**开发日志**

**作者：李华**

**组长：张同学**

**指导老师：王老师**

**小组成员：张同学 赵同学 李华**

目录

[CDraw类及其相关类的设计 5](#_Toc104241329)

[设计CDraw 类 5](#_Toc104241330)

[开发目的 5](#_Toc104241331)

[具体步骤 5](#_Toc104241332)

[开发时间 6](#_Toc104241333)

[设计CClicle CPLine 类 6](#_Toc104241334)

[开发目的 6](#_Toc104241335)

[具体步骤 6](#_Toc104241336)

[开发时间 7](#_Toc104241337)

[设计CText 类 7](#_Toc104241338)

[开发目的 7](#_Toc104241339)

[具体步骤 8](#_Toc104241340)

[开发时间 9](#_Toc104241341)

[系统文档的设计 10](#_Toc104241342)

[开发目的 10](#_Toc104241343)

[具体步骤 10](#_Toc104241344)

[开发时间 11](#_Toc104241345)

[矢量图形的绘制 11](#_Toc104241346)

[颜色框的设计 11](#_Toc104241347)

[开发目的 12](#_Toc104241348)

[具体步骤 12](#_Toc104241349)

[开发时间 13](#_Toc104241350)

[绘制矢量图形 13](#_Toc104241351)

[开发目的 14](#_Toc104241352)

[具体步骤 14](#_Toc104241353)

[开发时间 14](#_Toc104241354)

[重绘 15](#_Toc104241355)

[开发目的 15](#_Toc104241356)

[具体步骤 15](#_Toc104241357)

[开发时间 15](#_Toc104241358)

[图形的基本操作 16](#_Toc104241359)

[开发目的 16](#_Toc104241360)

[具体步骤 16](#_Toc104241361)

[开发时间 18](#_Toc104241362)

[拉框 18](#_Toc104241363)

[开发目的 18](#_Toc104241364)

[开发时间 18](#_Toc104241365)

[矢量图形的存取 18](#_Toc104241366)

[开发目的 18](#_Toc104241367)

[具体步骤 19](#_Toc104241368)

[开发时间 19](#_Toc104241369)

[矢量图形的选择 19](#_Toc104241370)

[开发目的 19](#_Toc104241371)

[具体步骤 20](#_Toc104241372)

[开发时间 22](#_Toc104241373)

[底图的相关功能 22](#_Toc104241374)

[开发目的 22](#_Toc104241375)

[具体步骤 22](#_Toc104241376)

[开发时间 24](#_Toc104241377)

[缓冲区的设计 24](#_Toc104241378)

[开发时间 24](#_Toc104241379)

[缓冲区 25](#_Toc104241380)

[开发目的 25](#_Toc104241381)

[具体步骤 25](#_Toc104241382)

[缓冲带设计 25](#_Toc104241383)

[开发目的 25](#_Toc104241384)

[具体步骤 25](#_Toc104241385)

[批量缓冲 26](#_Toc104241386)

[开发目的 26](#_Toc104241387)

[具体步骤 26](#_Toc104241388)

