### 矢量符号库设计文档

作者：熊晓乐

# 引言

作为形象化的语言，符号能够非常直观的想人们显示所要表达的内容。地图符号即地图语言，是直观表达地理事物和现象的一种重要的可视化工具。

目前，计算机制图中符号绘制（符号化）方法有2种，即信息法和编程法。

信息法也称符号库法，又可以进一步分为直接信息法和间接信息法，直接信息法直接保存图形的每个细部点，但是获得符号信息比较困难。间接法是存放图形的几何参数，然后根据相应的算法进行绘制。

编程法中每个符号或者同一类符号编制一个子程序，然后调用相应程序方法进行绘制，MICAPS矢量符号库采用的就是这种方法。

# 概述

MICAPS4.0符号库是在吸收前几版MICAPS符号库的开发和使用的基础上，进行全新的设计的符号库。重新处理输入设备传入的顶点，优化了渲染速度。

# 总体设计

## 符号库结构

每个符号库都有许多个符号组成，每个符号都有各自的特性，同时符号之间又有关联的地方。矢量符号库的设计思路如下图所示，

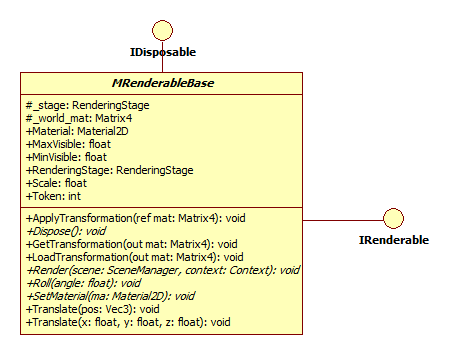


符号库大体分为3类：点符号、线符号和面符号。首先是点符号，点符号分为单纯的点和在某一指定点绘制指定的标签。其次是线符号，线符号比较复杂，分为单线和双线符号，其中单线符号包括单线箭头符号、单纯的单线、单线+符号（这里符号可以包括正方形、三角形、圆、标签、竖线等），双线符号包括双线带箭头符号（这里双线箭头又包含填充和不填充、带箭头和不带箭头，带尾部和不带尾部，带中间箭头和不带中间箭头以及这些类型之间组合形成的箭头符号）、单纯的双线、双线+符号（这里的符号类似“单线+符号”里的符号）。最后是面符号，面符号是根据点形成封闭的多边形，然后根据指定的方式进行填充，填充方法包括颜色填充、图案填充、符号填充这3种。

## 符号库设计

### 接口的定义

某些矢量符号或多或少存在一些相似的方法，此时可以定义一些接口规定子类行为，提高后续程序的可扩展性。如对基类MRenderableBase作如下设计：



基类继承了渲染接口IRenderable和资源释放接口IDisposeable接口。

### 基类的抽象

常用的矢量符号种类很多，其几何特征和内部结构特征千差万别，但是矢量符号的绘制过程大体类似。对于几乎全部的矢量符号的设计来说，其外部输入数据大体相同，只是对矢量符号的顶点、索引、材质等的处理设置不同而已，最后渲染的方式也是大体相同的。

因此，我们可以根据他们之间的共性抽出基类，后续开发其他类似的矢量符号时可以直接继承该基类，提高代码的简洁以及可读性。如基类MRenderableBase中的Render和Roll方法。

### 类的层**次结构的形成**

所有的矢量符号都继承基类的属性集与操作集，从而形成基类的子类。子类可以享用父类的公用属性字段，也可以重载父类的虚方法，重新编写方法从而达到子类的所需的要求。

### 类的聚合与复合类的形成

聚合的实质是使复合类从直线类等各类中吸收信息，并且可以定义自己的信息。此处的复合类是继承于基类后对各个基本符号进行聚合的。

### 资源的释放

设计基类时继承IDisposable接口，在子类中可以方便的重载Dispose方法进行非托管代码的释放。托管代码由垃圾回收机制（GC）自动回收处理。

### 符号库的外部调用

符号集的调用很方便也很简单，我们只需在其他项目工程中引用该动态链接库（dll），然后引用命名空间后就可以很方便调用里面的方法进行绘制矢量符号。当然在设计矢量符号绘制方法的时候为了方便外部调用给出了合理的默认值。

