可视计算与交互概论大作业报告

邱荻 2000012852

一、选题介绍

Drawing Software

在这个任务中，我制作了一个绘图板 VCIPainting，包含了直线，曲线，圆等多个图形以及多种操作，为其制作了精美的 UI 交互界面，就像专业的绘图板一样。

【Demo】：https://disk.pku.edu.cn:443/link/734706003DC11DAA847A86BBD1747B54
有效期限：2023-03-08 23:59

二、运行环境

QT5.6.1

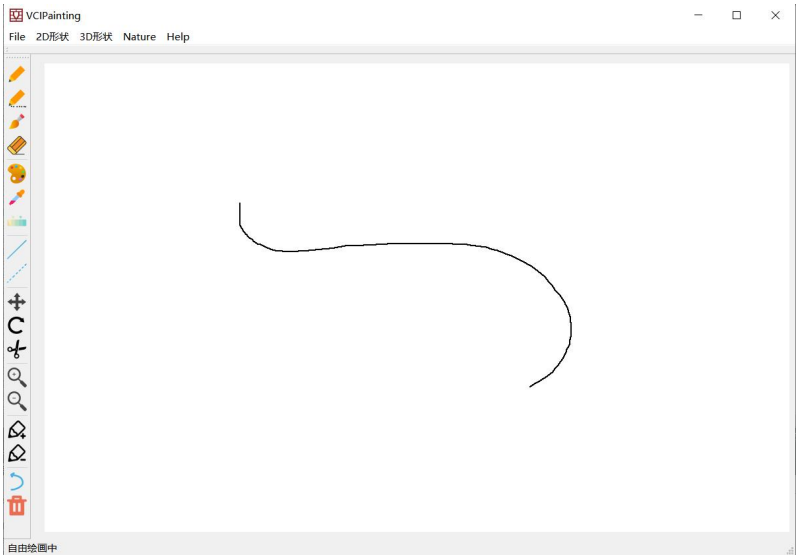
三、实现思路

打开和保存图片：

实现思路：
直接使用 QFileDialog::getOpenFileName 和 QFileDialog::getSaveFileName。

画自由线条 drawPoint：

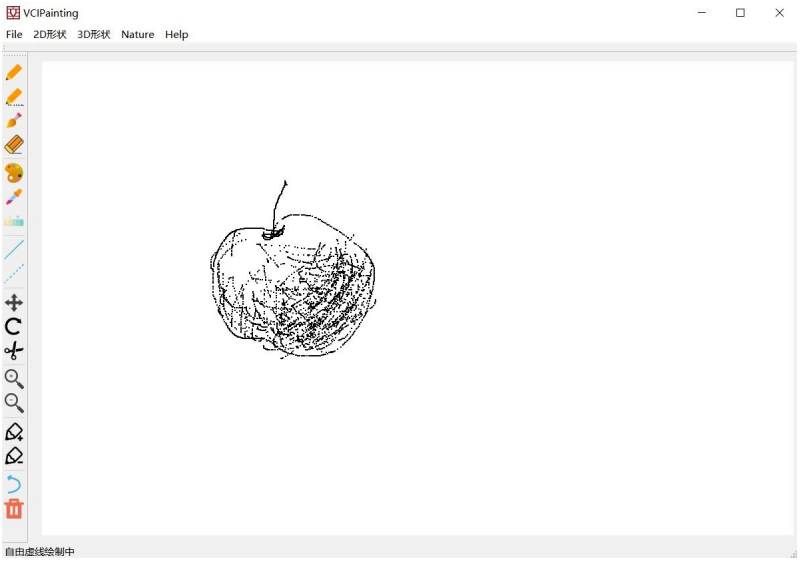
使用方法：鼠标点击确定起点，任意移动画线，直到松开。
实现思路：记录鼠标按下之后移动直到松开的所有点，每两个点之间连一条直线，即可画出自由实线。