# CDraw类及其相关类的设计

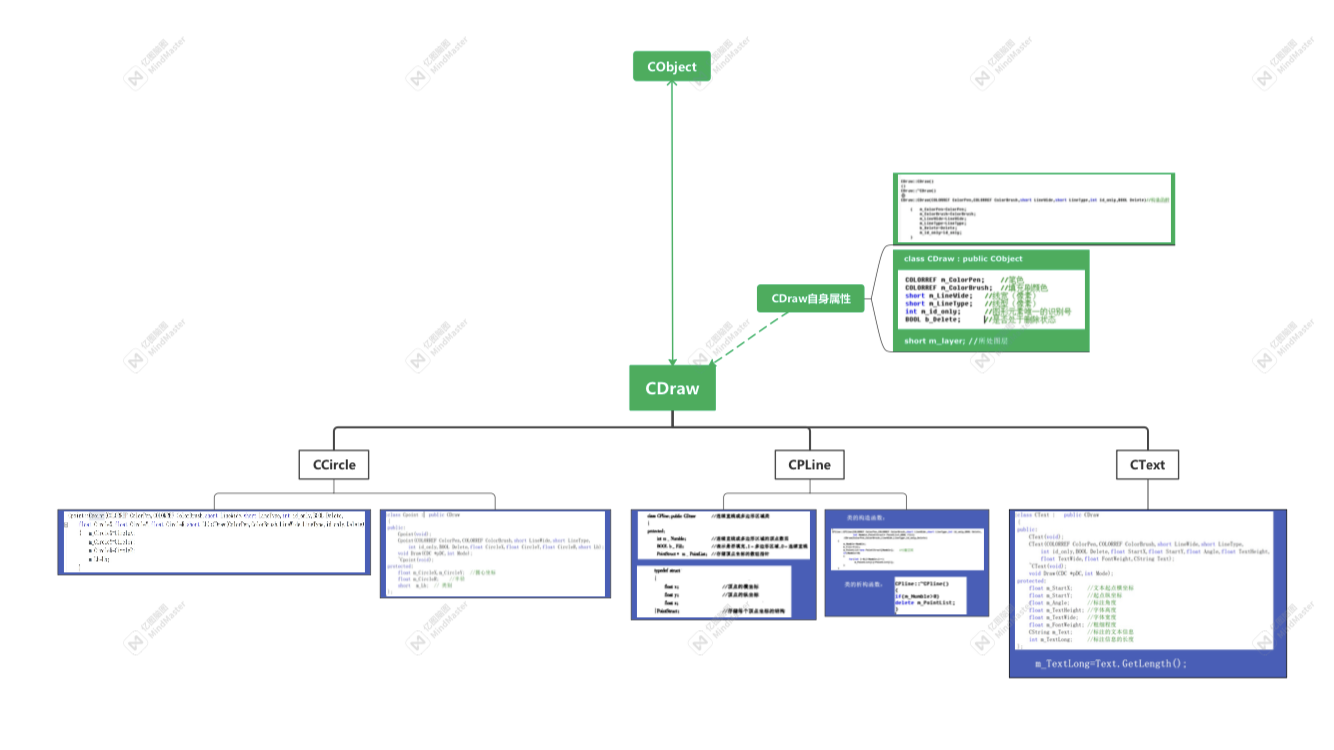
## 设计CDraw 类

### 开发目的

建立工程，搭建工程框架，创建点类CDraw，添加相应变量，为日后的功能开发做准备。

### 具体步骤

当日向工程中加入图形元素基类**CDraw**，以**CObject**类作为基类，添加变量**m\_ColorPen**（笔色），**m\_ColorBrush**（填充刷颜色），**m\_LineWide**(线宽)，**m\_LineType**（线型），**m\_id\_only**（图像元素的唯一识别号），**b\_Delete**（是否处于删除状态），**m\_layer**（所处图层），添加构造函数。



### 开发时间

2022/2/29