# 具体分析

## 点符号

点状符号表示地理现象。点状符号图形比较规则、固定，不随它在地图上的具体定位位置变化而改变。在地图上，点状符号都有确切的定位点和方向性。根据点状符号的制图特性，点状符号可进一步分为以下二级分类：

　　a　独立点状地物　包括点状地物符号、点状分布的地理现象、点状图面整饰符号。

　　b　地图注记　包括点状符号、线状符号、区域符号的地图注记以及文字型地图整饰。

MICAPS中的点符号大部分都是用来表示天气状况相关的信息，例如，晴天、雾、小雨、小雪、沙暴等等，还有一些标记的点状符号，例如修改标注、添加标签等。这些点符号大小是固定的，都是按照屏幕坐标系里的大小来衡量的，不会随着屏幕坐标的缩放而改变大小。

## 线符号

当地图符号所指代的概念在抽象意义下可认为是定位于几何上的线时，称为线状符号。这时，符号沿着某个方向延伸，且宽度与地图比例尺可以没有关系，而长度与地图比例尺发生关系。例如，等高线、等深线等均为线状符号。线状符号比较复杂，种类很多，在符号库结构中已经详细的介绍过，这里就不做重复。

线状符号的大体思路如下，

插值

输入点

根据规则创建索引

根据需要处理点

索引

线符号

顶点

插值点

渲染

### 单线符号

单线大多数继承与LabelLineSymbol类，而LabelLineSymbol又继承与LineSymbol类，LineSymbol类图如下



设计的LabelLineSymbol的类图如下图所示，



设计的所有的单线继承关系如下图所示



设计的单线大体上分为以上类图中这么多，下面对其进行具体的阐述（从FenLiuQu按照顺时针方向）。

#### 分流区

FenLiuQu（E:\workspace\m4\Bin\image\q_FenLiuQu.png）类表示的是一个折线箭头符号，是在单线符号的基础上重新准备顶点绘制而成的。相邻的折线锋值点间和峰值的距离是固定的，简单的说就是屏幕坐标上的距离是固定的。折线的峰值点数量是动态计算的，缩放屏幕坐标系时不改变形状大小，但是数量却会随着改变。

#### 霜冻线

FrostLineSymbol类表示的是一个锯齿状的符号（E:\workspace\m4\Bin\image\t_sdline.png）。霜冻线两头类似于‘凵’状的符号，而中间都是些‘|’状的符号。相邻两个‘|’状符号之间的距离是固定的，并且‘|’状符号的个数动态计算生成，缩放屏幕坐标系时不改变形状大小，但是数量却会随着改变。

#### 季节温度特征线

LineAndQuads类表示的是单线和方块的组合图形（E:\workspace\m4\Bin\image\q_WenDuTeZhengXian.png）。季节温度特征线中的方块数量会动态的计算，随着放大数量不断增多，随着缩小数量不断减少。

方块的边长是按照屏幕坐标衡量的，缩放屏幕坐标系时不改变形状大小，但是数量却会随着改变。

#### 干线

LineAndHalfCircle类表示的是单线和半圆的组合图形（E:\workspace\m4\Bin\image\q_GanXian.png）。干线的空心半圆是紧密相连的，上一个圆的最后一个点和下一个圆的第一个点挨着，空心半圆的半径是按屏幕坐标衡量的，它的数量动态计算，缩放屏幕坐标系时不改变半圆形状大小，但是数量却会随着改变。

