

MyGraphicsSystem 系统使用说明书

作者姓名：戚赞 学号：161220097

(南京大学 计算机科学与技术系, 南京 210093)

摘要：经过三个月的学习，我终于基本完善了我的图形学大作业，实现了基本要求的功能和一些附加的优化操作，使我的图形学大作业基本能够像一个比较好的绘图软件。其中我实现了直线、圆、椭圆、曲线(Bezier曲线)、多边形、填充区域的输入和编辑功能，还能够对于直线、圆、椭圆、曲线、多边形进行变换操作，还实现了对于图片的保存功能，包括平移、旋转、缩放。此外还有3D图形的显示，其中还包括了对于3D图形的视角转换的操作，从而可以从多个角度来进行对于3D图形的观察。

关键词：子功能，核心代码介绍，系统操作方式，运行结果截图。

一、开发环境、代码编译、程序运行说明

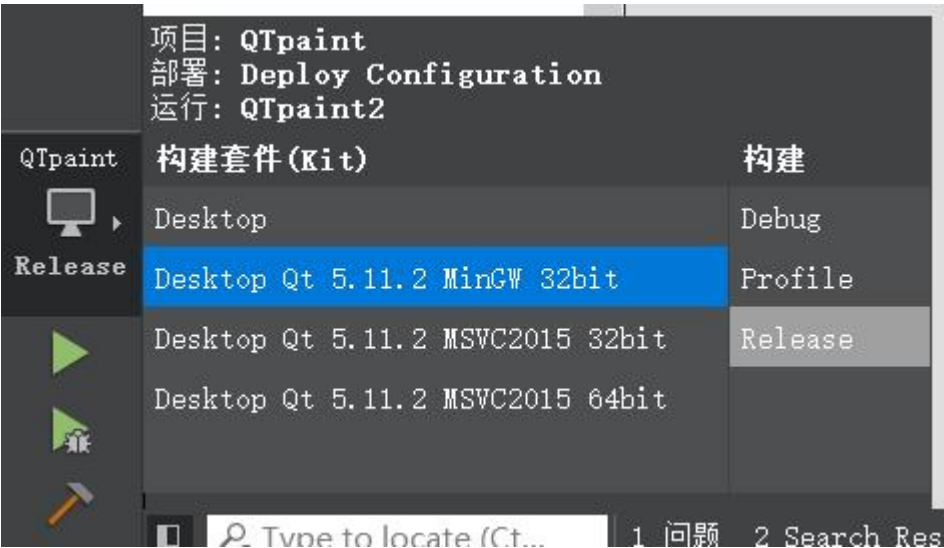
①开发环境说明

编程语言	系操作系统	GUI框架	3D图形库
C++	WINDOWS 10	QT 5.11.2	OpenGL

使用的IDE为QT 5.11.2，提供了现成的GUI框架，从而能够更加容易的写出一个良好交互的应用程序。

②代码编译说明

1. 安装QT 5.11.2版本，并且安装构建套件Desktop Qt 5.11.2 MinGW 32bit
2. 点击Qtpaint.pro文件，打开项目
3. 选择Desktop Qt 5.11.2 MinGW 32bit 套件



4. 点击运行按钮(ctrl + R)，实现项目编译运行，从而能够输出应用程序
5. 最后编译成功，实现了输出

```
QTPaint2 x
13:55:44: Starting E:\pro_QT\build-Qtpaint-Desktop_Qt_5_11_2_MinGW_32bit-Release\release\Qtpaint...
```

③程序运行说明

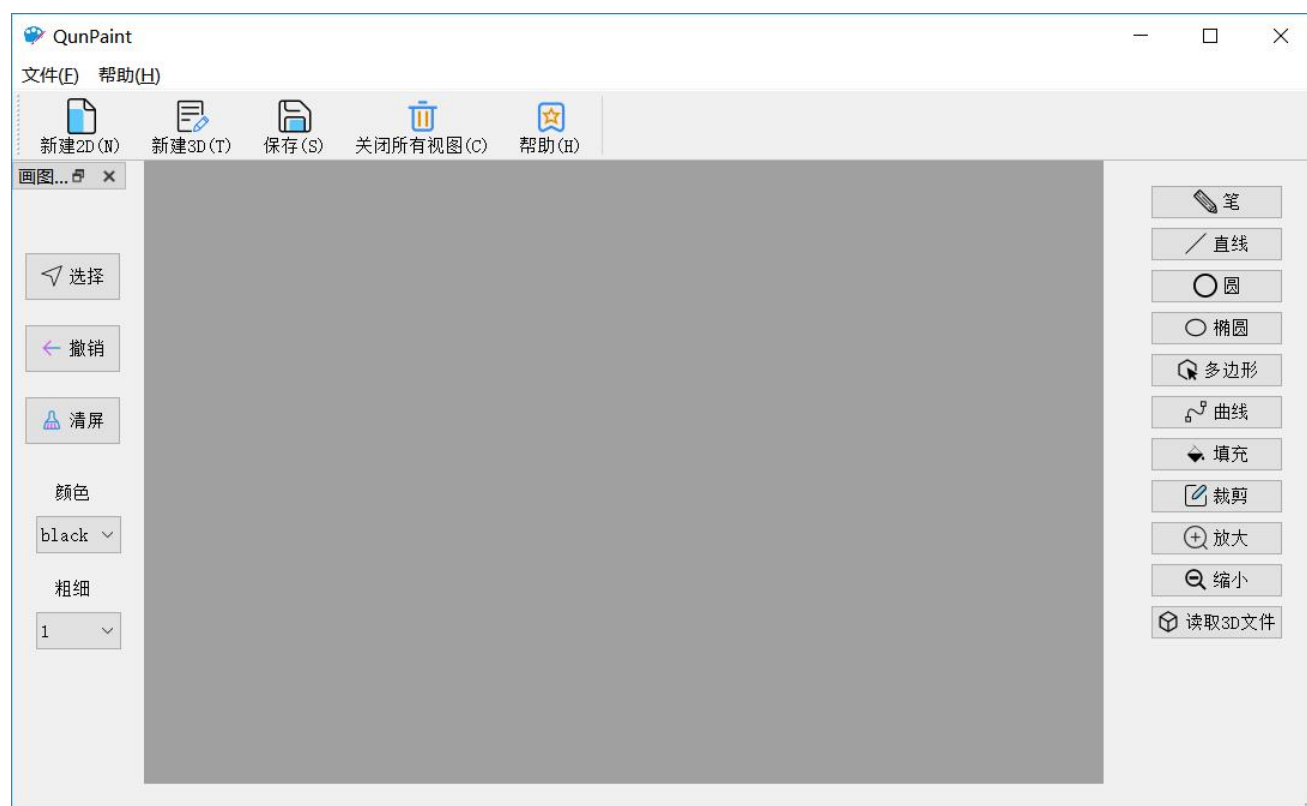
两种方式

1. 按照上面的编译过程进行编译代码，从而实现输出，操作程序运行
2. 我已经将**应用程序包括所要用的配置文件**全部打包成一个**qunpaint.exe**文件，运行这个.exe文件能够直接应用程序，不需要额外的将代码进行编译

二、系统功能及核心代码简介和相应的运行效果截图

①系统的整体框架介绍

先观察一下系统的整体构成，看看显示的主界面。



主界面包括一下四个方面进行了组成。

1.菜单栏

菜单栏包括了文件的操作，主要功能为——新建2D画板（快捷键 ctrl + N），新建3D画板(快捷键 ctrl+ T)，保存2D画板(快捷键 ctrl + S)，关闭所有视图(快捷键ctrl + C)和帮助(ctrl + H)。

由于2D画板和3D画板的实现是不一样的接口，为了系统程序的鲁棒性，我设置了整个画板只会有一个画板进行显示，从而使得程序的鲁棒性提高了，比如打开3D画板点笔等是没有效果的。而由于实现了修改退出提示保存的消息，我就设置了关闭所有视图，利用这个按钮来进行关闭操作。

帮助按钮——显示程序的一些说明，作者的联系方式，以及源代码的位置——GitHub上的网页代码信息。

2.左侧可移动工具栏

包括了一些常用的工具。

选择按钮——点击选择，就不会在画板上触发任何事件，你可以面对画板点击，而不影响画板的使用。

撤销按钮——每次确定绘图时，就将图片进行了保存的机制，所以点击撤销按钮，会将上一次的保存的显示出来，代码和运行效果在下面展示

全屏按钮——将画布清空，同时保存，如果不小心清空，点击撤销按钮能够返回到上一层次。

3.右侧核心功能按钮

包括选择图元，填充，裁剪，放大，缩小，打开3D文件等一系列的功能，是整个画板的核心功能所在。

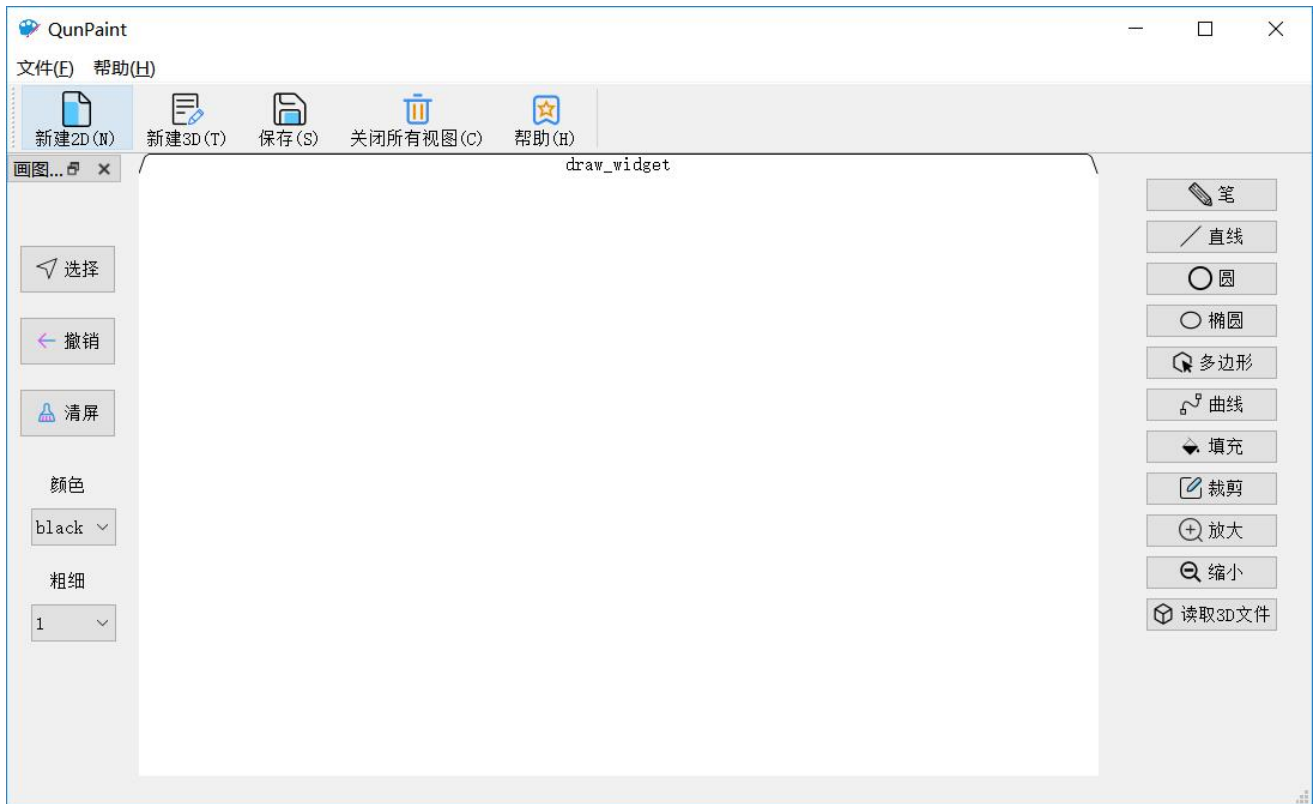
②2D图形的绘制

首先说明绘制图形的过程包括——初始化绘制图形，图形的编辑变换和图形的完成绘制操作三个过程。2D图形一些普遍按钮的说明：编辑模式下，绿色点为旋转点，蓝色点为平移点，红色点为可编辑点，缩小与放大为右侧按钮键。

在编辑模式下，必须点击到空白地方，才能退出编辑模式，完成其余图形的绘制！

1.打开画板

点击新建2D按钮，或者使用快捷键Ctrl + N来打开一个2D画布，效果如下



核心代码讲解：

```
void MainWindow::on_actions_T_triggered() //新建
{
    this->ui->mdiArea->closeAllSubwindows(); //关闭视图
    this->cur_widget = nullptr; //cur_widget为窗口指针
    this->cur_3D_widget = nullptr;
    mywidget *draw_ = new mywidget;
    QMdiSubwindow * t = ui->mdiArea->addSubwindow(draw_); //加入视图
    draw_->showMaximized(); //使得最大化
    cur_widget = draw_; //设置2D窗口指向新建窗口
    ui->mdiArea->setActiveSubwindow(t); //设置为活动窗口
    t->setWindowTitle(tr("draw_widget").arg(1)); //设置名称
    t->show(); //展示
}
```