## 设计CClicle CPLine 类

### 开发目的

添加折线和多边形类CPline以及文本类CText，在两个类中分别添加相应变量，为图形的绘制做准备。

### 具体步骤

1.向工程中加入折线和多边形类CPline，以CDraw类为基类，添加变量m\_Numble（连续直线或多边形区域的顶点数目），b\_Fill（表示是否填充），m\_PointList（存储顶点的数组指针），添加构造函数。

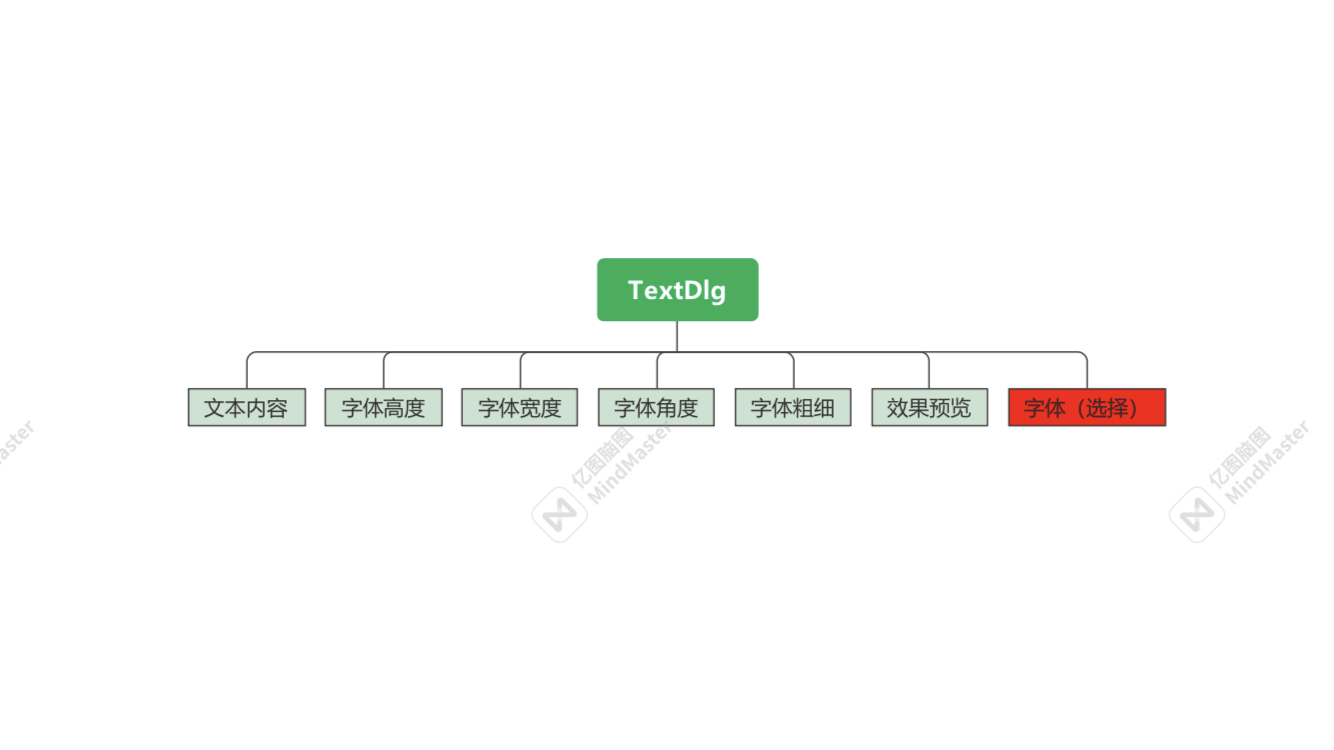
2.向工程中添加标注文本类CText，以CDraw类为基类，添加变量m\_StartX（文本起点坐标），m\_StartY(起点纵坐标)，m\_Angle(标注角度)，m\_TextHeight（文字高度），m\_TextWide(文字宽度)，m\_FontWeight（粗细程度），m\_Text(标注的文本信息)，m\_TextLong(标注信息的长度)