#### 湿轴

ShiZhou类表示的是折线箭头（E:\workspace\m4\Bin\image\q_shizhou.png）。湿轴的设计跟分流区相似，详情请参考分流区。

#### 标签分割线

LabelSplitLine类表示的是标签分割单线的符号（ ）。设计标签分割线时让这个类实现两个功能，一种是标签分割单线，也就是说单线不是连续的，它被标签给分开，挖去了中间的若干个点。另一种是标签贴在单线上，标签并不切割单线，换种说法就是单线贯穿标签，单线是连续的，并没有被分割开。两种形式的标签的样式都可以随意更改的。缩放屏幕坐标系时不改变标签形状大小，但是数量却会随着改变。

10

#### 单线箭头

SingleArrowLineSymbol类表示的是单线箭头（E:\workspace\m4\Bin\image\Arrow_Solid.png ）。单线箭头就是普通的箭头符号，但是箭头的头部不会因为缩放而改变，在屏幕上永远是固定的大小。

#### 显著降温区

LineAndSoildCircle类表示的单线和实心圆的组合符号（E:\workspace\m4\Bin\image\q_JiangWenQu.png）。跟季节温度特征线类似，实心圆的数量会动态计算，缩放屏幕坐标系时不改变形状大小，但是数量却会随着改变。继承于LineAndSoildCircle的类有2个，分别是：

#### 未定义符号2

SoildLineAndCircle分割线和圆的组合符号（E:\workspace\m4\Bin\image\q_NoSet2.png），这里线的宽度比普通的单线宽，实心圆的数量也是动态计算的，相邻两个实心圆是等间距的。

#### 温度脊

SoildLineAndCircle由独立的实心圆形成的符号（E:\workspace\m4\Bin\image\q_WenDuJi.png），该符号只有实心圆，这些圆的实现跟父类一样。

#### 温度槽

LineAndTriangle类表示的是独立的三角形形成的符号（E:\workspace\m4\Bin\image\q_WenDuCao.png）。温度槽的设计跟SoildLineAndCircle的设计类似，只有三角形，并且三角形会动态的生成，实时的计算数量，缩放屏幕坐标系时不改变形状大小，但是数量却会随着改变。

#### 辐合线

LineAndCross类表示的是单线和‘×’的组合符号。跟SoildLineAndCircle的设计类似，‘×’符号的数量是动态计算的，只要单线的长度够固定距离的长度就会绘制‘×’符号，缩放屏幕坐标系时不改变形状大小，但是数量却会随着改变。

#### 干舌

GanShe类表示的是锯齿状符号（E:\workspace\m4\Bin\image\q_ShiShe.png）。跟霜冻线设计类似，但是干舌头部和尾部没有‘凵’状符号，实现更简单。‘|’状的符号和霜冻线设计一样，也是动态计算生成，缩放屏幕坐标系时不改变形状大小，但是数量却会随着改变。

#### 干侵入特征线

GanQinRu类表示的四边形和空圆的组合符号（E:\workspace\m4\Bin\image\q_GanQinRuTeZhengXian.png）。准确地说干侵入特征线属于双线符号，因为它由两条单线构成的，但是它没有继承于DoubleLineSymbol类，而是设计时让其继承于LabelLineSymbol，所以把干侵入特征线归为单线符号。四边形和空心圆的个数都是根据固定的屏幕距离动态计算生成的，同时缩放屏幕坐标时两者的形状大小都不会变形，但是两者的数量随着缩放却会改变。

#### 锋符号

设计的锋符号分为4类，分别为暖锋、冷锋、静止锋、锢囚锋， 它们之间的关系如下图所示，



这四种类型的锋符号设计时是继承于ISymbolAssembler接口，接口定义了实现符号时需要准备顶点的方法。下面依此对四种锋符号进行阐述分析，

暖锋

冷暖气团的交界面称为锋面。暖锋指锋面在移动的过程中，暖气团起主导作用，推动锋面向冷气团一侧移动，这种锋面称为暖锋。WarmFront类表示暖锋，它是的是单线和两个实心半圆组合的符号（E:\workspace\m4\Bin\image\q_nuanfeng.png）。两个半圆方向是相同的。从左向右画时半圆的方向朝上，从右向左画时半圆的方向朝下。两个半圆紧密的连接在一起，他们的形状相同，个数也相同并且个数是根据固定屏幕距离动态计算出来的，缩放屏幕坐标系时，半圆的形状大小并不改变。