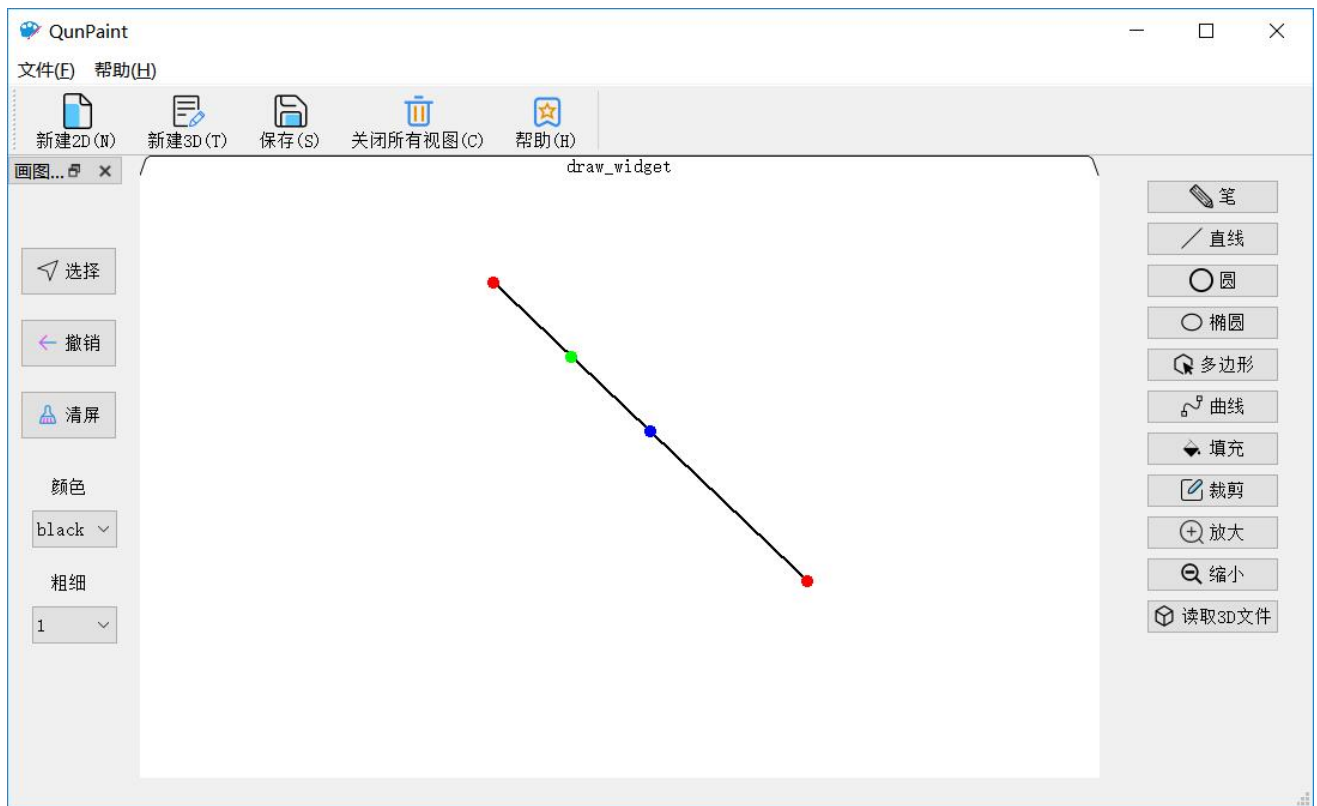
从而显示了一个画布功能。

2.直线的绘制

①直线的绘制：

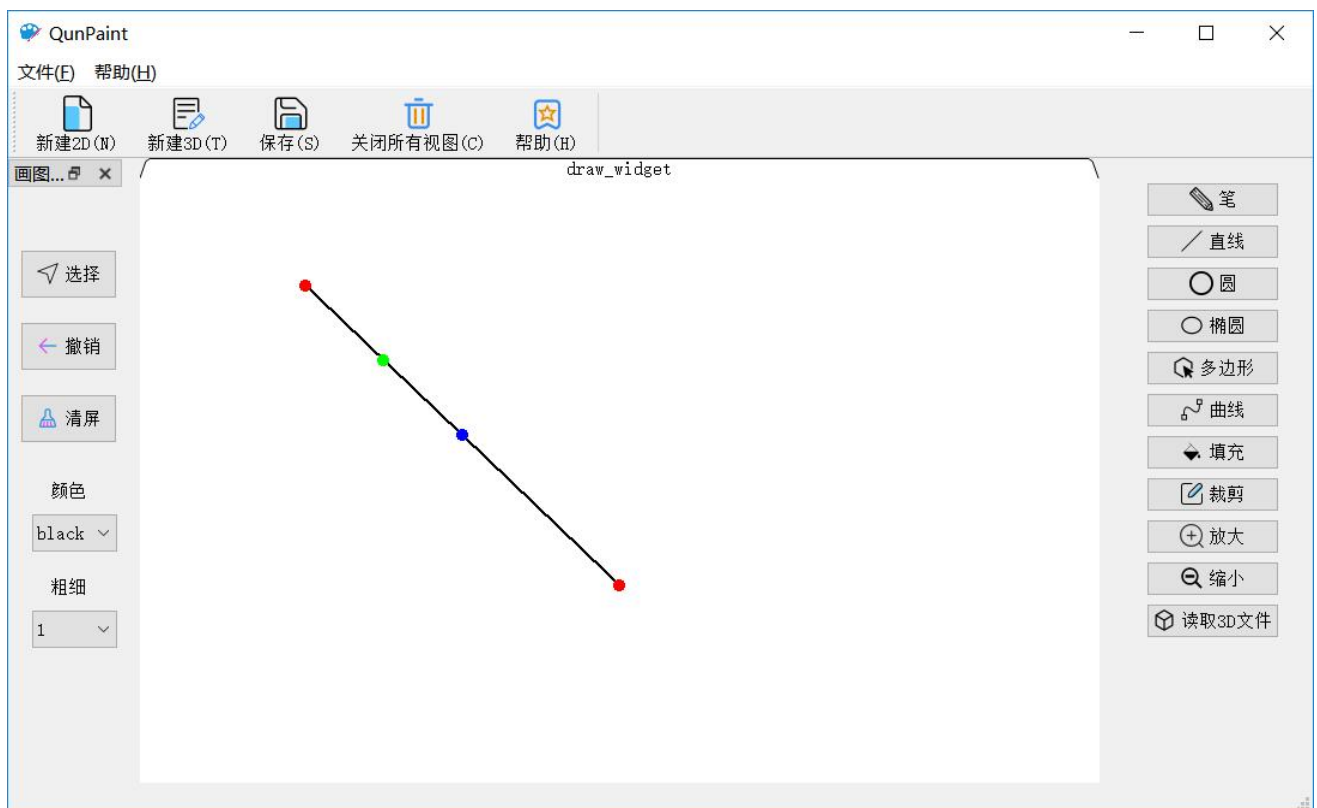
1. 点击右侧直线按钮一下，进入绘制直线的模式
2. 在空白画板的任意一处按下鼠标按钮，然后不松开进行拖动，确定画的直线的位置
3. 松开，直线绘制完成，进入编辑模式，如下图所示