3.向工程中添加点类Cpoint，以CDraw类作为基类，添加变量m\_CircleX，m\_CircleY（圆心坐标），m\_CircleR(半径)，添加构造函数。

### 开发时间

2022/3/3

## 设计CText 类

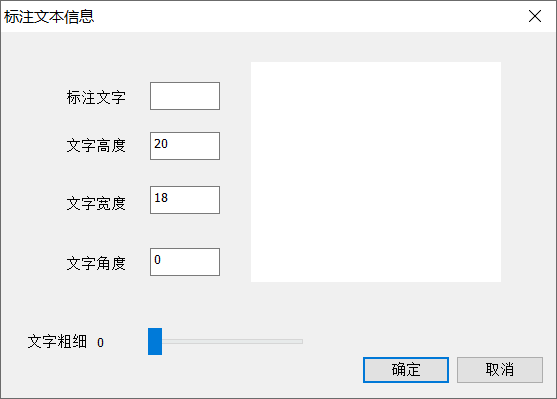


### 开发目的

增加调整文本样式的对话框，同时利用滑块控制文字的粗细。在调整文本样式的对话框中增加大小适中的矩形，使设置的文字样式可以在该矩形框中进行预览。

### 具体步骤

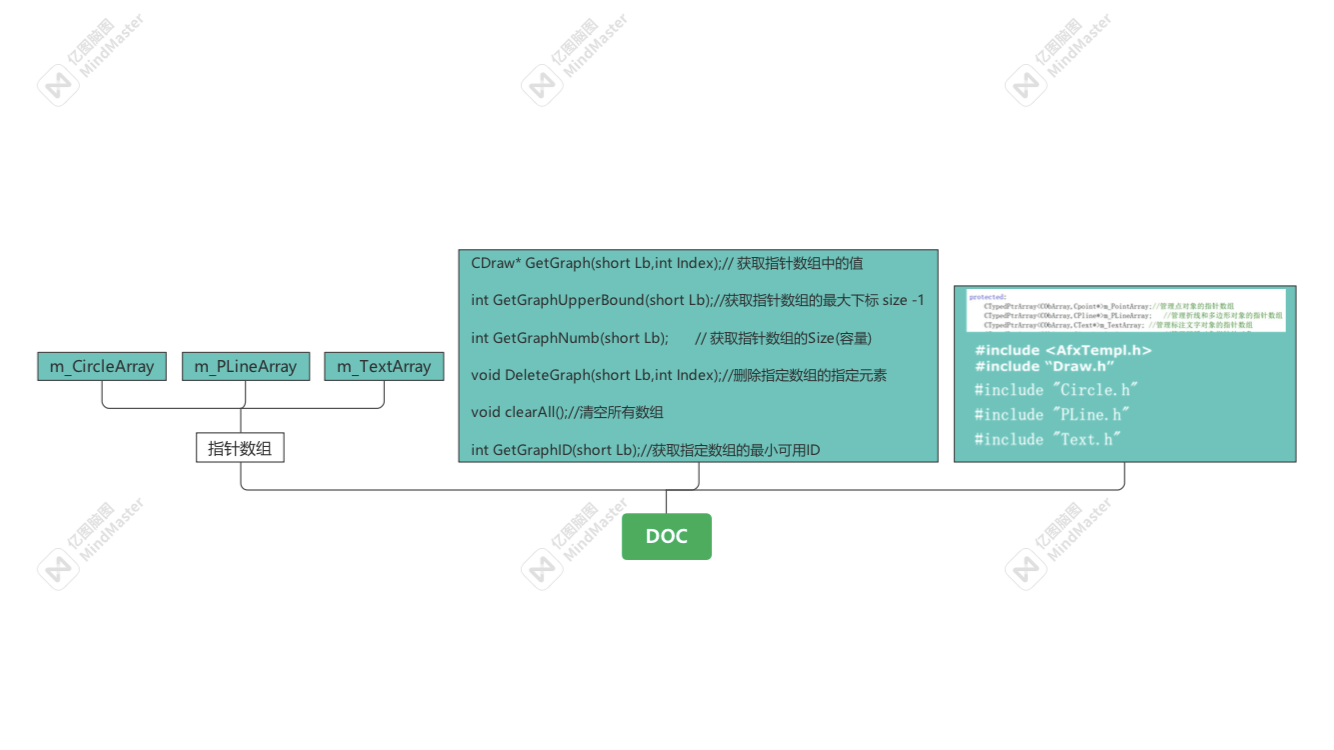
1. 在资源视图中，添加了一个Dialog对话框，并为其添加了一个类CTextDlg,并为其添加四个文本框，分别设置文字的内容，高度，宽度，角度，并分别为其添加对应的变量，以及一个滑块用来调正文字的粗细。并添加一个OnInitDialog()来设置滑块的上限和下限，并添加滑块事件以控制文字粗细。
2. 在CText类中，添加Draw函数，进行参数的各种运算，使之符合设置文字所需的数学格式，定义一个CFont类型的变量，并把各种参数传递给CFont对象，使之能够画出来符合已设参数的字。
3. 在View视图中，添加OnLButtonDown()函数，设置一个左键点击屏幕的事件，使之弹出对话框CTextDlg,并传递文字高度，宽度，粗细等参数，然后调用文档类中的AddText函数，来添加一个Text元素，并调用其Draw函数。
4. 打开工程文件→资源视图→Dialog→双击打开上次定义的对话框，右键滑块控件→属性→控件事件，添加NM\_CUSTOMDRAW事件，点击进入相应函数。



