和坐标轴垂直或者平行。

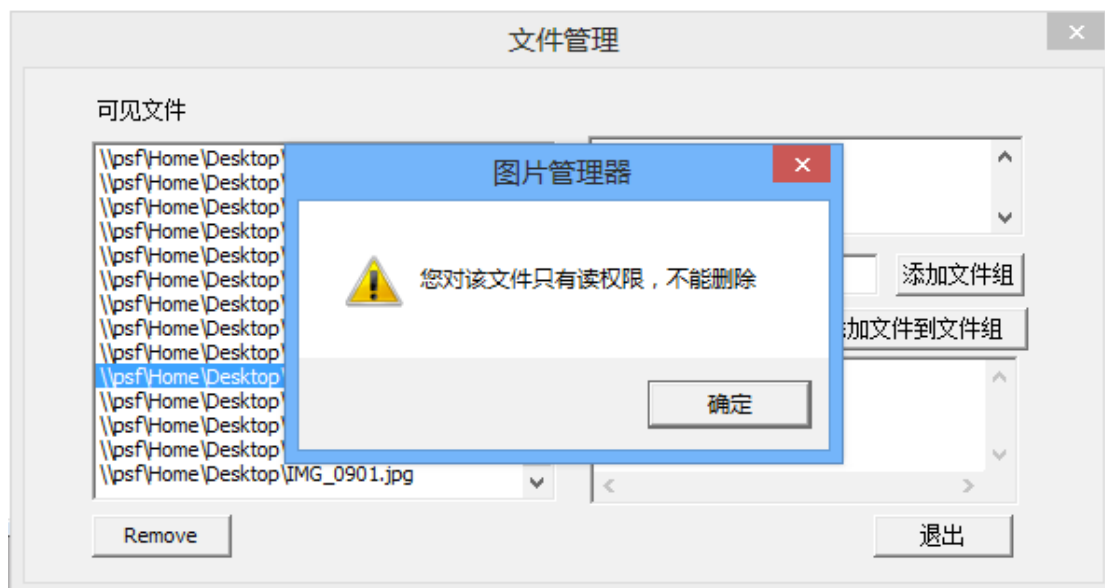


绘制过程中可以使用 Ctrl+Z 来撤销上一步绘制。

5.2.8 文件的删除



选中文件名点 **remove** 即可删除，如果不符合条件，则会不允许删除。



5.2.9 更改画布大小



用户可以通过更改画布大小命令来改变绘图区域的大小。