画自由虚线 drawDottedPoint：

使用方法：鼠标点击确定起点，任意移动画线，直到松开，虚实程度可由鼠标移动速度控制，速度越快，线越虚。适合素描画或者像素画。

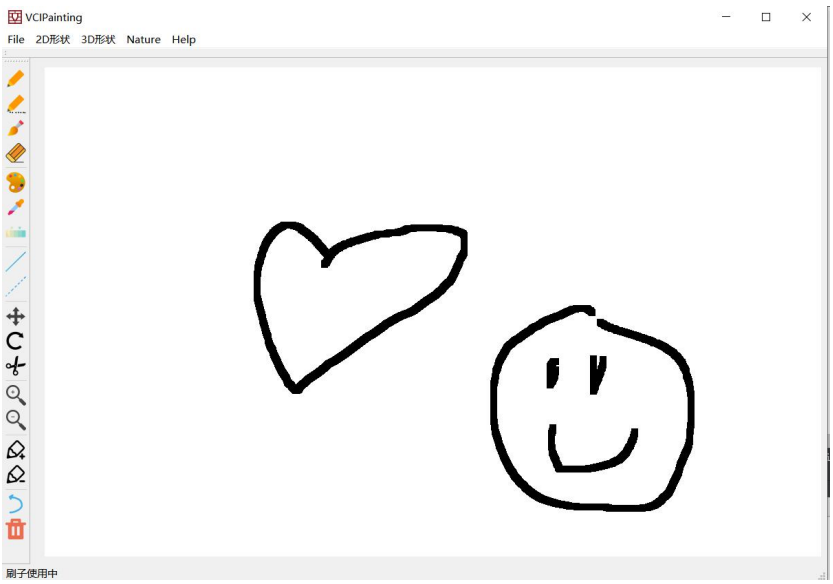
实现思路：记录鼠标按下之后移动直到松开的所有点，画出这些点，即可画出自由虚线，且虚实程度可由鼠标移动速度控制。



刷子 drawBrush：

使用方法：鼠标点击确定起点，任意移动画线，直到松开。

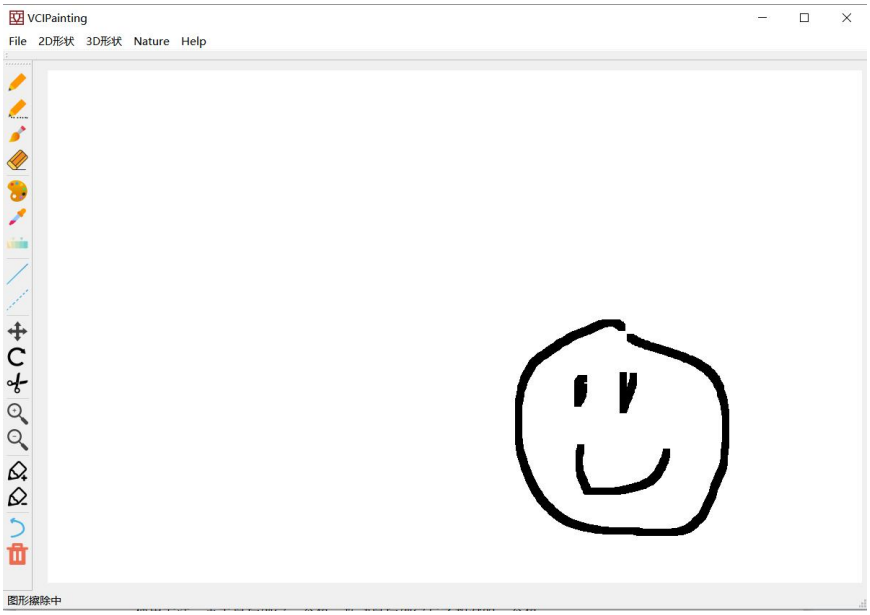
实现思路：在 drawPoint 的基础上将画笔大小初始化为【画版目前选择的笔刷大小+8】，以达到更粗的刷子效果，填色效率加倍。