### 开发时间

2022/3/7 - 2022/3/9

# 系统文档的设计



### 开发目的

在菜单中增加相应的绘图选项，同时设置相应ID；在文档类中添加指针数组，便于存储各类图形，并添加相应函数，为将来的系统功能做充足准备。

### 具体步骤

1.在资源视图的Menu中，增设了“鼠标绘图”子菜单：点（包括圆形和三角形）、圆、直线、多边形、标注文字，并分别设置它们的ID。

2.在View视图中分别为各个选项添加响应函数，以及设置两个变量，PushNum用来记录按下鼠标的次数，m\_DrawCurrent表示当前正在进行的序号，点（包括圆形和三角形、直线、多边形、标注文字。

3.在文档类中添加头文件#include <AfxTempl.h> #include “Draw.h”以及点、线、文字三个类的头文件，添加三个指针数组分别为m\_PointArray；m\_PLineArray；m\_TextArray;分别管理点、线、文字。

4.在文档类中添加增加

①增加图形元素的函数，分别为AddPLine，AddPoint ，AddText;

②返回类型分别为点、线、文字的类的指针。

③返回指向图形元素的指针的函数：GetGraph;

④返回图形元素数目的函数：GetGraphNumb

⑤获得ID的函数:GetGraphID （要预先在文档类中添加一个m\_Index[ ]数组，以及在CDraw类中定义一个可以返回图形元素ID的函数GetID）

⑥清空图形元素的函数:clea

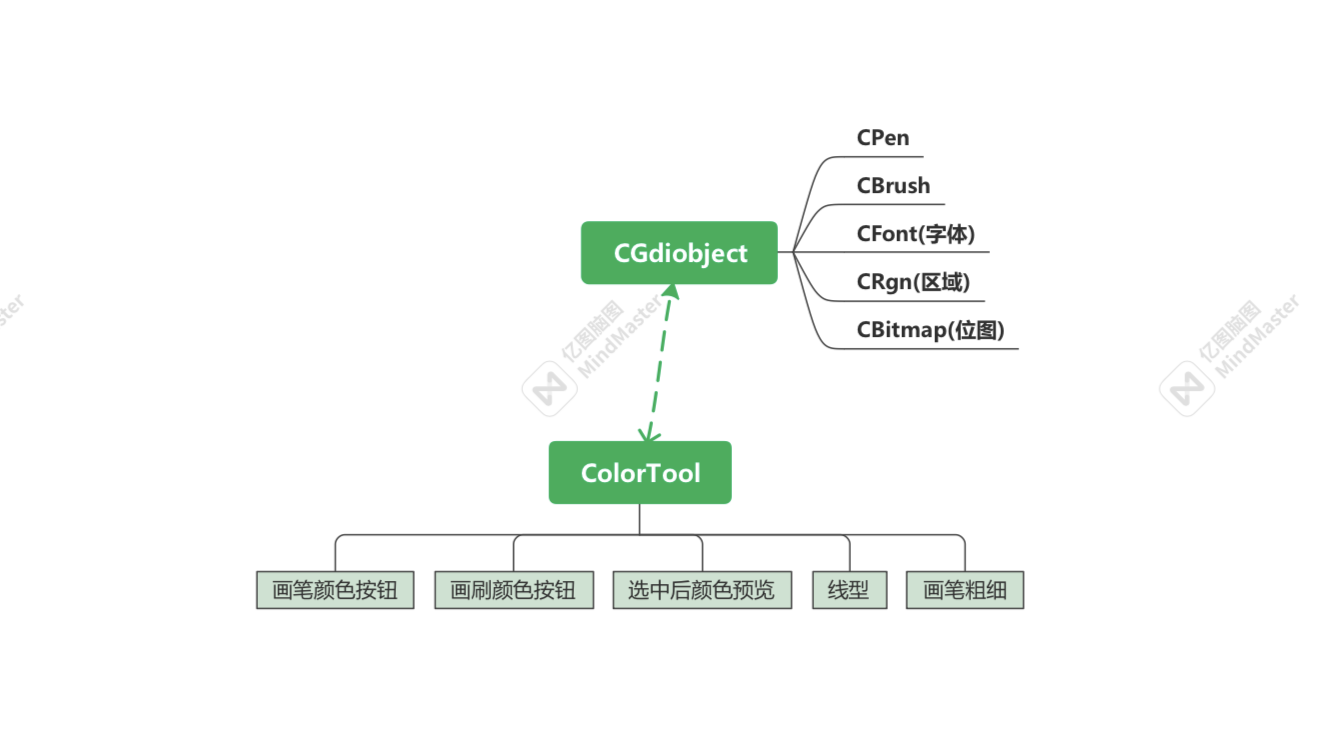
⑦删除某一图形元素的函数：DeleteGraph

### 开发时间

2022/3/10 - 2022/3-11

# 矢量图形的绘制

## 颜色框的设计



### 开发目的

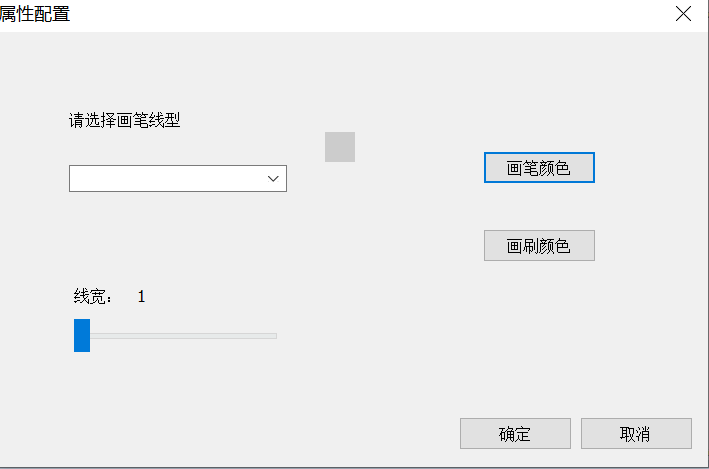
添加制图工具对话框，实现对线型、画笔、画刷、粗细等的统一设置。

### 具体步骤

1.在菜单中添加“绘制工具”选项，在“鼠标绘图”下添加即可，添加后更改选项ID为“Tool”，点击类视图，右键视图类→类向导→添加ID为Tool的消息处理函数OnTool()；