具体见截屏的GIF图，已放在实验报告中一起提交。

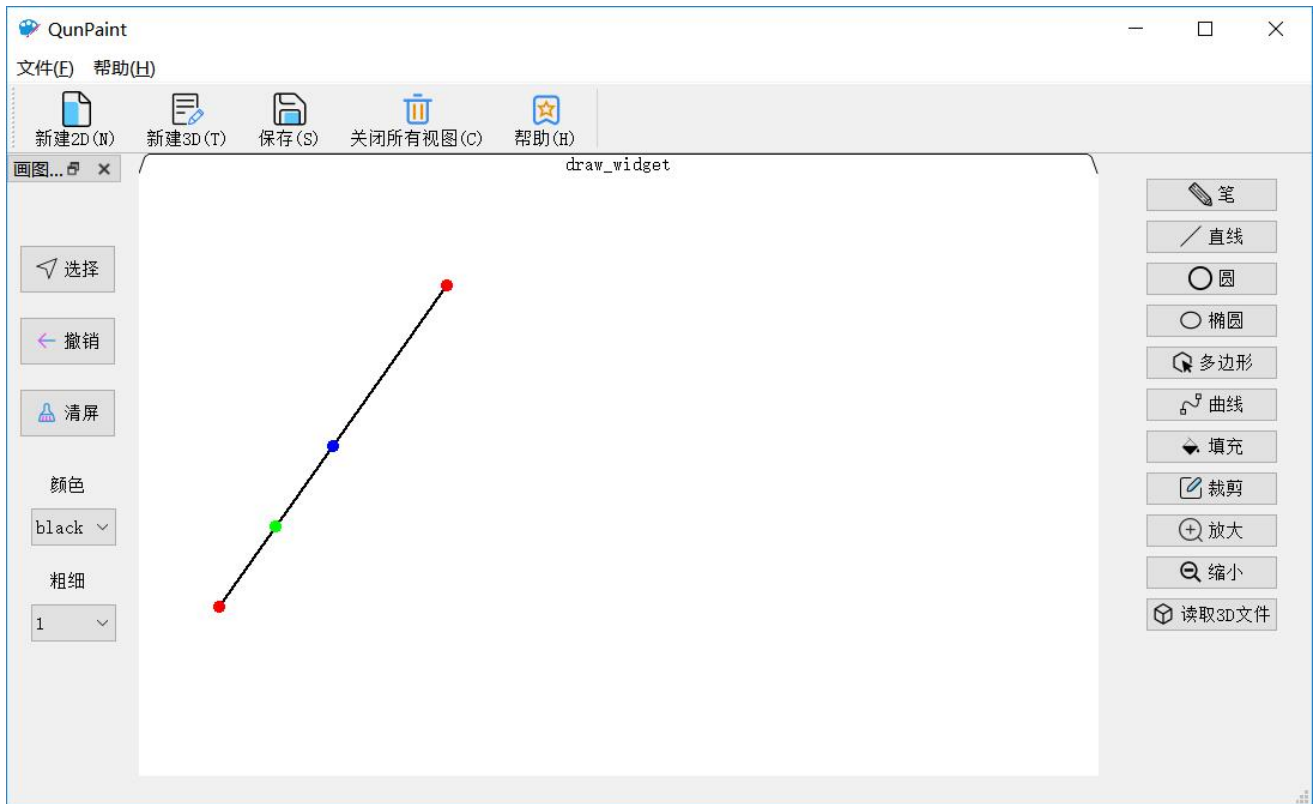


②直线的变换：

1.平移功能：点击**蓝色**的点，按住，拖动，最后拖动到想要的位置，松开，继续进入编辑模式

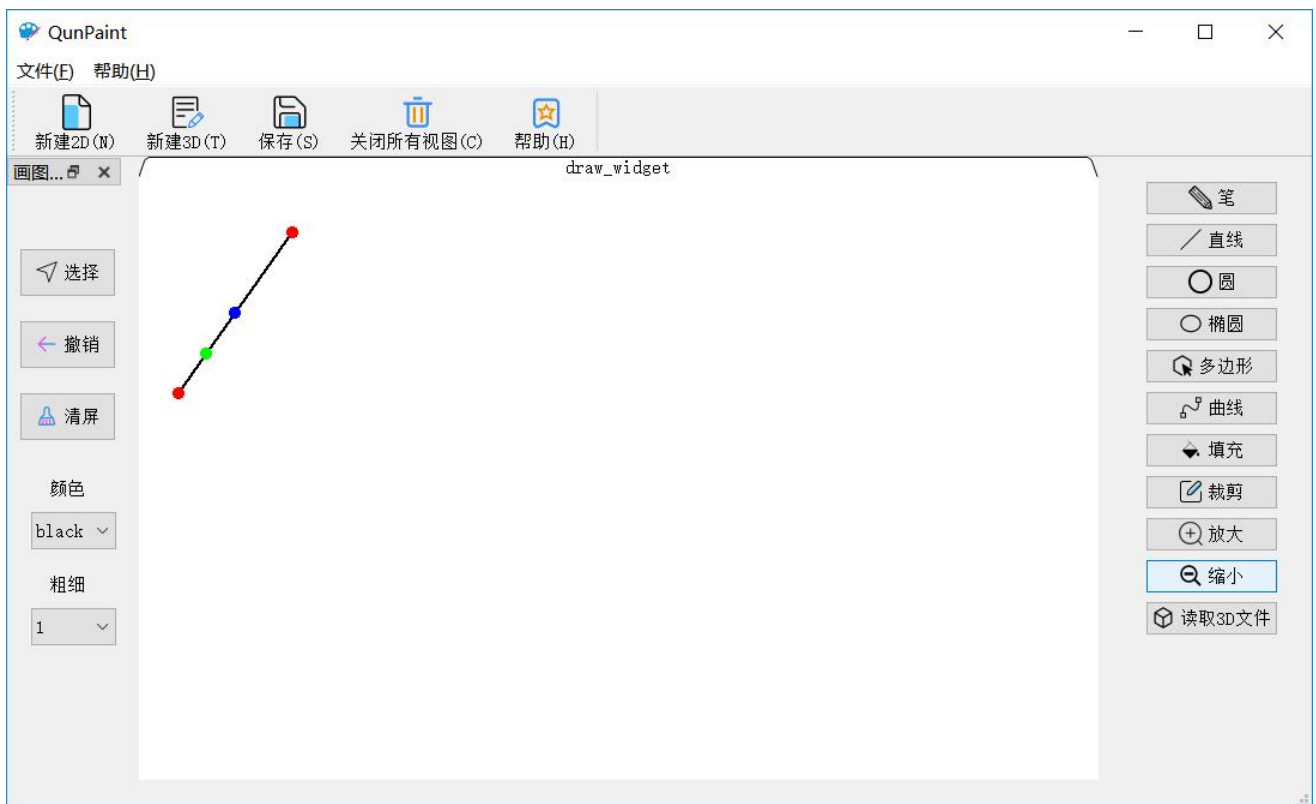


2.旋转功能：点击**绿色**的点，按住，拖动，最后拖动到想要的位置，松开，继续进入编辑模式，直线会跟随旋转，效果如下图所示。



3.缩放功能：点击缩小，或放大按钮，会将直线按照**原点**进行缩放操作

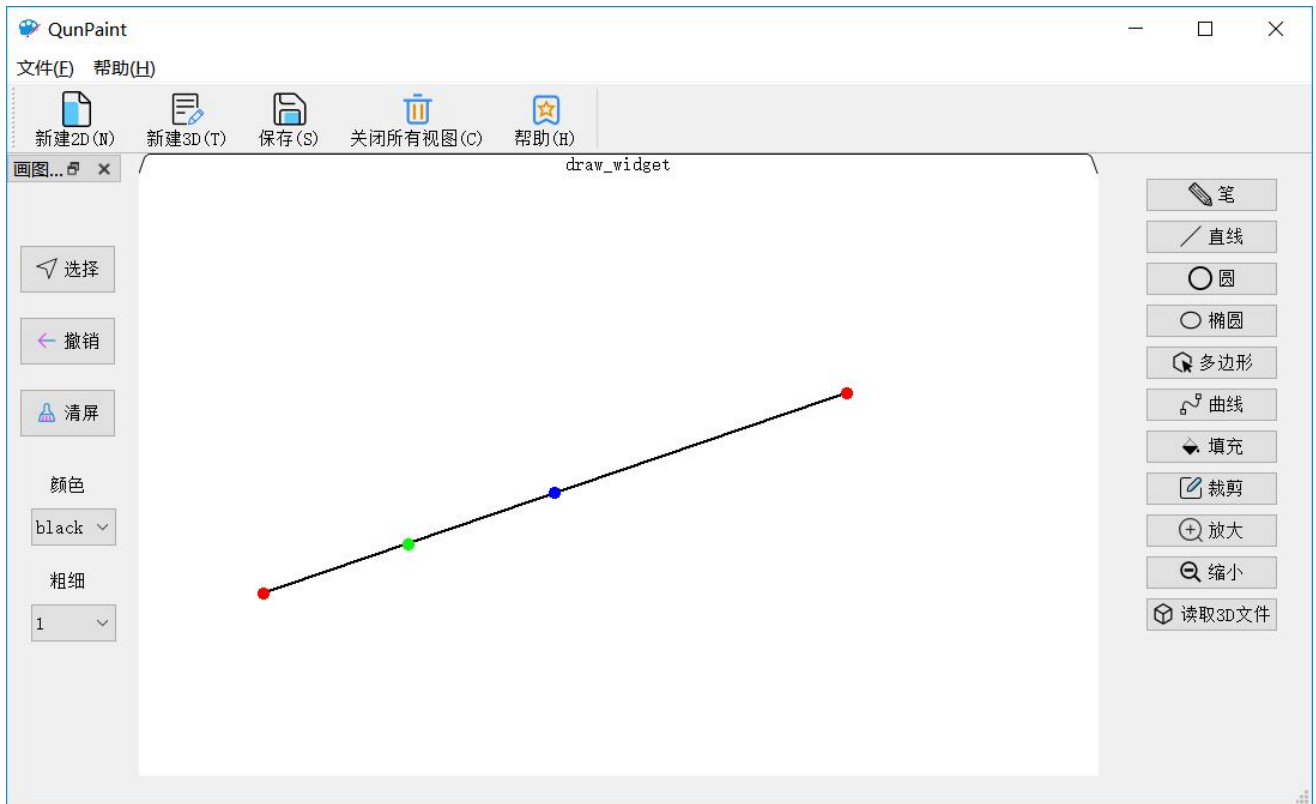
缩小：



放大：和一开始的位置是一样的

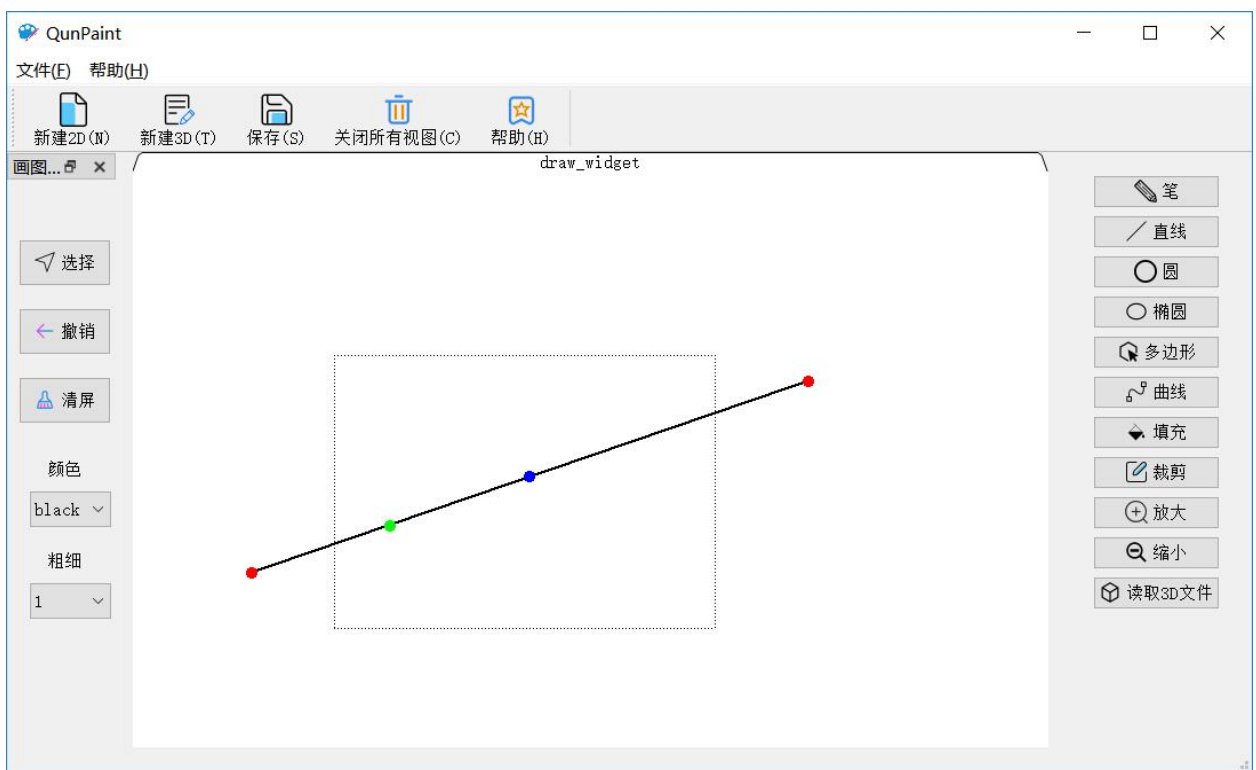
③直线的编辑：

按住任意红色的顶点进行拖动，直到拖动到想要的位置，松开，进入编辑模式，完成编辑

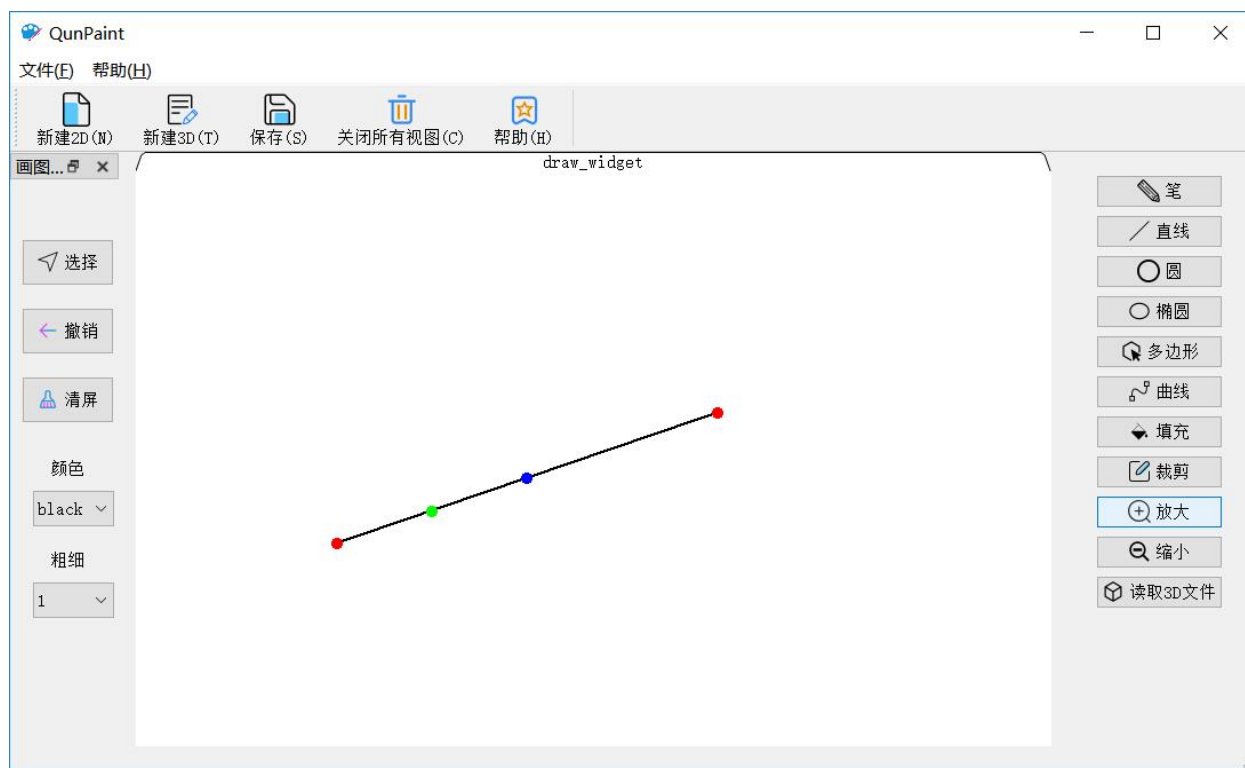


④直线的裁剪功能：//特有功能，裁剪按钮的专属

1. 在直线的编辑模式下点击裁剪按钮
2. 在画板上点击任何一处，拖动，拖动出一个矩形裁剪框，如下图所示，如果不满意，可以重新在画板上拖动出一个新的矩形



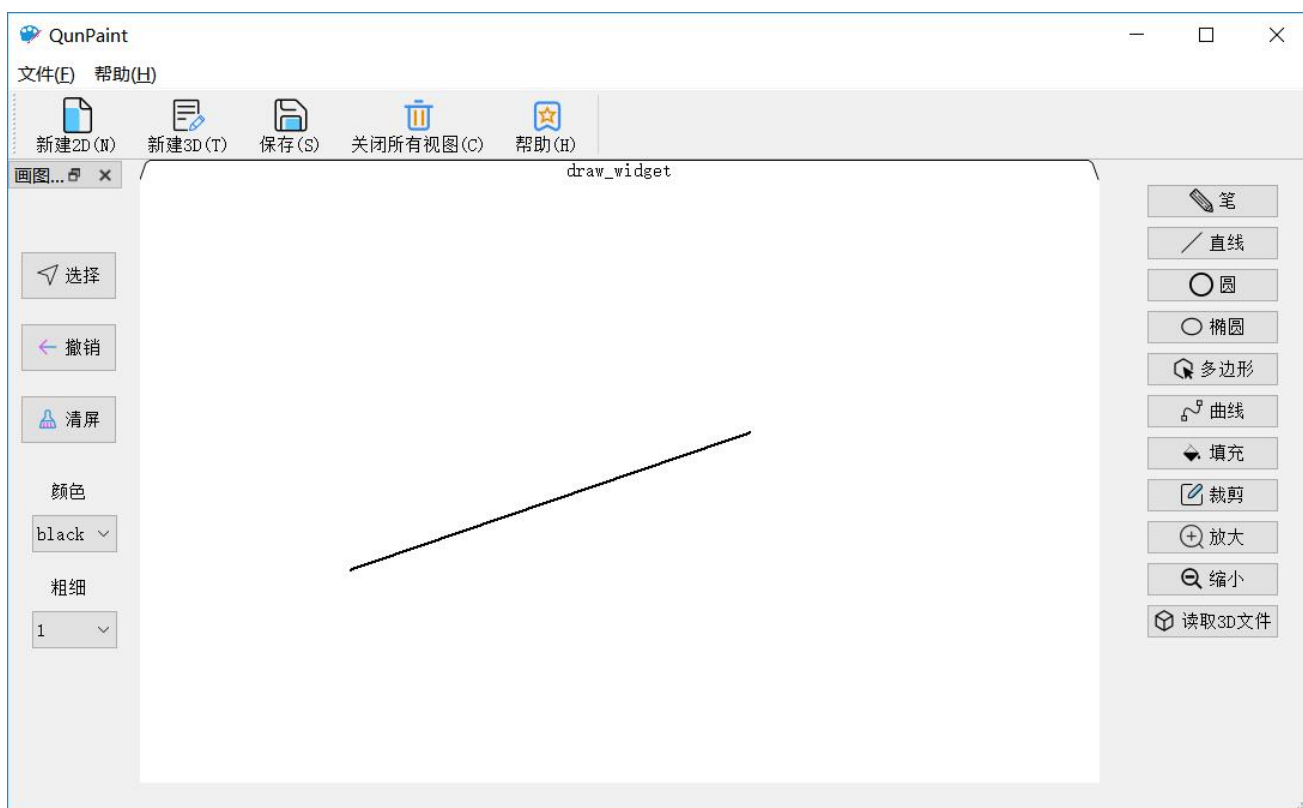
3. 当裁剪范围满意的时候，点击裁剪，按钮，会自动裁剪，并且生成的直线可以继续编辑且可以裁剪，效果如下