擦除 Eraser：

使用方法：点击需要擦除的图形，就让图形消失（有别于自由擦除，如果有自由擦除的需求，直接把颜色调成白色用自由画笔即可）。

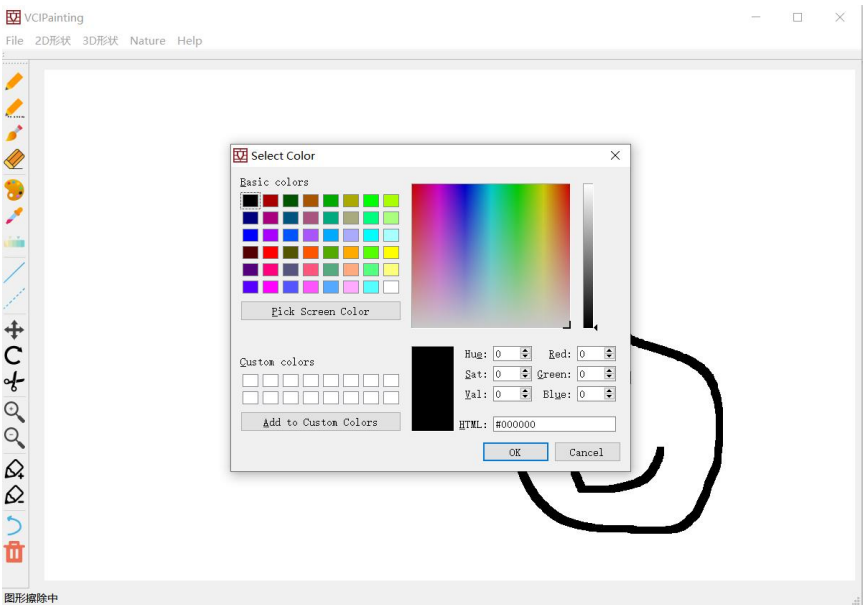
实现思路：遍历所有已画图形，找到鼠标点击的是哪个图形，然后将这个图形删掉。



调色盘 Palette：

使用方法：点击后会出现颜色选择的界面，点击即可选择画笔颜色。

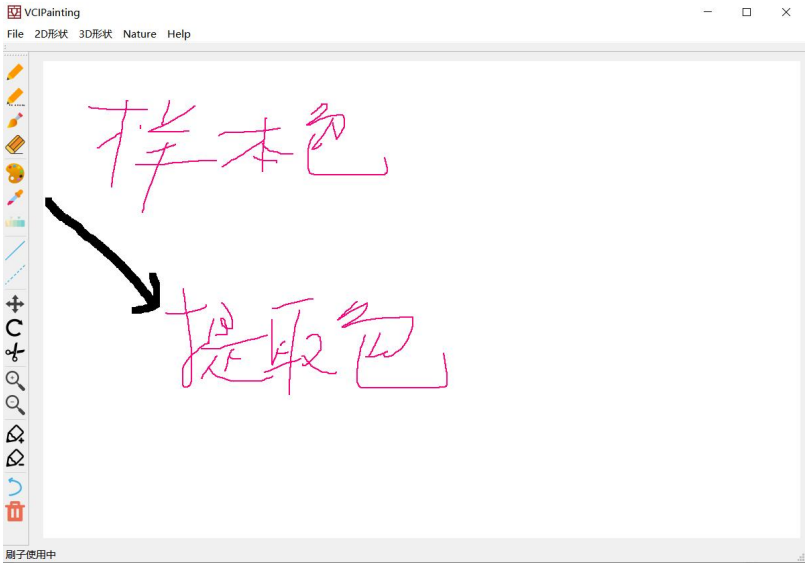
实现思路：直接调用 QT 的 QColorDialog 类。



提色器 Dropper：

使用方法：点击想要提取的颜色，即可使画笔成为这种颜色。

实现思路：遍历所有图形，找出鼠标点中的图形，然后将画笔颜色设置为该图形的颜色。如果鼠标没有点中图形就设置为白色（背景色）。



渐变显示 Gradient：

本作品的**创新**点。

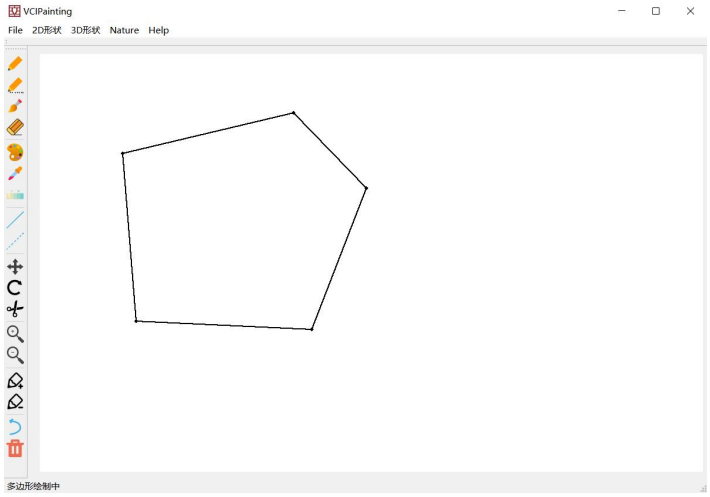
使用方法：点击即可看到画面变成**【动态水墨风】**，很有韵味，**中国风十足**。再次点击即可复原。

实现思路：点击后在 paintEvent 中不再以整个图形为单位画，而是一个点一个点画，且画笔颜色由与时间有关的随机数生成，每个点的颜色与时间以及它所处的位置有关，rgb 随着点的次序 i 和随机数 a 变化，这样就可以实现渐变的水墨风效果，且会随时间流动，有动态效果。