2.点击资源视图，添加对话框资源，右键新建的对话框→添加类，基类为CDialogEx，

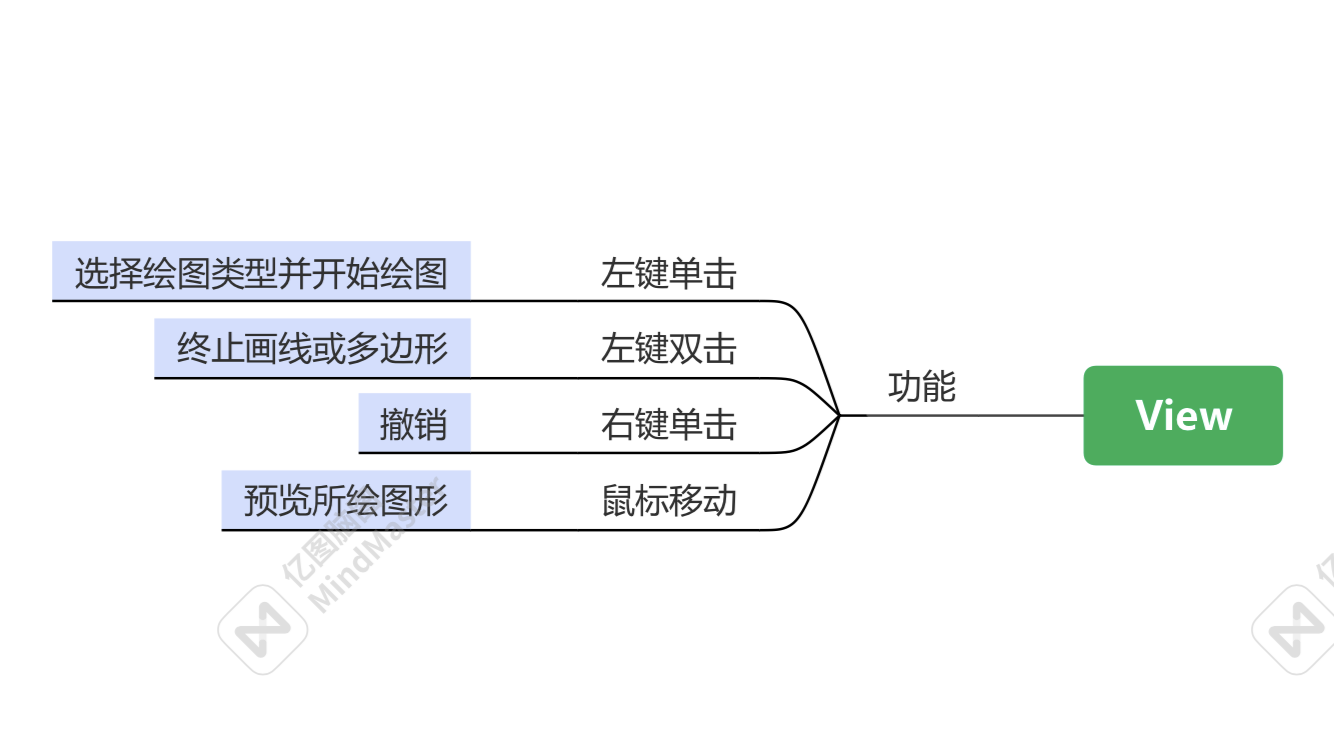
3.为对话框添加相应的控件，控件的布局如下图所示：



### 开发时间

2022/3/17

## 绘制矢量图形



### 开发目的

绘制完整的圆、线和多边形，显示绘制轨迹，并实现鼠标右键取消

### 具体步骤

1.在视图类中添加鼠标移动响应函数OnMouseMove( )

2. 在CPline类里添加Draw()函数，该函数时为了用数组存储直线和多边形的点以及区分直线不封口和多边形封口

3. 在视图类中鼠标移动响应函数OnMouseMove( )

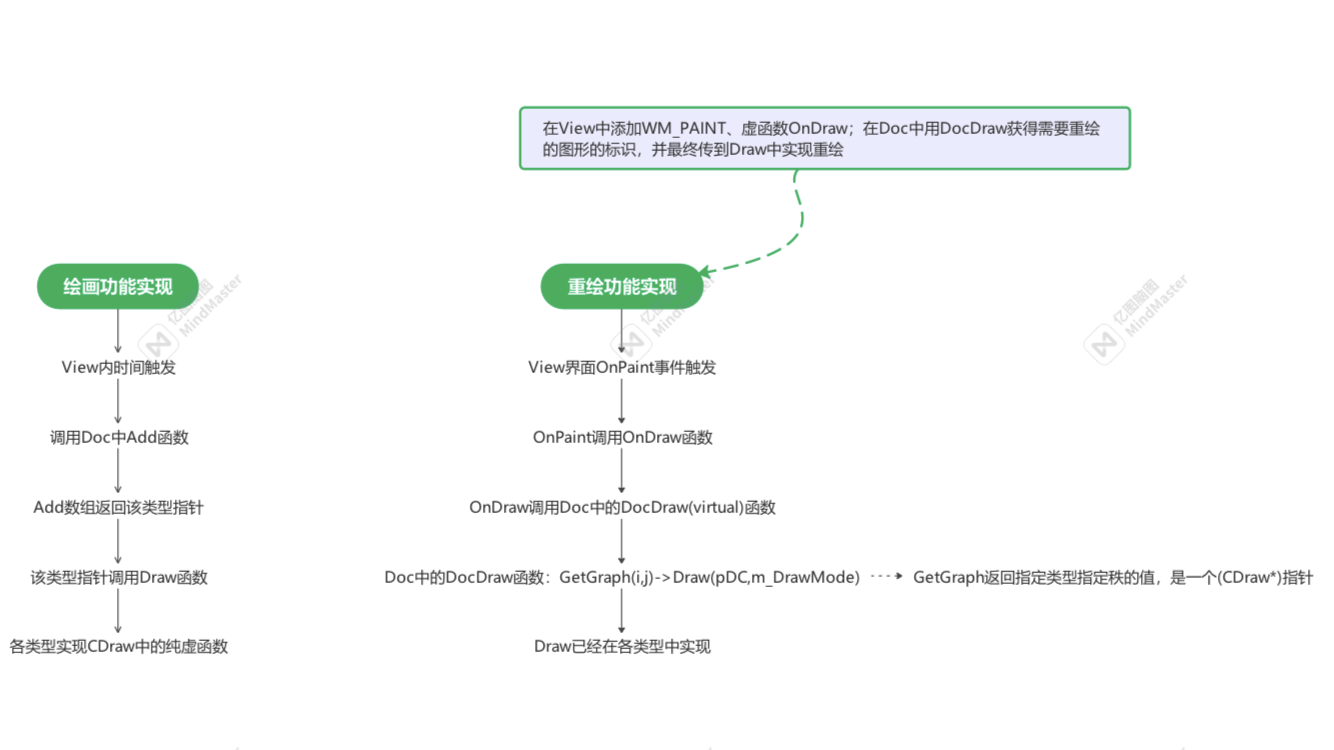
4.在视图类中添加鼠标双击响应函数OnLButtonDblClk(),实现双击鼠标结束画图