4. 直线的裁剪完成

⑤直线绘制完毕

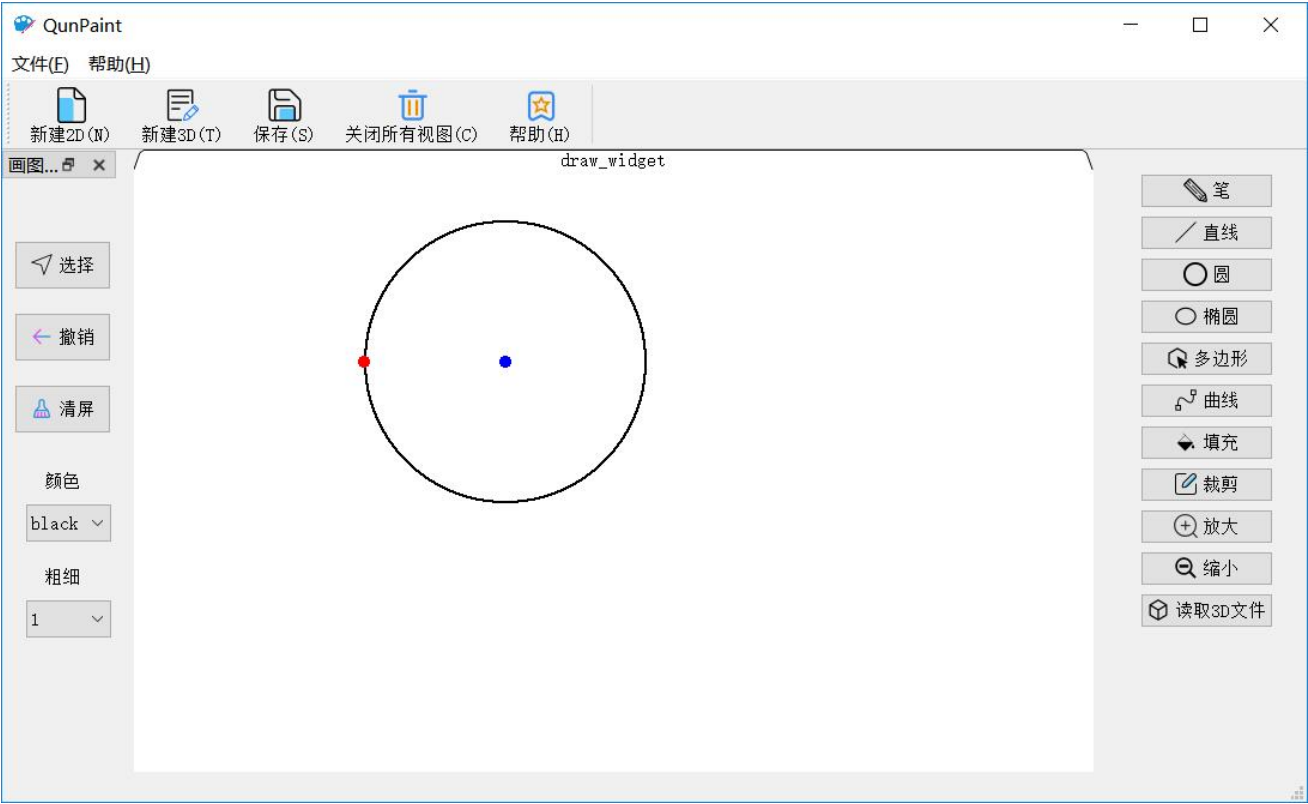
点击除彩色顶点的任意一处，退出编辑模式，完成绘制。



3.圆绘制

①圆的初始化绘制：

点击圆按钮，在画板上点击任意一处，拖动，到任意点松开，完成绘制：



②圆的平移和编辑：

平移：点击蓝色点，拖动，松开完成平移

编辑：点击红色点，拖动，松开完成编辑

缩放：点击放大或缩小按钮，实现缩放功能

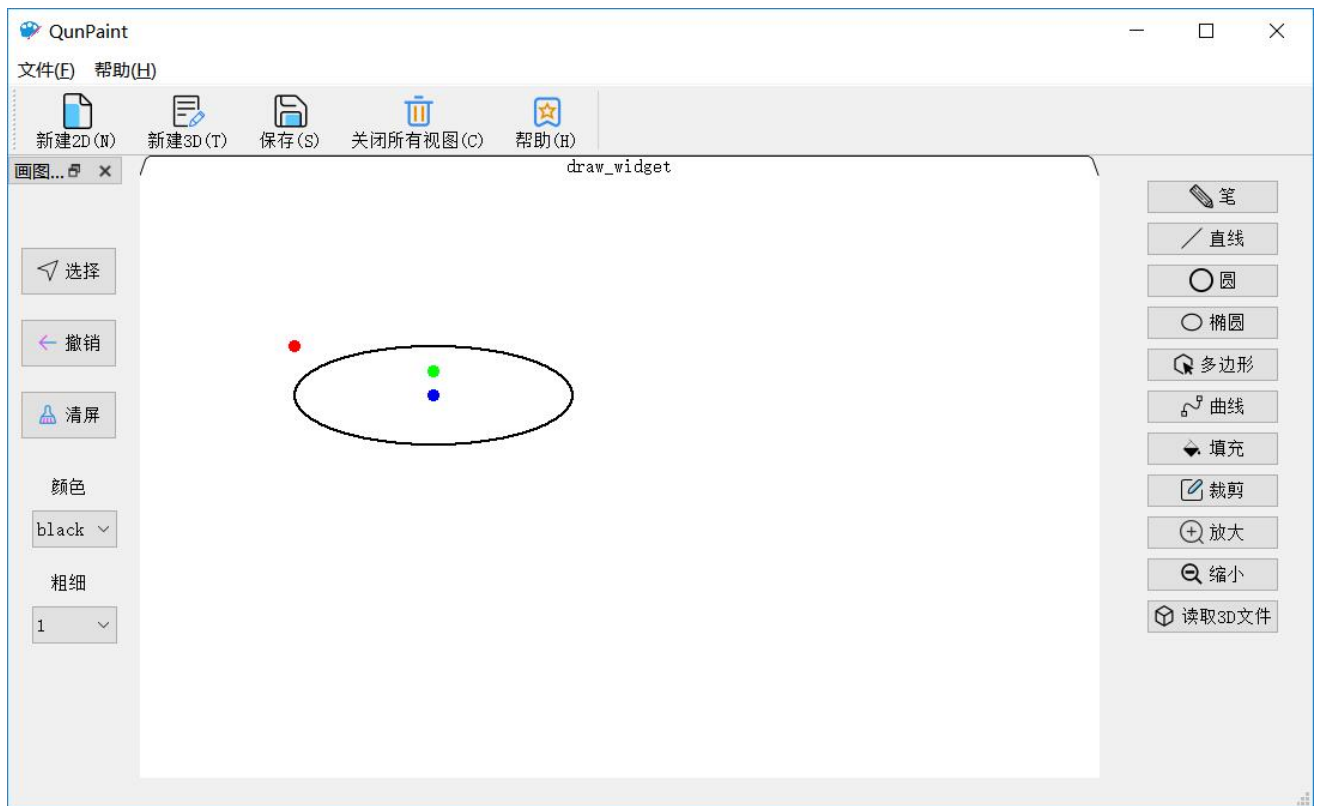
具体图片不上，可见报告中的操作的GIF图

③绘制完毕：点击除彩色顶点的任意一处，退出编辑模式，完成绘制。

4.椭圆绘制

①椭圆的初始化绘制：

点击椭圆按钮，在画板上点击任意一处，拖动，到任意点松开，完成绘制：



②椭圆的平移，旋转和编辑：

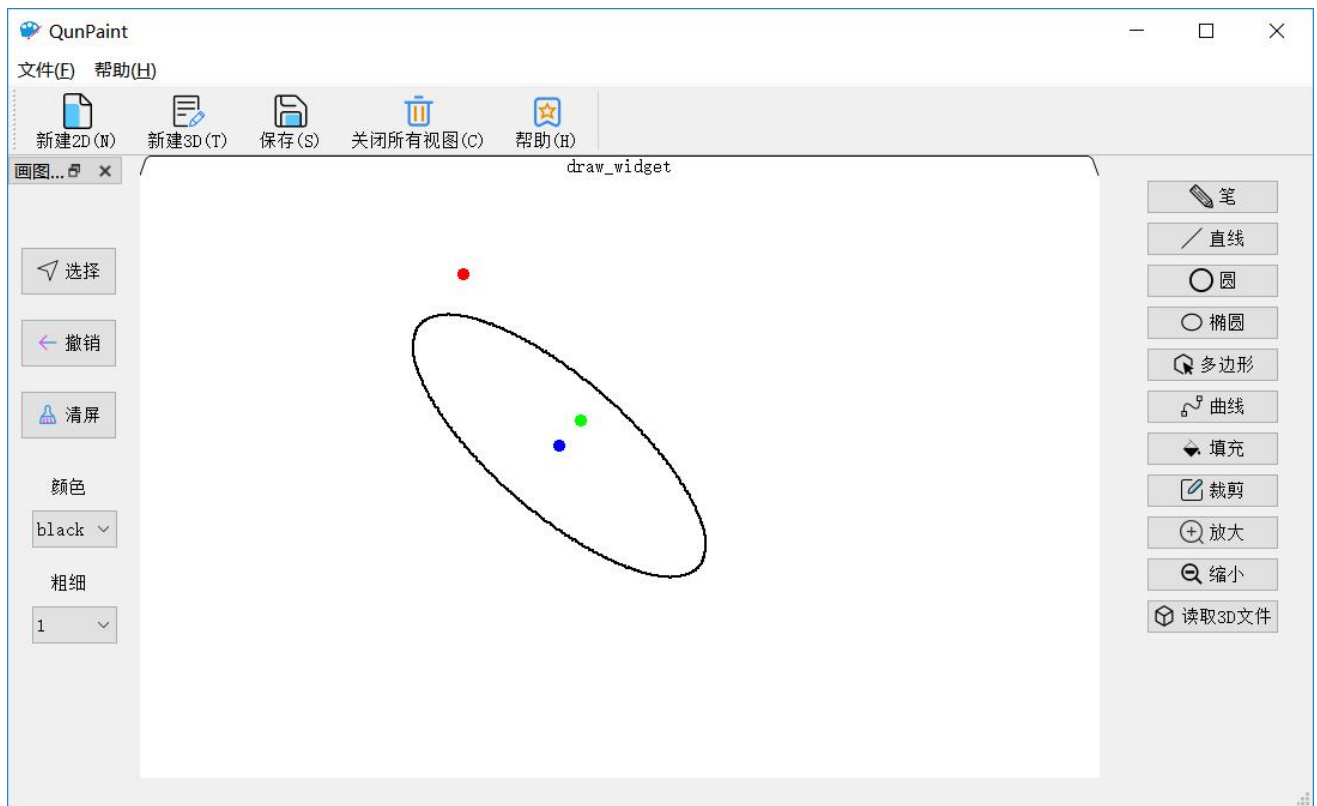
平移：点击**蓝色点**，拖动，松开完成平移

编辑：点击**红色点**，拖动，松开完成编辑

缩放：点击**放大**或**缩小**按钮，实现缩放功能

旋转：点击**绿色点**，拖动，松开完成旋转

下图为经历以上操作最后的效果图：



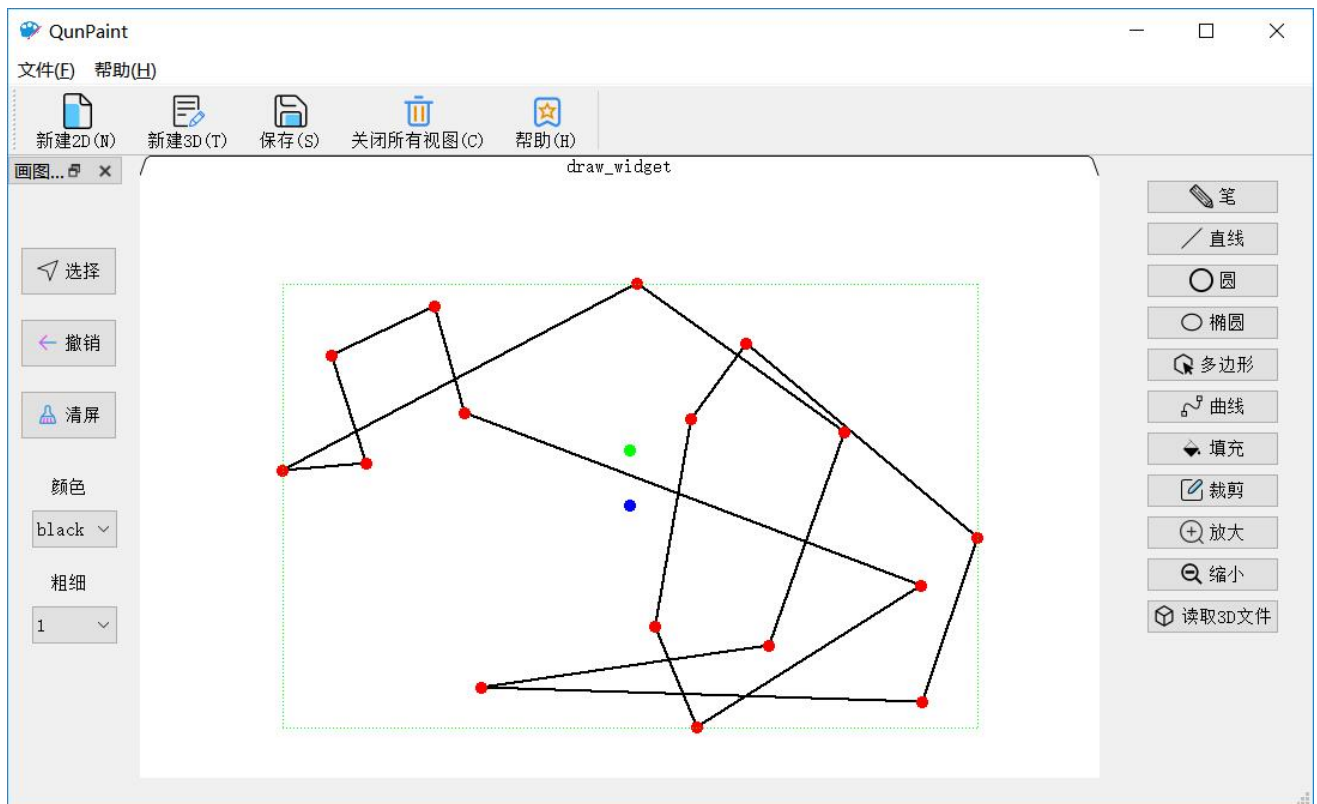
具体可见报告中的操作的GIF图

③绘制完毕：点击除彩色顶点的任意一处，退出编辑模式，完成绘制。

5.多边形的绘制

①多边形的初始化绘制：

点击多边形按钮，在画板上点击一个初始点，选取下一个点，按住鼠标，拖动或者不拖动，松开鼠标，继续选取下一个点，直到最后一个点与第一个点重合，完成绘制，进行显示。效果如下：



②多边形的编辑，变换功能：

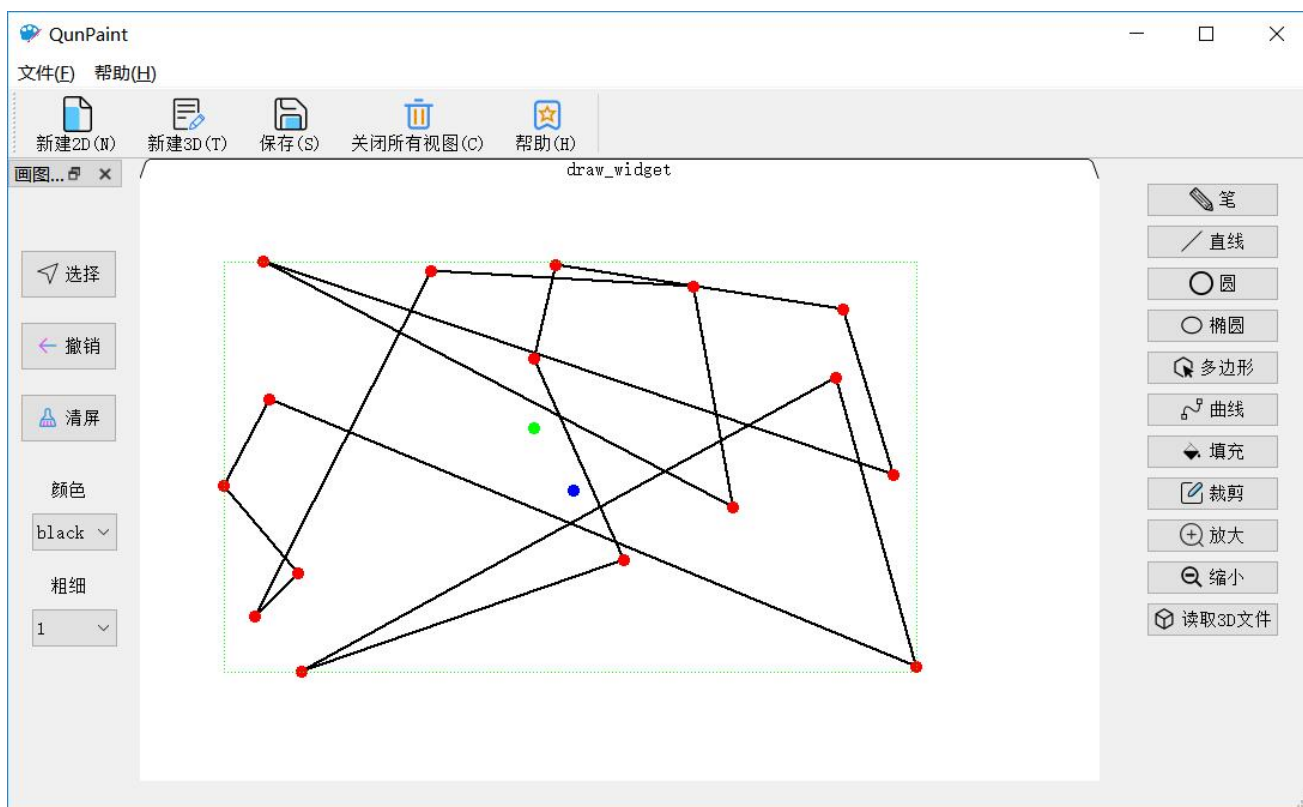
平移：点击**蓝色点**，拖动，松开完成平移

编辑：点击**红色点**，拖动，松开完成编辑

缩放：点击**放大**或**缩小**按钮，实现缩放功能

旋转：点击**绿色点**，拖动，松开完成旋转

下图为经历以上操作最后的效果图：



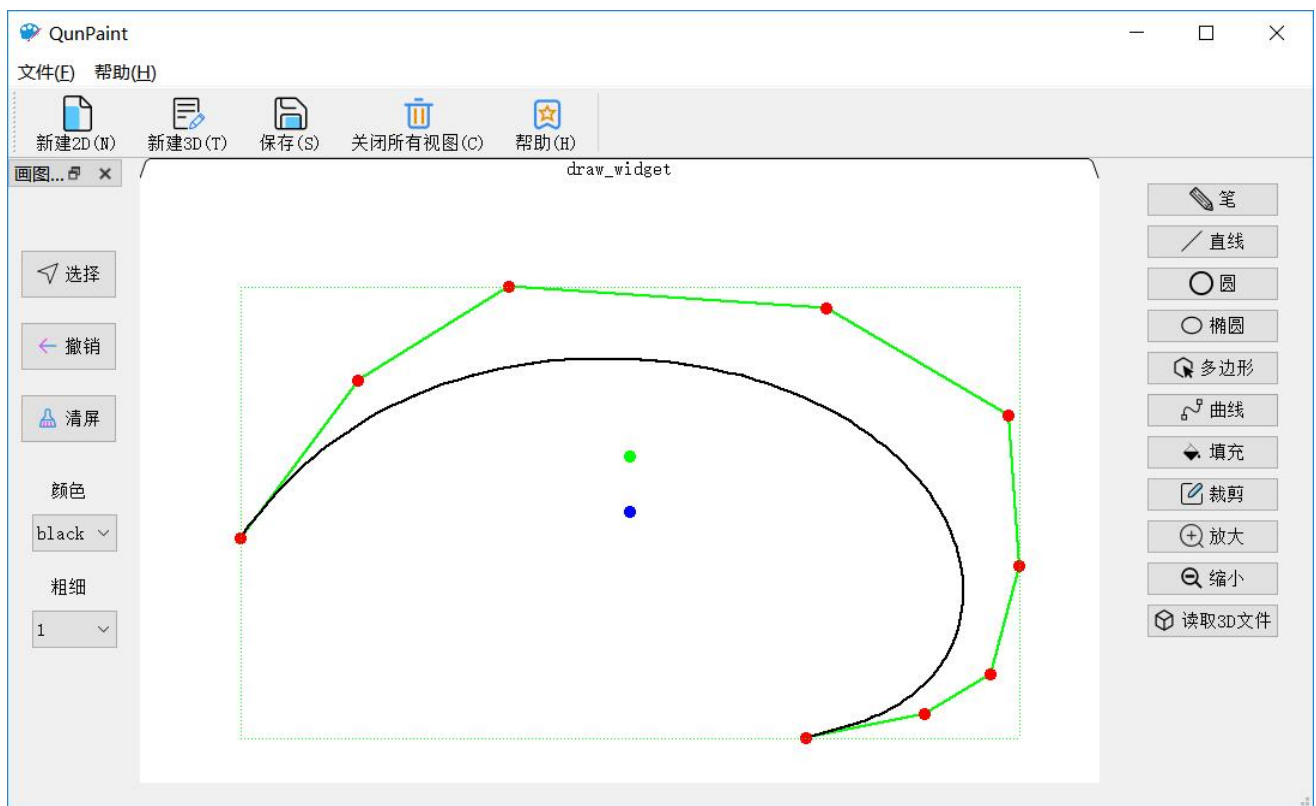
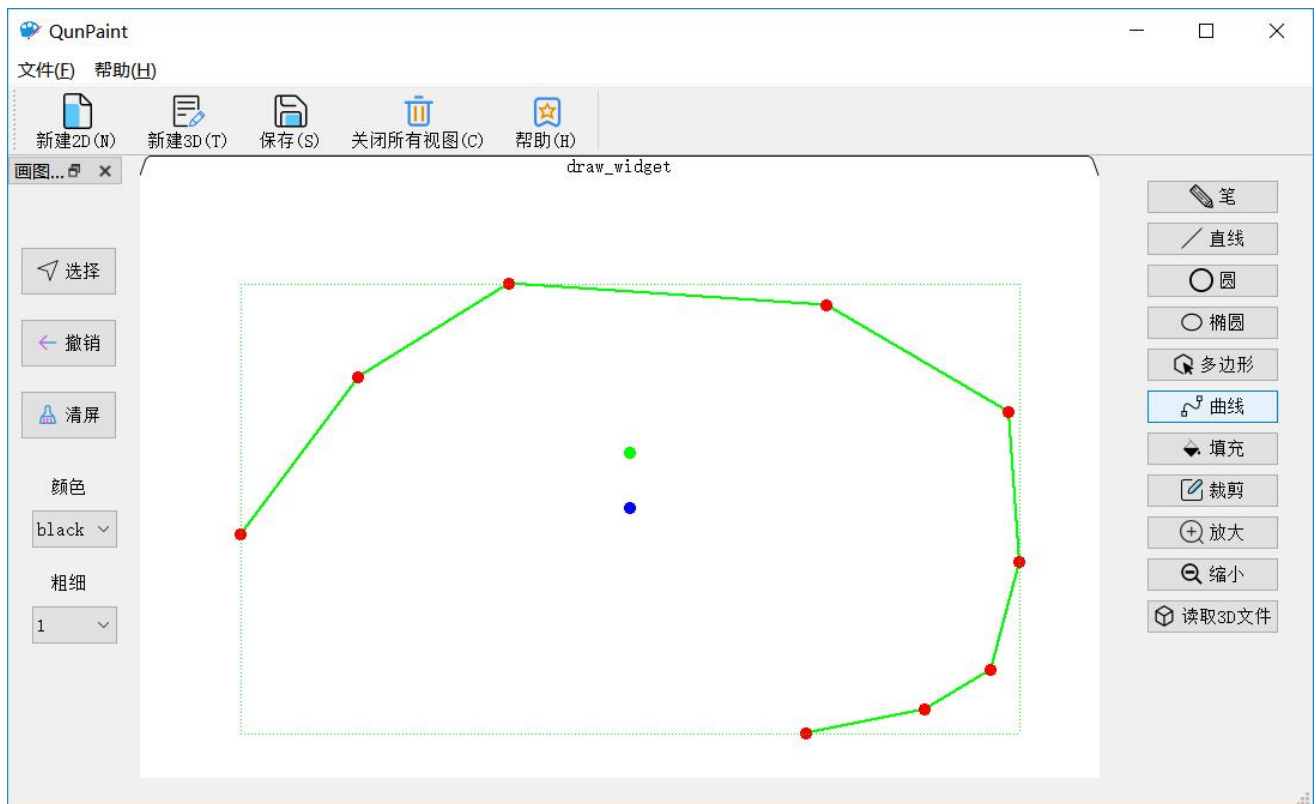
具体可见报告中的操作的GIF图