画直线 drawLine：
使用方法：鼠标按下决定起点，松开决定终点。
实现思路：使用 Bresenham 画线算法，参考 lab1

画虚线 drawDottedLine：
使用方法：鼠标按下决定起点，松开决定终点。
实现思路：在 drawLine 的基础上改为依次十个点画十个点不画，即可实现虚线效果。

画多边形 drawPolygon：
使用方法：鼠标点击多个点，将点依次相连形成多边形。
实现思路：鼠标点多个点，将每个点和后面的那个点依次调用实现好的 drawLine 函数相连，将最后一个点和第一个点相连。



画三角形 drawTriangle:
使用方法：点击鼠标确定三角形的位置，拖动确定三角形的大小。
实现思路：以多边形为基础，初始化的时候直接初始化为一个三个点的多边形，然后由鼠标位置计算出三个参数的位置。

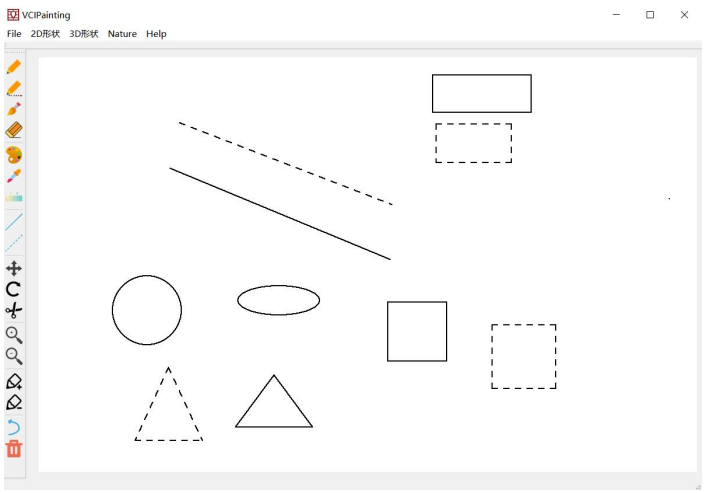
画长方形 drawRectangle:
使用方法：点击鼠标确定长方形的位置，拖动确定长方形的大小。
实现思路：以多边形为基础，初始化的时候直接初始化为一个四个点的多边形，然后由鼠标位置计算出四个参数的位置。

