# easyX+

## 意图

为什么写这个图形库呢？

因为

1. easyX官网自称easyX是一套针对VC的图形库，但却没有使用C++的功能，对不起“EasyX Library for C++”这个名字（也许叫它EasyX Library for C更好）。
2. 它的函数命名并不合理，既不是Camel命名法也不是Pascal命名法（我的意思并不是说不采用这两种方法就不合理），而是一股脑的小写字母，严重模糊了代码的可读性。
3. easyX宣称面向初学者，但很多内容对初学者并不友好（例如颜色模型中的HSL、HSV）。对于这些内容，初学者基本很少接触。而当他们晋升为高级程序员之后，他们又不满足于简单的easyX转而使用openGL等。

## 特点

* 简单，易上手
* 组织合理
* 计时器功能（timer）。
* 键盘支持

## 使用平台

Windows，系统版本待定。

## 使用语言

C++。

## 模块划分

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 剪裁区 | Window | 颜色模型 | Geometry | Text | Mouse | Keyboard | Timer |

## Window

所有的可绘绘操作（包括几何图形和文字输出）都与特定的window相关联。除此之外，剪裁区也与特定的window关联。Window相当于一个画布，所以的绘制都在某个特定的窗口上进行。

#### 属性

该模块只包含于窗口有关的一些属性：

* 大小，即长、宽。
* 背景色。

#### 特点

* 多窗口支持
* 窗口之间各自独立

因为该图形库致力于简单，所以那些复杂的且与图形绘制无关的属性都不被支持（例如边框类型，窗口按钮等等）。

## 颜色模型

颜色模型模塑图形的颜色，并遵循统一的接口。默认支持的颜色模型仅为RGB颜色模型。

## Geometry

Geometry模块囊括大部分的几何图形，包括（注意：该模块不支持‘体’）：

* 点 ： 点
* 线 ： 直线、弧、贝塞尔曲线、折线
* 面 ： Chord、椭圆、Pie、多边形、矩形

注意**线**与**面**的区别：线不是封闭区域，而面是。

### 几何属性

几何属性是与几何图形本身相关联的属性。就图形学来说，图形的属性主要分为**颜色**和**形状**（这个库中表现为点的数量语位置）。

## Text

该模块为文本支持模块，假定文本与一下属性相关联：

* 字体，包括字体类别、大小、样式（斜体，加粗等）
* 颜色，包括字体颜色、背景填充颜色
* 位置
* 输出区域（以矩形表示）

## Mouse

我希望鼠标与键盘的支持对初学者来说是非常友好的。因此，这个模块只提供这些功能：

* 检测单个按键状态：按下、释放
* 获取鼠标位置

这里不可能出现鼠标左右按键同时按下的情况，因为我觉得这不太可能（-\_-）。所以，同时检测两个以上的鼠标按钮状态在这里是不支持的， 并且除传统的左右键以外的按钮都在这个库的功能范围外。

## Keyboard

键盘模块提供的功能为：

* 检测单个按键的状态（按下或释放）
* 检测多个按键的状态（实际上，只检测是否有多个按键同时按下，同时释放没有意义）

## Timer（可选）

每个Timer由一段执行代码（即函数）进行初始化，每个window最多只能由一个timer与之相关联。Window关联timer后，可以进行timer的取消与更改。

## 剪裁区（可选）