③绘制完毕：点击除彩色顶点的任意一处，退出编辑模式，完成绘制。

6.曲线的绘制

①曲线的初始化绘制：

点击一个初始点，选取下一个点，按住鼠标，拖动或者不拖动，松开鼠标，继续选取下一个点，等待选择控制点差不多的时候，再次点击曲线按钮，进行绘制。效果如下：



②曲线的编辑，变换功能：

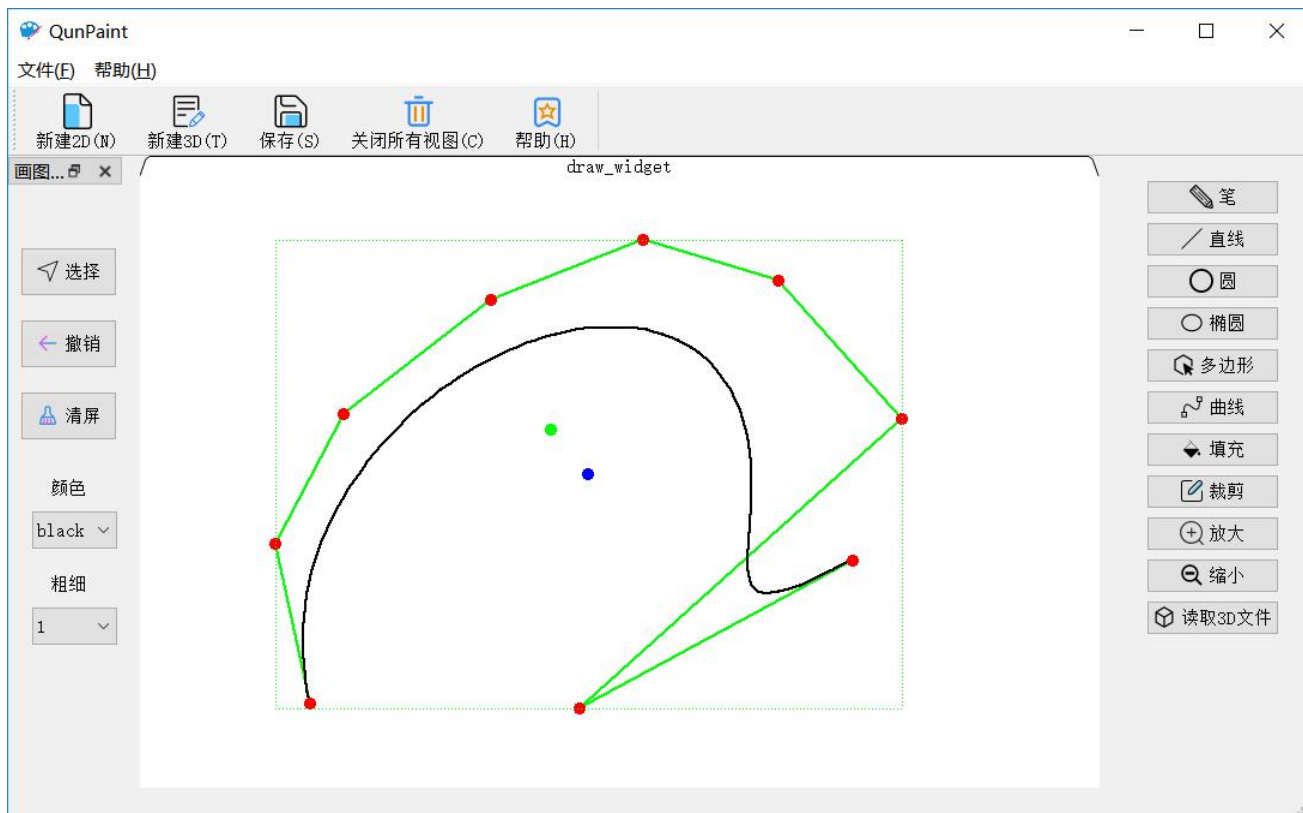
平移：点击**蓝色点**，拖动，松开完成平移

编辑：点击**红色点**，拖动，松开完成编辑

缩放：点击**放大**或**缩小**按钮，实现缩放功能

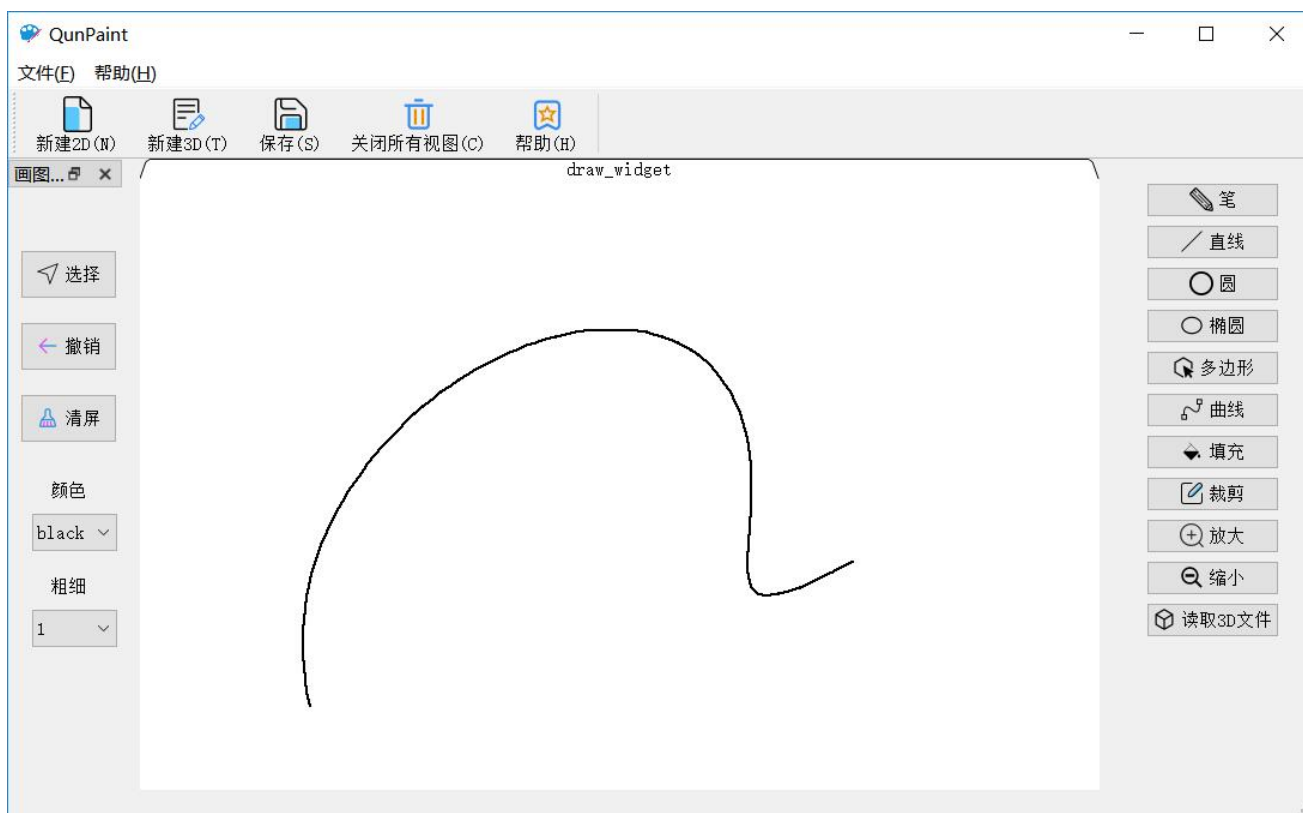
旋转：点击**绿色点**，拖动，松开完成旋转

下图为经历以上操作最后的效果图：



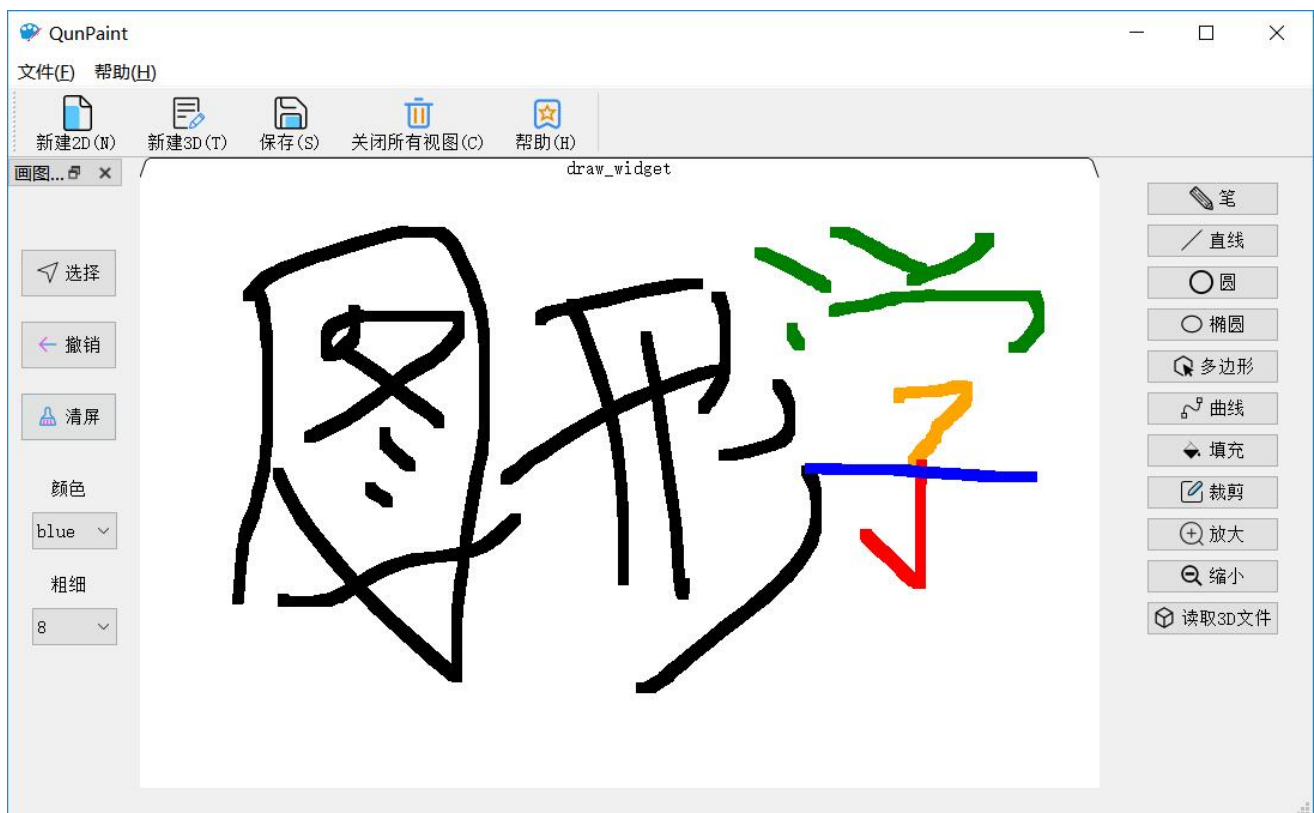
具体可见报告中的操作的GIF图

③绘制完毕：点击除彩色顶点的任意一处，退出编辑模式，完成绘制。



7.笔的操作