画正方形 drawSquare:
使用方法：点击鼠标确定正方形的位置，拖动确定正方形的大小。
实现思路：以多边形为基础，初始化的时候直接初始化为一个四个点的多边形，然后由鼠标位置计算出四个参数的位置，并保证四个参数位置是符合正方形要求的。

画虚线多边形/长方形/正方形/三角形：
使用方法：和实线版使用方法相同。
实现思路：将原来连线时使用的 drawLine 改为 drawDottedLine。

画圆 Circle：
使用方法：点击之后拖动确定大小然后松开。
实现思路：第一个参数为圆的中心，第二个参数为鼠标点击的点到鼠标松开的点的向量。首先确定半径 r，取向量的 xy 中小的那一个。使用中点画圆法，可直接参考计算机图形学--2种圆绘制算法原理及代码实现_画圆算法_emandora 的博客-CSDN 博客

画椭圆：
使用方法：点击之后拖动确定大小然后松开。
实现思路：第一个参数为椭圆的中心，第二个参数为鼠标点击的点到鼠标松开的点的向量。使用中点画椭圆的算法，可直接参考计算机图形学之绘制椭圆_LLOZEL 的博客-CSDN 博客_计算机图形学画椭圆。

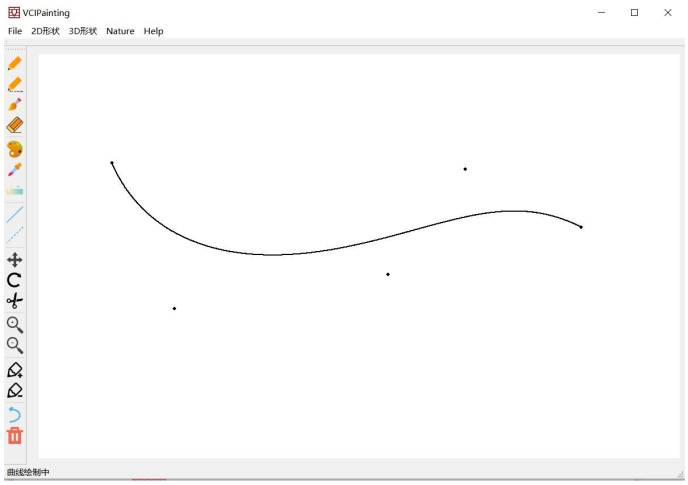


画曲线 drawCurve:

使用方法：点若干个点，画版会以这些点为参数形成贝塞尔曲线。

实现思路：用贝塞尔曲线实现，参考 lab1。经实验发现 t 设置为在循环中每次增加 1e-3 效果很好，既满足了直线的连续性，又保证了运算时间不会太长。

注：点数在 16 个以内的运算时间完全可以满足用户需求，超出 16 个之后运算时间可能会达到秒的量级（不过十几个也够用了）。



平移 Traslate：

使用方法：点击需要移动的图形，将其拖拽到相应位置。

实现思路：圆和椭圆的第一个参数是中心，用来决定位置，所以将其修改为鼠标位置 pos 为中心，其他图形的其他各个参数代表了图形上一些关键点的位置，将其加上【鼠标目前位置-鼠标点击的位置】。

旋转 Rotate：

使用放大：点击要旋转的点，然后鼠标移动进行旋转。

实现思路：记下中心点到鼠标点击的点的方向和中心点到鼠标移动时/释放时的方向，计算出夹角，然后将原图形的各个参数保持中心不变，将中心到参数的向量进行旋转。

$$M(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

裁剪 Clip：

使用方法：框出需要的直线，即可实现直线的裁剪。

实现思路：Cohen-Suthland 算法，对直线段两端端点进行编码 再进行求或求与操作从而判断是否简弃或简取。将每条线段的端点都赋予四位二进制编码 D3D2D1D0