5.在视图类中添加鼠标右键按下响应函数OnRButtonDown()

### 开发时间

2022/3/14 – 2022/3/17

## 重绘



### 开发目的

实现图形的重绘，保证在窗口最小化和移动到屏幕边时，原有的图形不消失。

### 具体步骤

1.在文档类的头文件中添加纯虚函数Draw()

2.在视图类cpp文件中的Draw()函数中实现图形的重绘，确保窗口缩小或者移动到屏幕边后，在返回时原来绘制的图形还在

### 开发时间

2022/3/14 – 2022/3/16

# 图形的基本操作

### 开发目的

放大图形（二倍、三倍）。

缩小图形（二倍、三倍）。

平移。

全图。

放大、缩小（框选）。

### 具体步骤

①放大：

在鼠标左键按下事件里：判断m\_DrawCurrent是否等于给定值，并且将point值分别赋予mPointOrign、mPointOld。此时PushNumb等于0，放大的原理其实就是令PushNumb自加使其非0，从而在判断语句中对图形的每一个坐标的每一个x/y值进行相应倍数的修改，使其通过坐标点坐标值扩大的方式来使图形放大。获取到所拉矩形边界的x、y的最大值和最小值，根据横向和纵向对比例尺分别进行放大计算，计算完成后将逻辑坐标转为实际坐标，取较大的比例尺进行重绘。

结束后PushNumb归0.

②缩小：

利用DPtoVP将实际原点坐标转为逻辑坐标形式，比例尺放大二倍，利用屏幕缩小四分之一圈计算现在的原点坐标，并转为实际坐标，重绘图形。三倍缩小同理。

③平移：

将m\_DrawCurrent赋予对应平移的数字，按照格式书写平移所需语句。

④全图：

1.在view.cpp的全图显示按键下添加代码，分别设置x最小值、x最大值、y最小值、y最大值，初始化时将非常大的数赋予最小值，将非常小的数赋予最大值。按照横向x和纵向y重新计算出比例尺bl1和bl2，取bl1和bl2中的较大值作为比例尺，并将minx和miny作为原点坐标，进行重绘。

2.在文档类的cpp中创建GetRect函数，来获取封装边界。进行循环嵌套，外循环代表分别在点类、线类、文字类中获取图形，内循环指在每个类中针对每个图形，获取到所有图形中x的最小值最大值，y的最小值最大值。

3.分别在三个类中编写GetRect函数，获取到封装边界的x、y的最大最小值。

4.在CDraw类的头文件里添加纯虚函数GetRect。

5.在view.cpp的重绘按键下添加代码，将存储的临时原点坐标（原本的实际原点坐标）重新赋给m\_xStart,m\_yStart,临时比例尺重新赋给blc，并进行重绘。

⑤放大缩小（框选）

把当前屏幕显示内容在所选框中显示，逻辑和ArcGIS类似。

### 开发时间

2022/3/14

## 拉框

### 开发目的

使用CrectTracker实现拉框，左键按下开是拉框，鼠标释放显示

### 开发时间

2022/5/15

# 矢量图形的存取

### 开发目的

数组（Array）：一个能动态减小增大的元素列表.

列表（List）: 能够动态减小增大，并允许插入元素的有序元素列表。

映射（Map）:一个把关键字映射到值的“字典”

等基于模版的类，利用不同形式，对输入或者编辑的信息进行路径分析。

### 具体步骤

多路径分析：

1.利用CMap类完成路径存储查询。

2.视图类中添加成员变量，建立映射。

3.在视图类中建立一个BOOL型的变量，用来控制菜单中的开始结束绘制的黑灰变化。利用BOOL型变量的初始值是真或者假来改变。

4.在onLButtonDown函数下的m\_DrawCurrent赋予对应路径分析的数字来执行代码。

5.需要给路径进添加重绘。

### 开发时间

2022/4/16 – 2022/4/18

# 矢量图形的选择

### 开发目的

矢量图形的选择、删除与恢复

点选、框选、减选、取消选择。

图形的删除、取消删除、清空回收站。

缓冲区的绘制。

### 具体步骤

1. 点选

1.在文档头文件中添加一个结构GraphSelectStruct,包含图形的类别和编号两个要素。然后添加两个变量，一个记录选中图形元素的数目，一个存储选中元素的性质，并在构函以及析函中进行初始化以及删除。