①绘制，点击笔按钮，在图上进行随意绘制，没有编辑模式，画完即结束，效果如下：（字丑）

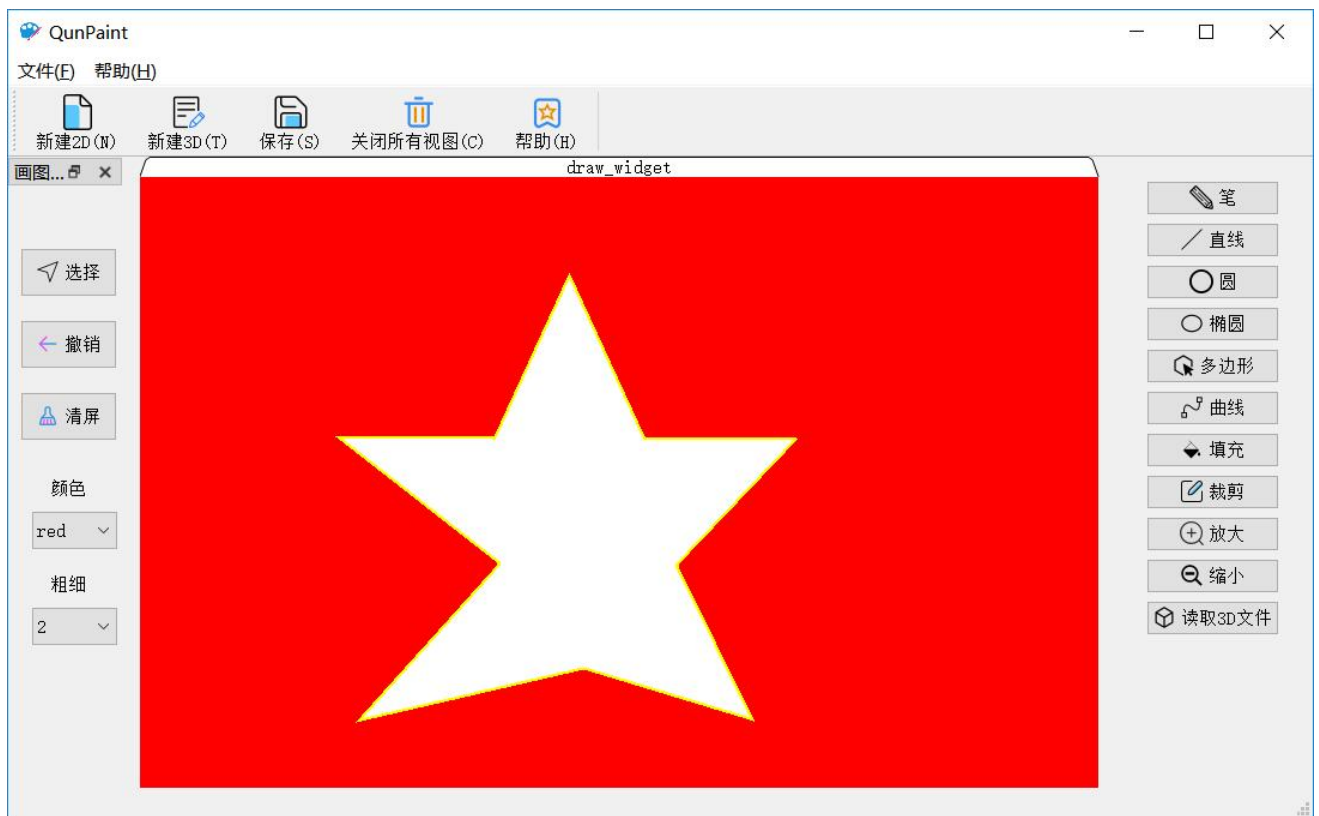
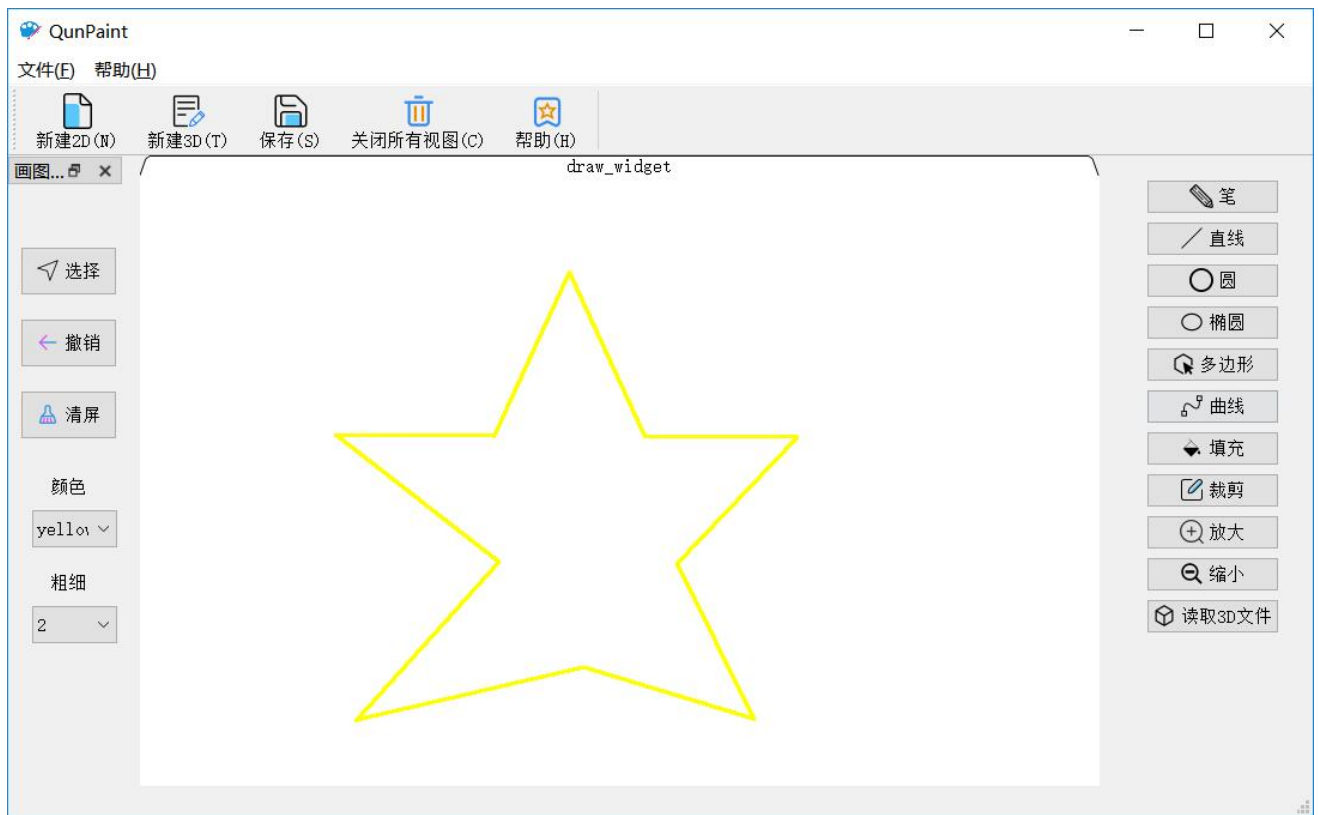


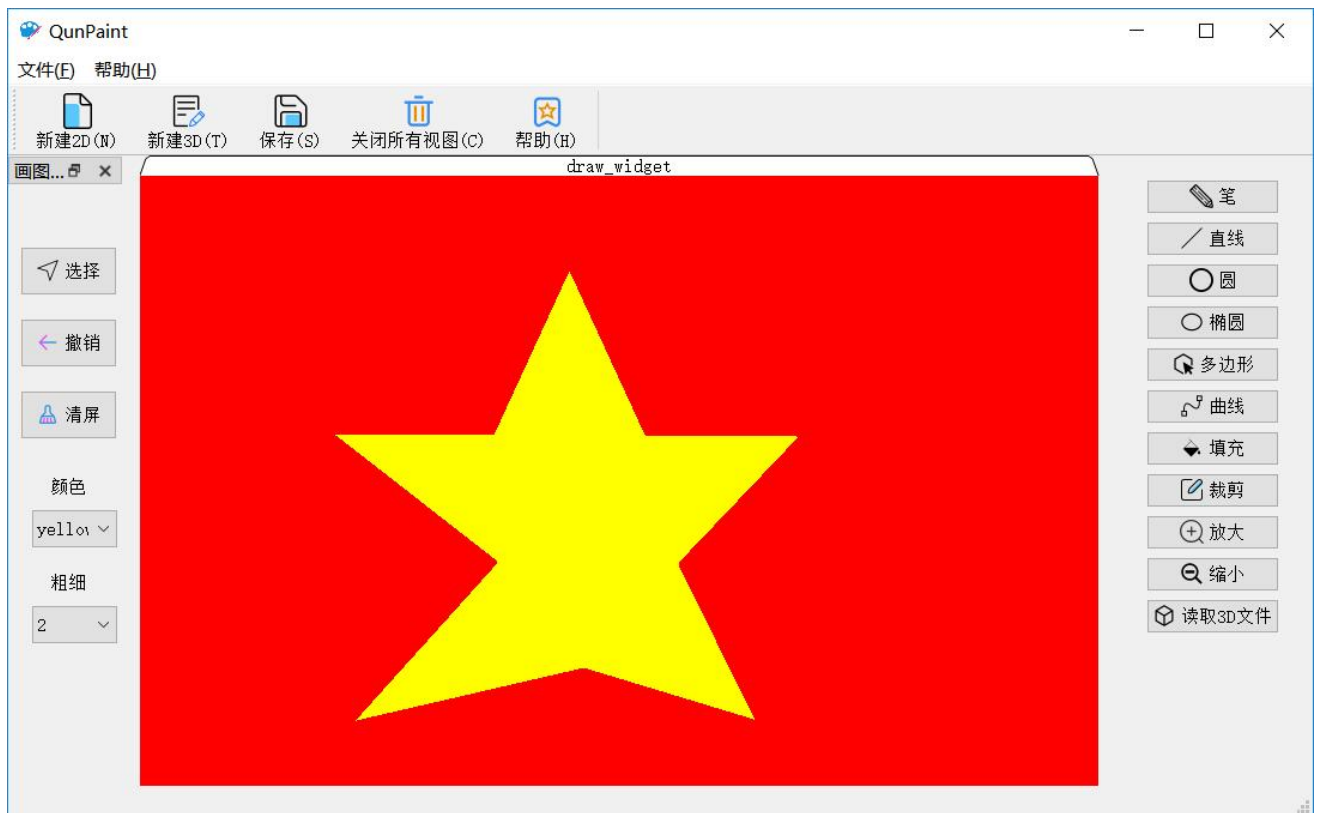
8.填充操作

1.点击填充按钮，选择相应的填充颜色

2.点击填充位置，即可完成填充

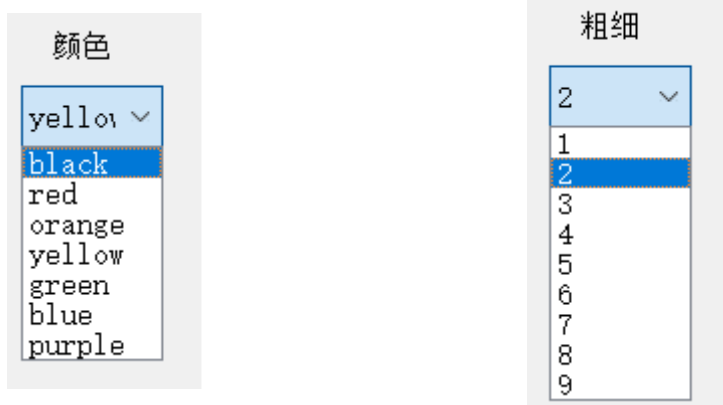
具体的效果如下，所示：





2D图形的功能性按钮：

①颜色，粗细的选择：



点击即可选择颜色粗细，只不过没有进行立即重画，需要进行一次操作编辑，或者完成绘制，才能显示颜色和粗细点的改变，这是一个写的不太好的地方。

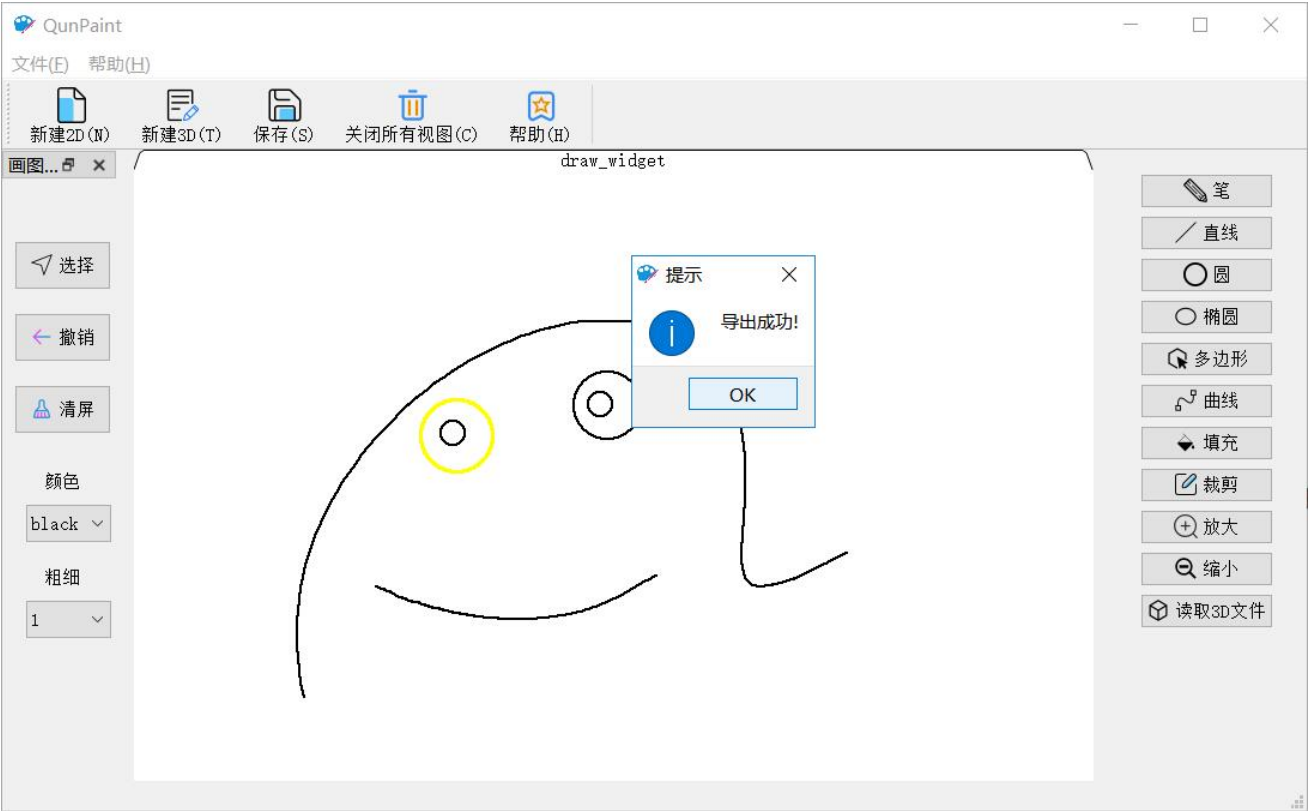
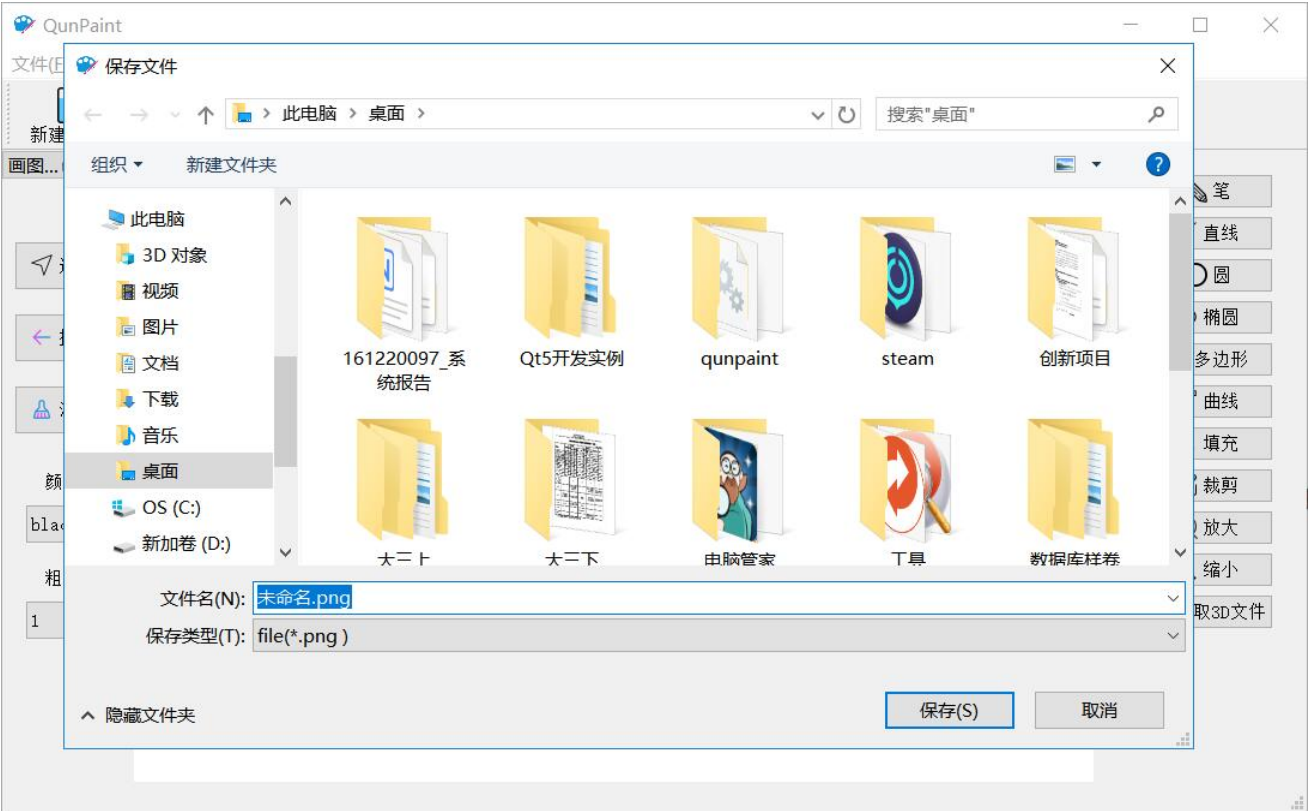
②撤销功能：撤销上一步的绘制图片，画的不好的玩家，可以撤销重新绘制