若 x<xleft 则 D0= 1 否则 D0 = 0

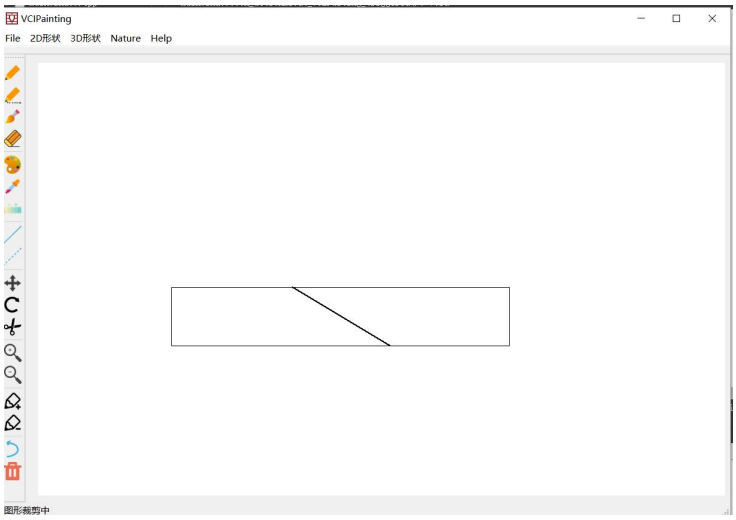
若 x<xright 则 D1= 1 否则 D1 = 0

若 x<xbotton 则 D2= 1 否则 D2 = 0

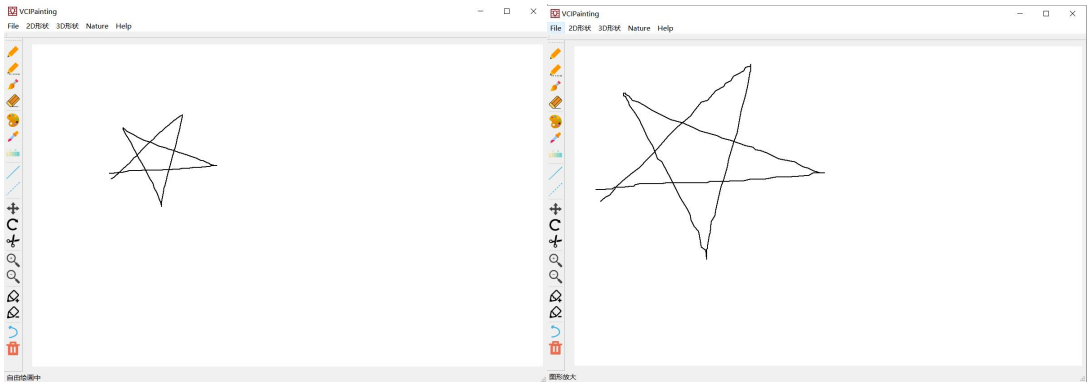
若 x<x~ top~ 则 D3= 1 否则 D3 = 0

再将窗口变成 9 个区域

D0 对应左边界
D1 对应右边界
D2 对应下边界
D3 对应上边界
如果两条线段进行或运算|
code1|code2 = 0 则对直线进行简取
如果两条线等进行与运算&
code1&code2 != 0 则对指向进行简弃
当这两个条件都不满足是 就是不能简取或者简弃时
需要求出直线段和窗口边间的交点 在交点处把线段一分为二
再按 左 右 下 上的顺序求出直线段与窗口做边界的交点为 p3
则 p1p3 必在窗口外 可简弃，然后重复进行操作。



图形放大 ZoomIn/缩小 Zoomout：
使用方法：点击所要放大/缩小的图形，即可实现放大/缩小效果。
实现思路：实现 scale 函数，以倍数为参数，放大即参数为 1.2，缩小即参数为 0.8。若是圆形或者椭圆形，就把第二个参数（圆形和椭圆形第一个参数决定位置，第二个参数决定大小）乘以 1.2 或者 0.8。若是其他图形，就保持中心不动，其他的参数点到中心的距离乘以 1.2 或者 0.8，即可实现放大或缩小。