冷锋

冷锋指锋面在移动的过程中，冷气团起主导作用，推动锋面向暖气团一侧移动，这种锋面称为冷锋。ColdFront类表示冷锋，它是单线和两个三角形组合的符号（E:\workspace\m4\Bin\image\q_LengFeng_Past12h.png）。冷锋的设计和暖锋的设计是一样的，只是把两个实心圆替换为两个三角形而已，其他属性特征类似，详情请参见暖锋。

静止锋

指当冷暖气团势力相当，锋面移动很少时，称为静止锋。一般把6h内锋面位置无大变化的锋定为静止锋。StationaryFront类表示静止锋，它是单线、实心半圆以及三角形的组合符号（E:\workspace\m4\Bin\image\q_jingzhifeng.png）。实心半圆和三角形的方向是相反的。从左向右画时，半圆的方向朝上，三角形的方向朝下，从右向左画时，半圆的方向朝下，三角形的方向朝上。半圆和三角形也是紧密的连接在一起的，个数也相同并且个数是根据固定屏幕距离动态计算出来的，缩放屏幕坐标系时，半圆的形状大小并不改变。

锢囚锋

指暖气团、较冷气团和更冷气团（三种形状不同的气团）相遇时先构成两个锋面，然后其中一个锋面追上另一个锋面，即形成锢囚。我国常见的是锋面受山脉阻挡所造成的地形锢囚或冷锋追上暖锋或两条冷锋迎面相遇形成的锢囚。OcclucionFront类表示锢囚锋，它是单线、三角形以及实心半圆的组合符号（E:\workspace\m4\Bin\image\q_GuQiuFeng.png）。三角形和实心半圆的方向是相同的。从左向右画时，方向朝上，从右向左画时，方向朝下。其它属性特征和静止锋类似，详情参见静止锋。

上面对4类锋符号作了详细的阐述，但是这四类都是不带标签的锋符号，为了使各类锋符号能够带标签重新设计了一个FrontWithLabel类，配合上面的四类一起使用就可以满足需求。

#### 未定义符号1

未定义符号表示的空心圆构成的单线（E:\workspace\m4\Bin\image\q_NoSet1.png）。该符号跟实心圆的设计一模一样，属性特征也是相同的，详情请参见实心圆。

### 双线符号

双线符号是继承自DoubleLineSymbol类，在单线的基础上通过偏移生成新的两条单线，然后对两条新的单线进行各种需求处理，形成新的双线符号。设计的DoubleLineSymbol的类图如下，



DoubleLineSymbol也是继承自LabelLineSymbol，该类可以绘制带标签的双实线也可以绘制不带标签的双实线。根据需求调用相应构造方法即可方便创建带标签的双线符号。

介绍各种符号之前，先来看下给个符号之间的继承关系，设计的符号类图关系如下图所示，



从类图上看，双线符号主要包括双线箭头，以及另外3个比较特殊的符号，分别是急流核（JiLiuHeNew）、等变高线（ShuangGanAndShuangxian）、24小时变温线（\_24hourWeather）。下面依此对这些符号进行阐述。

#### 双线箭头

从类图上看，双线箭头包括5类，

ArrowMidLineSymbol

双线中间添加箭头（E:\workspace\m4\Bin\image\Arrow_DoubleLineMiddle.png）。该箭头是由双实线和一个箭头构成的，双线之间的距离和箭头三点之间的距离是屏幕坐标系下的距离，不会随着缩放而改变形状大小。