③清屏：功能清空画布，效果为展示一张新的画布

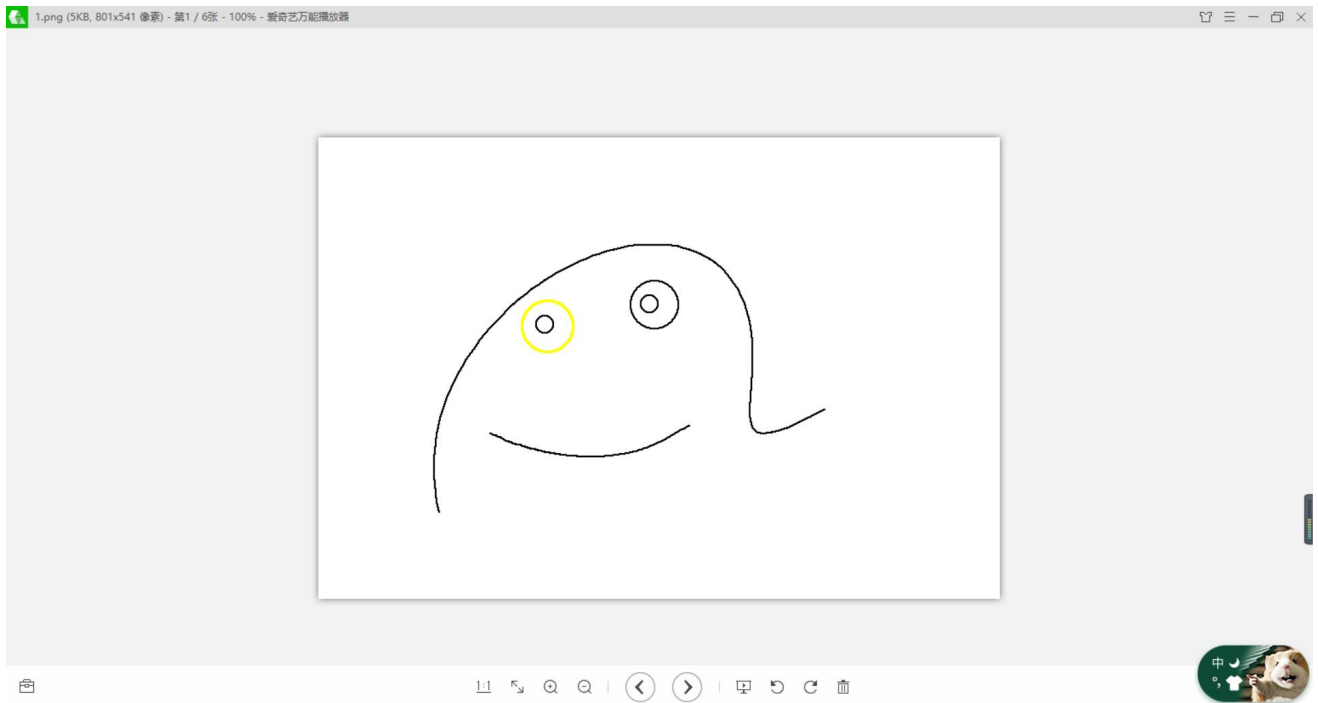
效果见保存的GIF图。

2D图形的保存功能：

步骤：点击保存按钮(快捷键CTRL + S)，弹出对话框，选择保存路径，进行指定输出。效果如下：



桌面上就有刚刚保存的图片，可以用其他图片查看器查看：

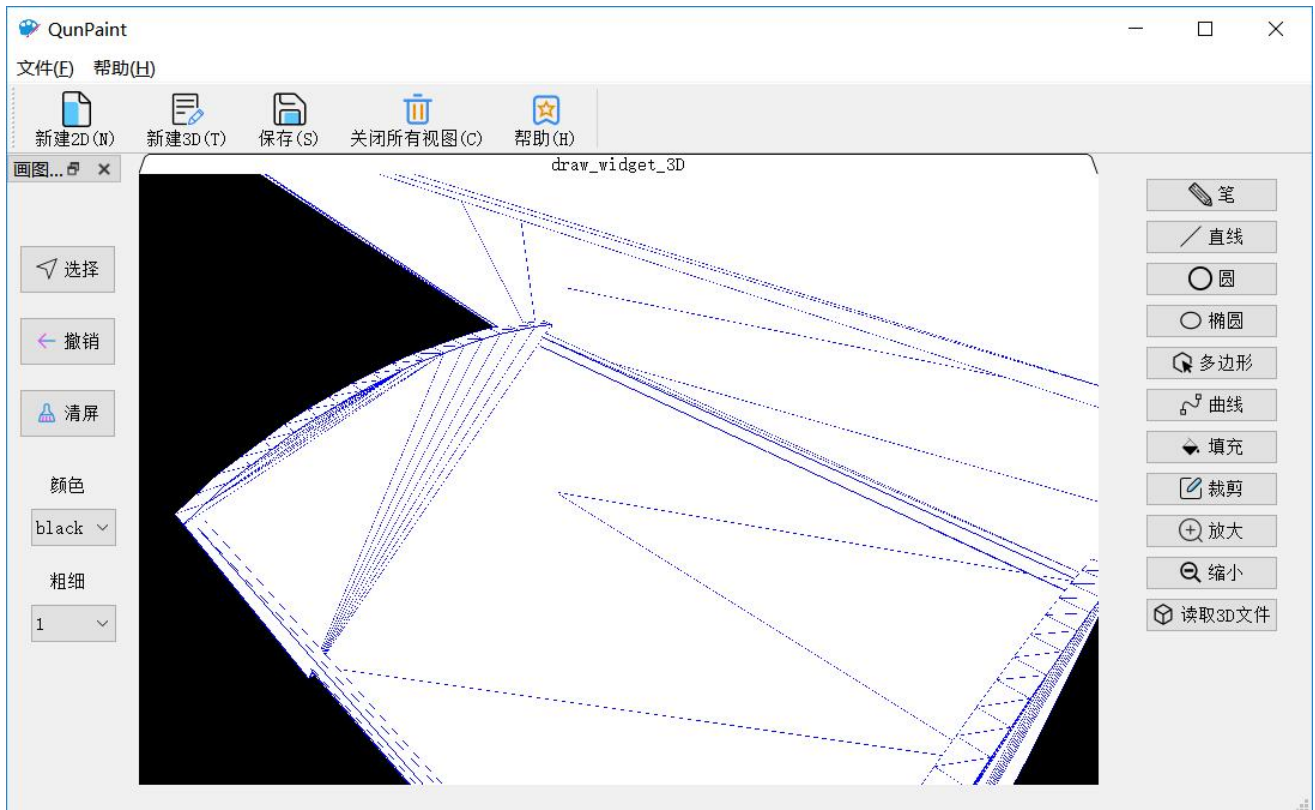
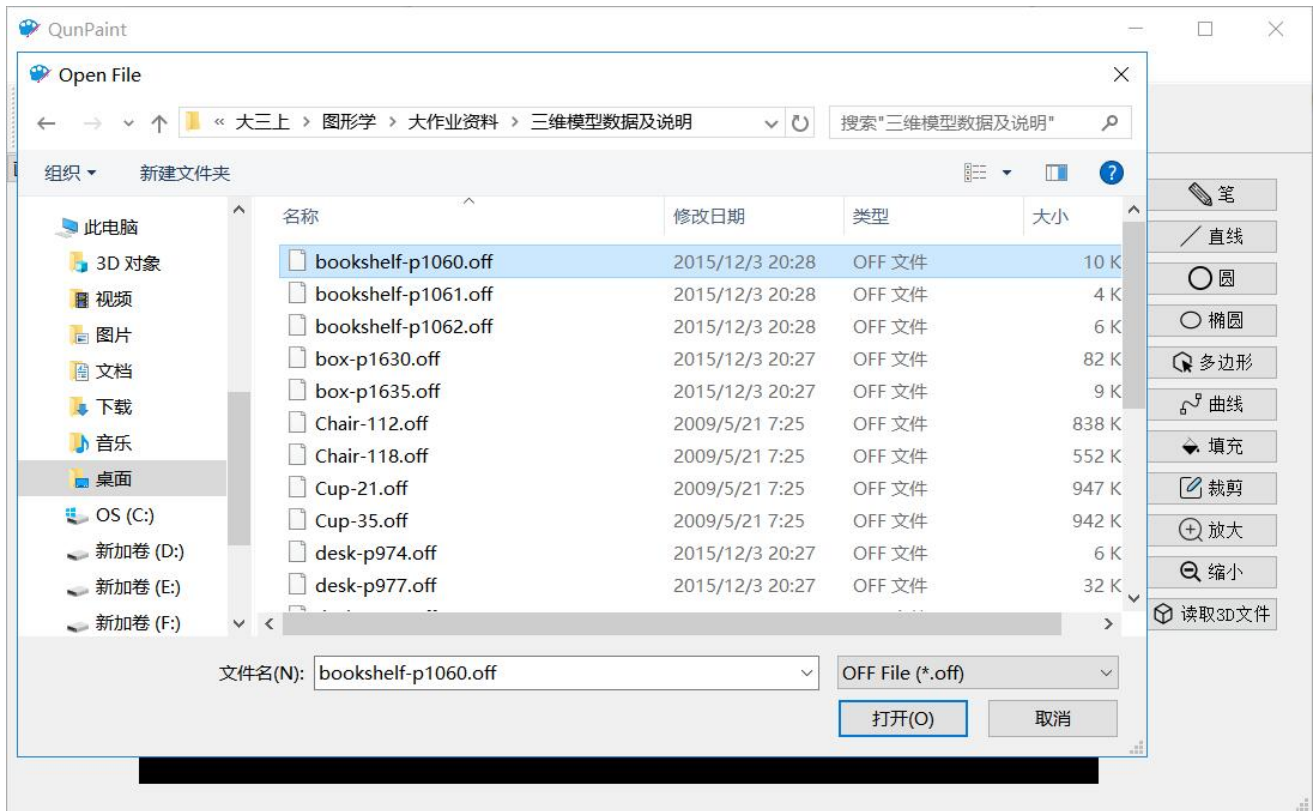


③3D图形的显示功能

步骤如下：

1. 打开一个3D画板
2. 按读取3D文件按钮
3. 读取文件并且显示按钮

效果如下：



是不是感觉不是很清楚，这是因为初始化视角的选择，没关系，提供可以变化的功能键：

```
void my_3Dwidget::keyPressEvent(QKeyEvent *event)
{
    switch(event->key()){
        case Qt::Key_Left:  angle+=5;break;
        case Qt::Key_Right: angle-=5;break;
    }
}
```

```

        case Qt::Key_Up:    offsetZ+=0.5;break;
        case Qt::Key_Down:  offsetZ-=0.5;break;
        case Qt::Key_A:    offsetX+=0.5;break;
        case Qt::Key_D:    offsetX-=0.5;break;
        case Qt::Key_W:    offsetY+=0.5;break;
        case Qt::Key_S:    offsetY-=0.5;break;
    default:break;
}
update();
}

```

首先必须点击一下画板区域! :

键盘AD, 三维坐标x偏移加减0.5;

键盘WS, 三维坐标y偏移加减0.5;

键盘↑↓, 三维坐标z偏移加减0.5;

键盘←→, 三维坐标旋转;