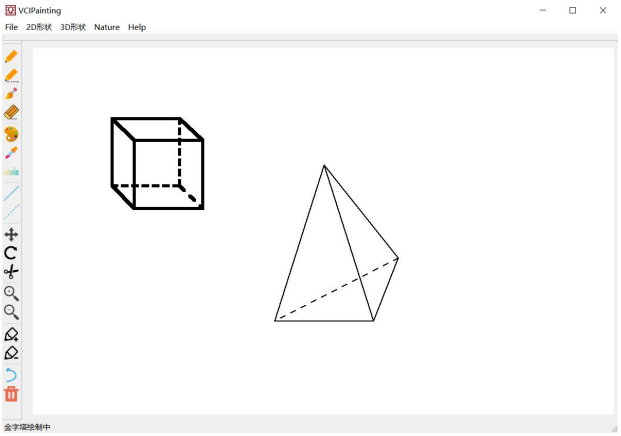
笔刷放大/缩小：
使用方法：按下按钮即可实现笔刷的放大/缩小。
实现思路：按下后讲 mainwindow 的笔刷尺寸增加或减少。

撤回 Drawback：
使用方法：按下后点击空白处，点击一下即可撤回最后画一个图形。
实现思路：将画过的图形用 QList 存起来，点击撤回就找出最后一个图形然后删掉。（不用手动遍历，可以直接用.at(i)）

清除画版 Delete:
使用方法：按下即可清除画版上所有的图形（此操作设计为不会删除背景，如果需要删除背景，可以直接重开）。
实现思路：如果是选了清除画版，就在 paintEvent 里删掉所有保存的已画图形 figures。并且图标使用了醒目的红色垃圾桶，意在提醒使用者谨慎点击。

3D 图形：
新意十足的长方体绘画 Cuboid：
使用方法：点击鼠标确定一个角，拖动鼠标确定与之相对的一个角。
实现思路：记录按下鼠标和鼠标释放的两个位置，以之为相对应的两个角，然后计算另外 6 个角的位置（将横纵坐标之差 4 等分），再依次连接为相应的实线或虚线。

新意十足的金字塔 Pyramid：
使用方法：点击鼠标确定一个角，拖动鼠标确定另一个点。
实现思路：记录按下鼠标和鼠标释放的两个位置，以之为三角形的两个顶点，然后计算另外四个点的位置，再依次连接为相应的实线或虚线。



Nature：

这个模块为特别模块，用户可以用此模块的功能来摧毁自己的画作，让画作动起来。当画手压力很大，或者不满意自己的画作的时候，想要将其销毁，但又不满足于简简单单的点击叉叉的时候，这个模块将很好地满足画手的需求。画手可以对自己所画的图形产生地震攻击或者飓风攻击，让自己所画的图形遭受一点伤害，以发泄心中的怀才不遇和愤懑，释放压力。最后程序在自然灾害中自动关闭，一切都归于沉寂，而画手也重振旗鼓，整装前行。

地震 Earthquake:

使用方法：点击图形，鼠标在图形旁移动或者旋转，即可让图形抖动，产生地震效果。然后程序就会自动关闭。

实现思路：在旋转的基础上进行改动，控制角度在一定范围之内。

飓风 Hurricane：

使用方法：点击图形，鼠标任意晃动，即可让图形旋转着飞走，像被可怕的飓风刮走一样。然后程序就会自动关闭。

实现思路：在旋转的基础上进行改动，使图形疯狂旋转，并且坐标的 x 和 y 逐渐减小。

其他人性化设计：

左下角有状态栏，帮助用户知晓自己所处的操作。

窗口可以手动调节大小，以适应不同大小的显示器和不同的用户习惯。

将画布的这个 label resize 即可。

帮助 Help，点击即可跳转到帮助的网页。

实现思路：使用 QDesktopServices::openUrl 函数。

为常用的操作设置了快捷键。