ArrowHeadLineSymbol

双线尾部添加箭头头部两条线相连接（E:\workspace\m4\Bin\image\Arrow_DoubleLineFront.png）。在双线的基础上添加了三角形形状的头部，然后尾部闭合。双线间的间距是屏幕距离，箭头的三角形边长也是屏幕距离，不会随着屏幕坐标系的缩放而改变形状大小。

ArrowHeadFilledLineSymbol

双线尾部添加箭头头部两条线相连接并且填充箭头（E:\workspace\m4\Bin\image\Arrow_FilledFront.png）。继承于ArrowHeadLineSymbol，只需要重新设置索引，更改绘制图元的样式，其它的属性特征和父类一样，详情参见父类的属性特征。

ArrowHeadTailLineSymbol

双线尾部添加箭头头部添加鱼尾形状的三角形（E:\workspace\m4\Bin\image\Arrow_DoubleLineBoth.png）。继承于ArrowHeadLineSymbol，鱼尾状的三角形是由两个小三角形成的，其它的属性特征和父类一样，详情参见父类的属性特征。

ArrowHeadTailFilledLineSymbol

双线尾部添加箭头头部添加鱼尾形状的三角形并且填充箭头（E:\workspace\m4\Bin\image\Arrow_FilledBoth.png）。继承于ArrowHeadFilledLineSymbol，和父类的区别就是箭头的中间部分填充了，其他的属性特征和父类一样，详情参见父类的属性特征。

#### 急流核

双线之间填充并且双线之间带斜网格（E:\workspace\m4\Bin\image\q_JiLiuHe.png）。在双线的基础上添加网格点，

双线之间的区域用指定的颜色进行填充。双线之间的区域宽度是屏幕距离，缩放时不会改变形状大小，但是网格点的数量会实时的计算变化。类图如下，



#### 等变高线

按照屏幕距离等间距的连接双线的顶点（E:\workspace\m4\Bin\image\q_BianGaoXian.png）。双线之间按照等屏幕距离添加‘|’形状符号，‘|’符号根据缩放动态的计算数量，同时形状大小不会改变。

#### 24小时变温线

每隔固定屏幕间距绘制相同的双线图案（E:\workspace\m4\Bin\image\q_BianWen.png），继承于ShuangGanAndShuangxian，双线中每条单线相邻两个点的中点连接形成了‘日’形状的符号，这些‘日’形状的符号是按照屏幕坐标等间距的分布排列的，会根据缩放动态的计算渲染，数量会改变，但是形状大小不会改变。

## 面状符号

当地图符号所指代的概念在抽象意义下可认为是定位于几何上的面时，称为面状符号。这时，符号所指代的范围与地图比例尺有关，且不论这种范围是明显的还是隐喻的，是精确的还是模糊的。用面状符号表示的有降雨范围，天气状态等等。色彩用于面状符号，对于表示制图对象的面状分布有着极大的实用意义。Hatch位图的使用让面状符号更加多样化，更加形象化。天气符号的使用使得区域的实际意义更加明确。

面状符号先是由单线经过首尾闭合形成的封闭多边形，然后再对多边形进行填充而形成的二维平面图形。面状符号大体分为3类，下图展示了它们的继承关系，



#### 颜色填充区域（E:\workspace\m4\Bin\image\q_marea.png）

颜色填充多边形是由PolygonSymbol类去实现的。设置好填充的颜色，然后用该颜色去填充不规则的多边形。

#### Hatch图填充区域（E:\workspace\m4\Bin\image\q_rain.png）

HatchFullPolygon类实现了用hatch图填充多边形的功能。设置Hatch位图，然后就可以用Hatch图去填充不规则的多边形

#### 标签填充区域（C:\Users\JD\Desktop\MICAPS测试\spcimages\22.bmp）

标签填充区域符号主要是方便天气区域符号的显示，例如沙暴区域等。标签填充区域符号由颜色填充区域和标签构成，所以PolygonWeather类继承于颜色填充区域符号类PolygonSymbol。设置标签的属性以及填充的颜色就可以用这两者来填充不规则的多边形区域。