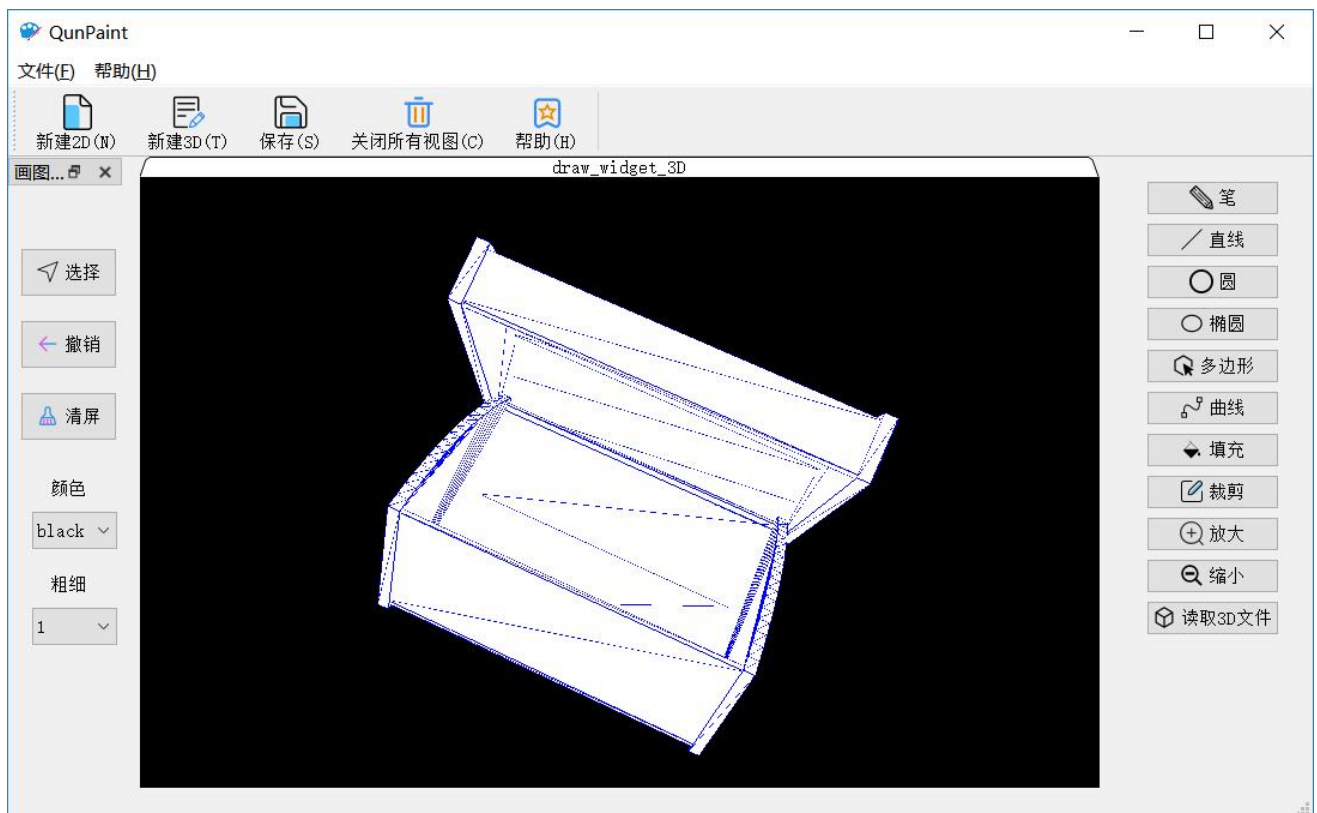
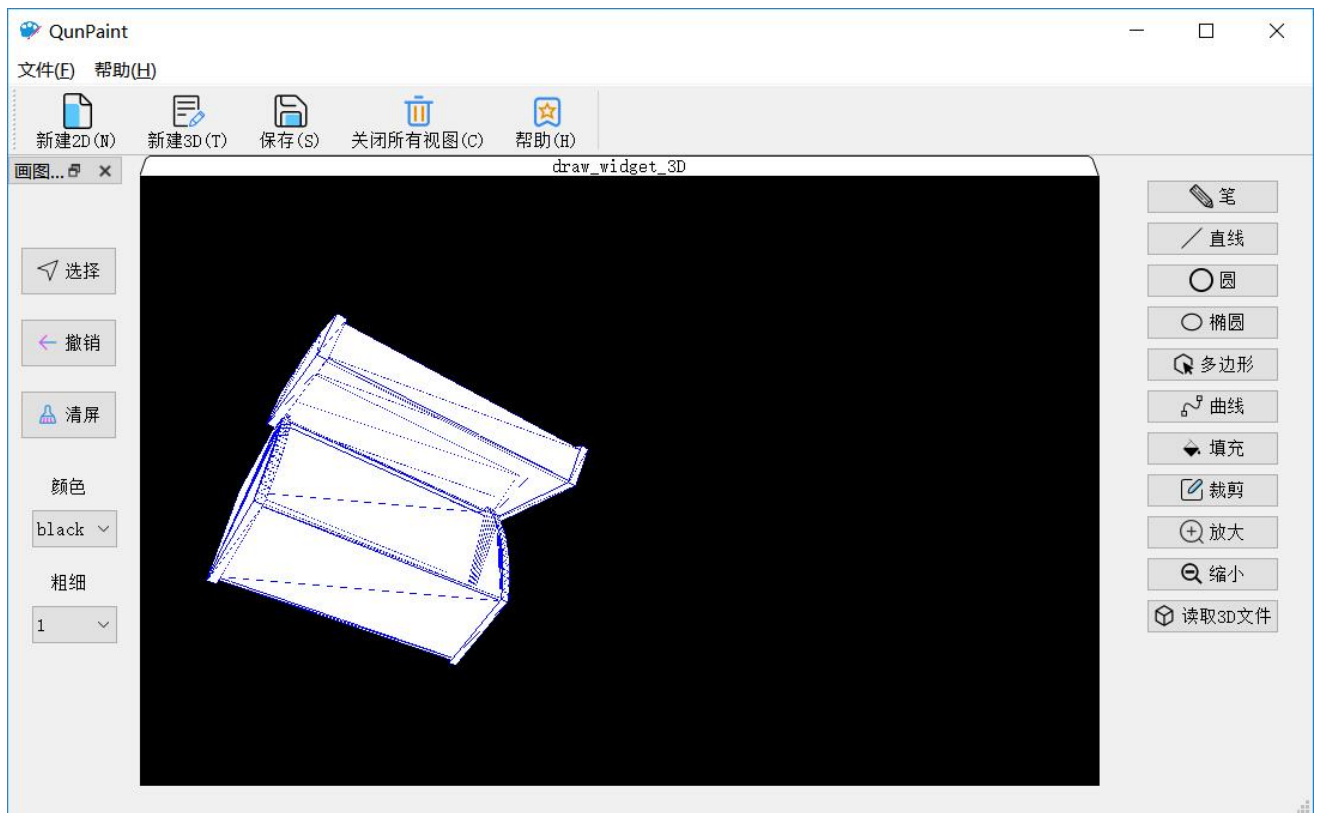
```

void my_3Dwidget::paintGL()
{
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
    glMatrixMode(GL_MODELVIEW); //模型视图
    glLoadIdentity(); //恢复初始坐标系
    glTranslatef(offsetX, offsetY, offsetZ); //更改坐标系
    glRotatef(angle, 0,0.5,0); //按角度在旋转
    gluLookAt(1.0, 1.0, 1.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0);
    this->draw3D(); //绘制3D图形
}

```

可以知道, 旋转是按(0,0.5,0)坐标旋转的, 是从(1,1,1)位置观察的, 所以不一定很正, 但是我们可以通过以上按钮进行调节, 来更好的观察3D图形。

下面是经过操作后, 不同视角下的图形:

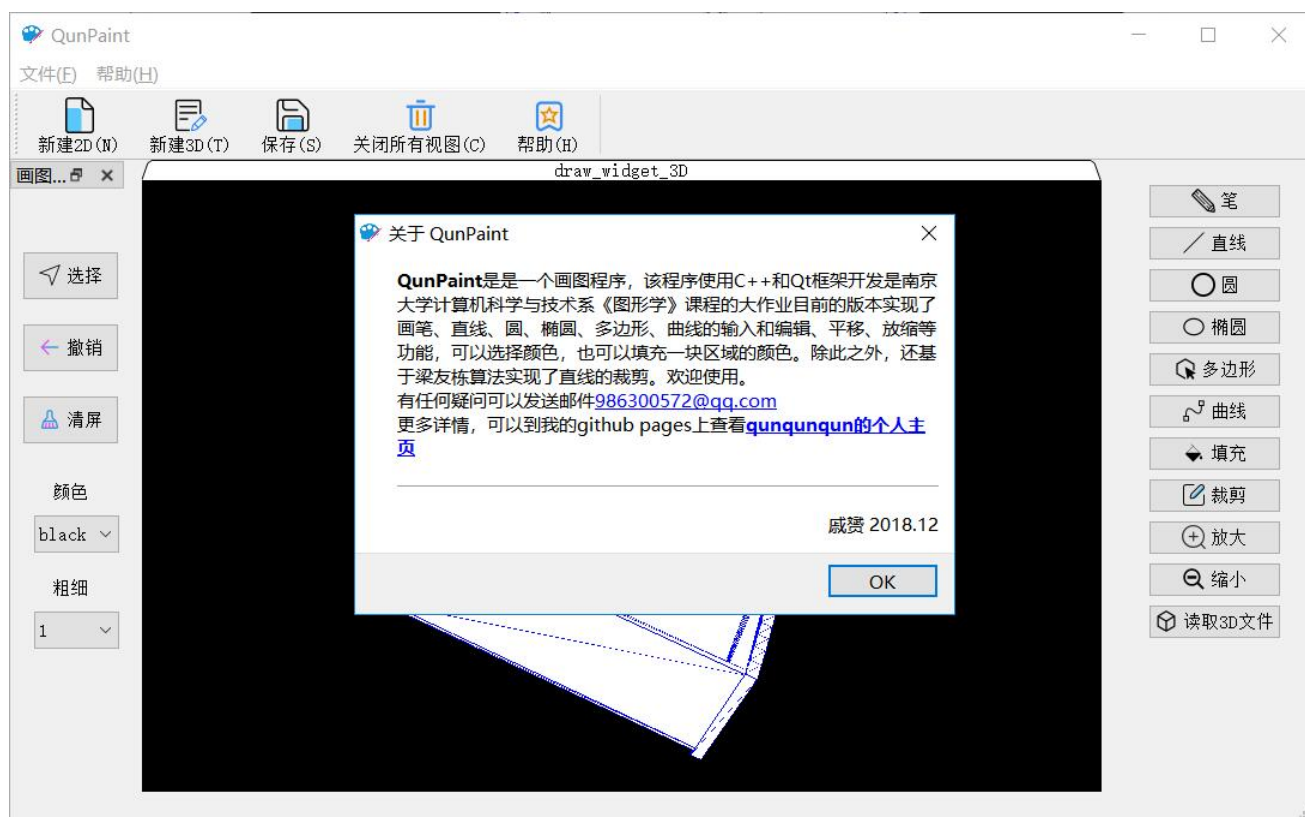


仍可以继续打开文件进行显示新的3D模型的操作。

④其他额外功能和设计

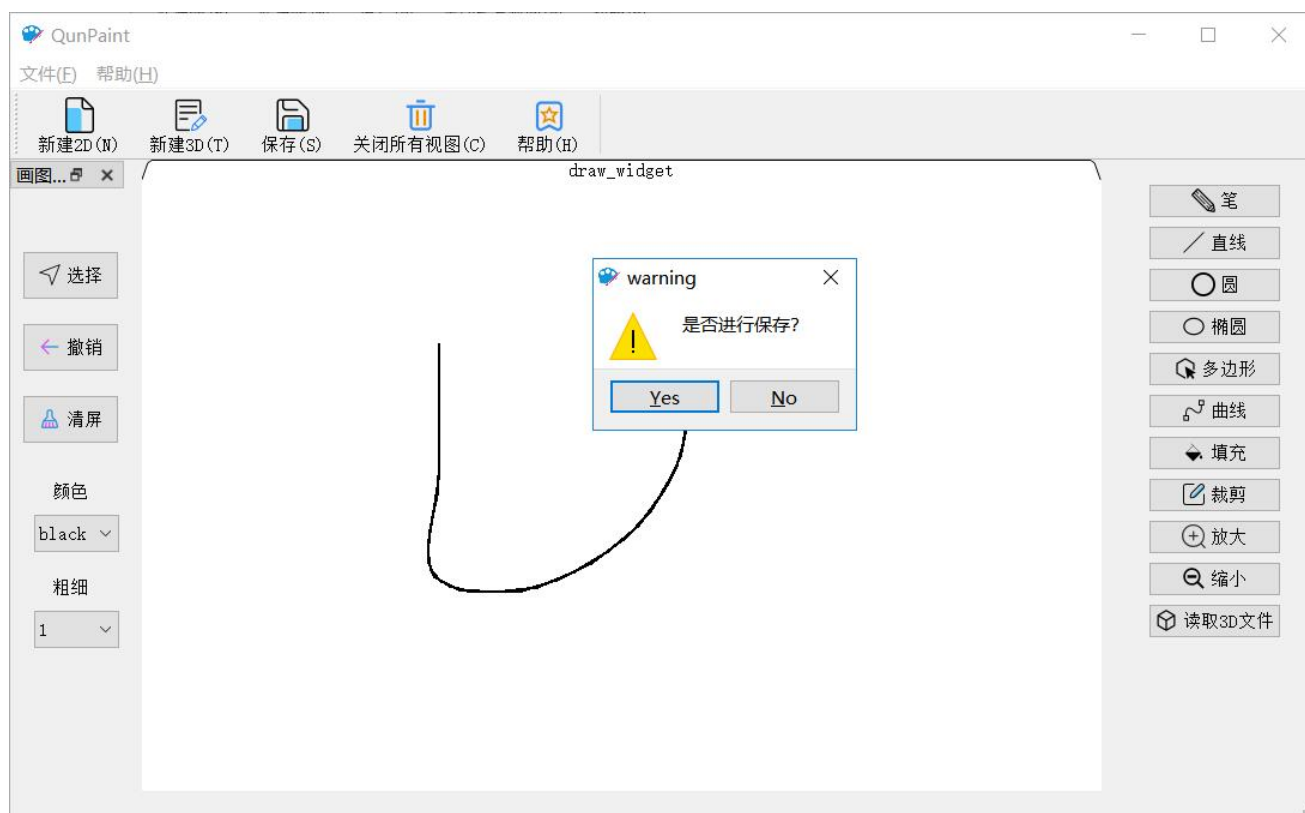
1.帮助按钮：

点击帮助(Ctrl + H), 会显示程序说明, 我的联系方式和我的GITHUB主页, 程序源码会在截止日期后存放进GITHUB上。点击蓝色连接可以打开我的Github主页



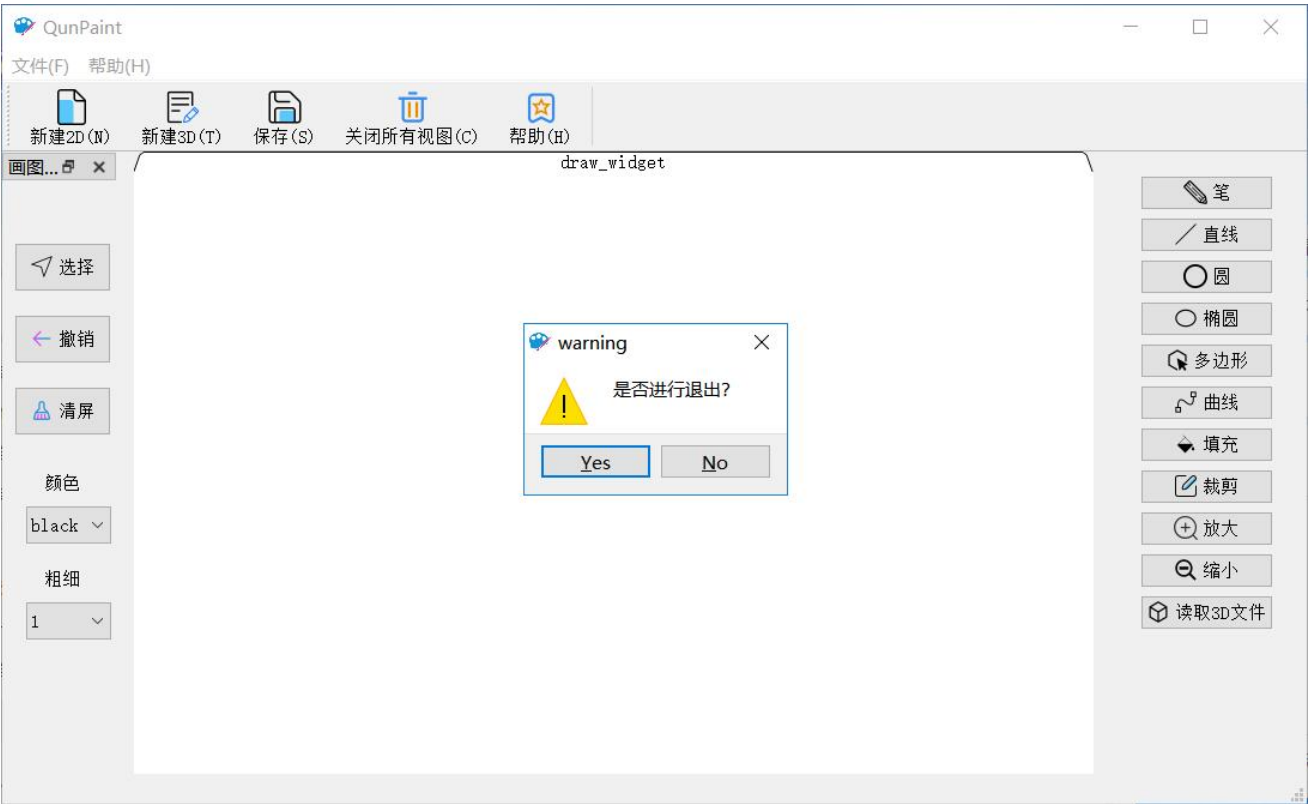
2.关闭视图提示保存

点击关闭2D视图, 会有提示保存的操作, 从而使用户意外点击关闭的不会损失当前的绘图。



3.关闭程序

点击关闭程序按钮，会有提示是否退出操作。



这些设计提高了用户的良好用户体验，使得我的软件设计更像一个画板了。

三、结束语

一些操作已经截成GIF图中随实验一起提交，可以点击GIF图进行查看！

三个月的努力下，我的软件终于完成了。整个系统的开发过程中几乎涵盖了课堂上所教授的大部分图形生成算法，整个实现的过程使我充分地认识到了理论学习与上机实现的差距，在上机实现实现算法的过程中遇到的种种困难也大大加深了我对相应的算法的理解。总的来说，这些基础而又重要的图形学算法使我对计算机图形学有了新的认识和看法。感谢图形学，领我进入了一个新的知识领域，并且让我学会将知识转化成现实的应用，让我受益匪浅！致谢 在此,我向图形学老师的严谨教学及助教们的工作表示感谢！

对于程序中，操作过快或者不当，可能会出现卡死的状态，但是基本没有什么明显严重的BUG，卡死之后可以重新执行.exe文件即可。

此外，如果对于QunPaint有任何问题和困惑，请联系我！

手机：14760821006 QQ：986300572 邮箱:986300572@qq.com

戚赞 2018年12月