2.在文档的cpp文件中，添加一个存储选中图形的数组AddSelectList，并在文档的Draw中添加一个重绘选中元素的循环语句。

3.在view视图的左键按下中，添加该执行语句。调用判断点是否点在图形元素上的函数，若在则选中高亮。

4.在点（圆），线（多边形），文本的文件中添加IsPoint函数——判断点是否点在图形元素上。

5.在Draw文件中添加IsPoint的纯虚函数（多态性）。

②框选

1.在view视图的左键按下中，添加框选所执行的语句。调用判断矩形框是否有与图形元素相交的函数，若有则选中高亮。

2.在点（圆），线（多边形），文本的文件中添加IsRect函数——判断所拉的矩形框是否有与其相交的图形元素。

3.在Draw文件中添加IsRect的纯虚函数（多态性）。

③减选

在view视图的左键按下中，添加框选所执行的语句。调用判断矩形框是否有与图形元素相交的函数，若有则取消高亮。

在文档的cpp文件中，添加一个存储选中图形的数组DSelectList，并在文档的Draw中添加一个重绘选中元素的循环语句。

在view视图的左键按下和鼠标移动实现弹簧矩形中，添加该执行语句。调用判断点是否点在图形元素上的函数，若在则取消高亮。

④取消选择

为取消选择菜单添加处理函数，将选中图形元素数量的变量置为0（n\_GraphSelect=0），并添加重绘函数，即可达到全部取消选择的功能。

⑤图形的删除

为删除菜单添加处理函数，遍历选中的图形元素，再调用Delete函数，最后调用重绘函数清空屏幕中选中的元素，达到暂时删除的目的。

⑥取消删除

传入Delete函数的参数为0。将删除掉的图形恢复。

⑦清空回收站

将删除的图形彻底删除。

1.在文档中建立一个Pack函数，按类别和id进行双层循环，将每个选中的元素都进行删除。

2.添加处理函数，调用文档中Pack函数，目的是将存储在数组中的图形元素移除，并将记录选中的图形元素数据置为0，调用重绘函数，清空屏幕。

### 开发时间

2022/4/20 – 2022/4/28

# 底图的相关功能

### 开发目的

完善底图的相关操作，为综合性分析创造前提，加载或平移、移除底图，并在图上进行定位及参数的设置。然后重绘及保存出图。

### 具体步骤

1. 加载底图

1.添加底图操作菜单设计，添加全局变量：CString name; BOOL b=false;

在文档类的头文件里添加成员变量及成员函数。

2.在Doc.cpp加载底图ID\_MAP\_ADD的执行函数 OnMapadd（）。

3.视图类添加成员变量及成员函数。

4.确定底图位置。

1. 移除底图

将加载好的底图移除。

1. 平移底图

和前文的平移合并为一个，方法如下：

1.拖动平移:在onLButtonDown函数下m\_DrawCurrent平移代码数字。

2.鼠标拉线平移：在OnLButtonDown中找到m\_DrawCurrent，让其对应另一个数字，在第二次按下鼠标左键中添加代码，将坐标转为实际坐标，然后改变xy进行判断，将实际坐标转为逻辑坐标。

1. 设置参数

1.输入边界

确定左上和右下的点，确定底图边界

2．图中取点

取两个已知坐标的点进行运算。

⑤保存出图

设置保存出图ID\_MAP\_PRINT，在view.cpp中添加相应函数OnMapPrint()，添加相应头文件写入代码实现保存功能。

1. 重绘

将已经设置好的数值再次赋予，做到遮盖后再次显示的功能。

⑦定位

1.新建dialog进行参数设置。

2.实现设置参数操作，设置全局变量判断是否设置参数。

3.查询定位，通过判断是否设置参数。输入想要查询的点的坐标进行定位查询，点并未真实生成，因此不能进行重绘，只是定义了用虚拟画布进行显示。

4.存档设置参数的比例尺。

5.显示经纬度，实现实际坐标转换为经纬度显示在状态栏。

6.修改测量函数。

### 开发时间

2022/5/2 – 2022/5/9

# 缓冲区的设计

### 开发时间

2022/5/10 – 2022/5/16

## 缓冲区

### 开发目的

实现对最后一个选中的图形进行缓冲区操作。

### 具体步骤

修改Draw函数的纯虚函数的参数值。

通过Dialog输入缓冲区大小级，将其传到数组中存起来，在点线draw函数的的函数中实现缓冲显示，设置画笔的宽度就是输入的缓冲区大小值。

## 缓冲带设计

### 开发目的

实现多环缓冲区

### 具体步骤

在GraphSelectStruct添加sumbuf层数，在Draw 中添加Mode等于3的情况，使用默认参数作为Draw,和缓冲区做法类似。

## 批量缓冲

### 开发目的

使相同的操作的缓冲区可以直接进行

### 具体步骤

在缓冲区设计时，将n\_GraphSelect-1调节为 i，进行for循环，使得所有被选中的内容全部进行多环缓冲。