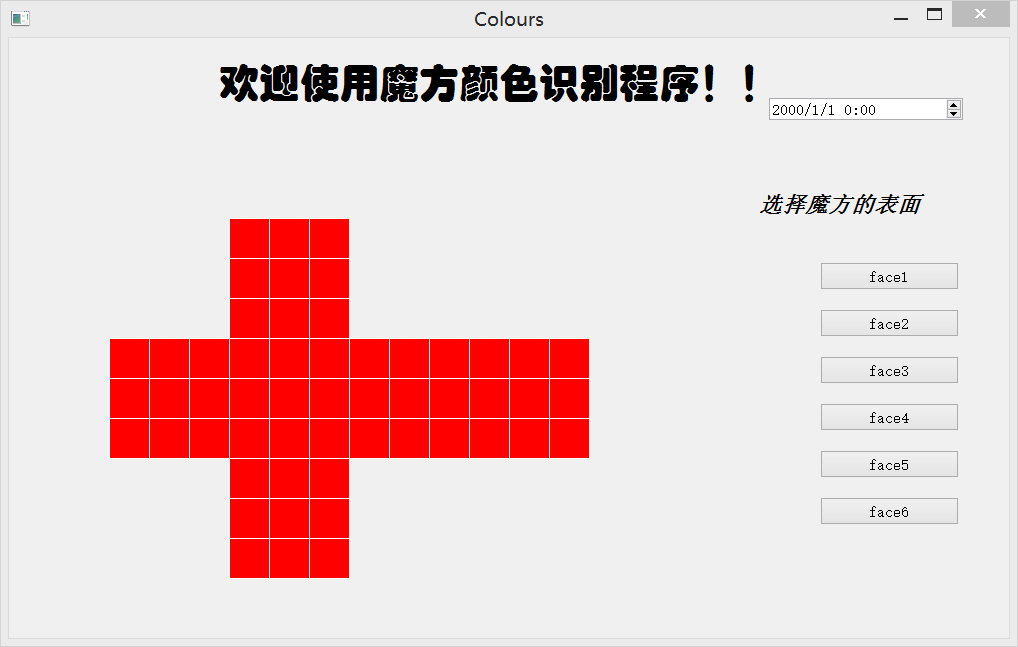
实验九

第二周实验报告

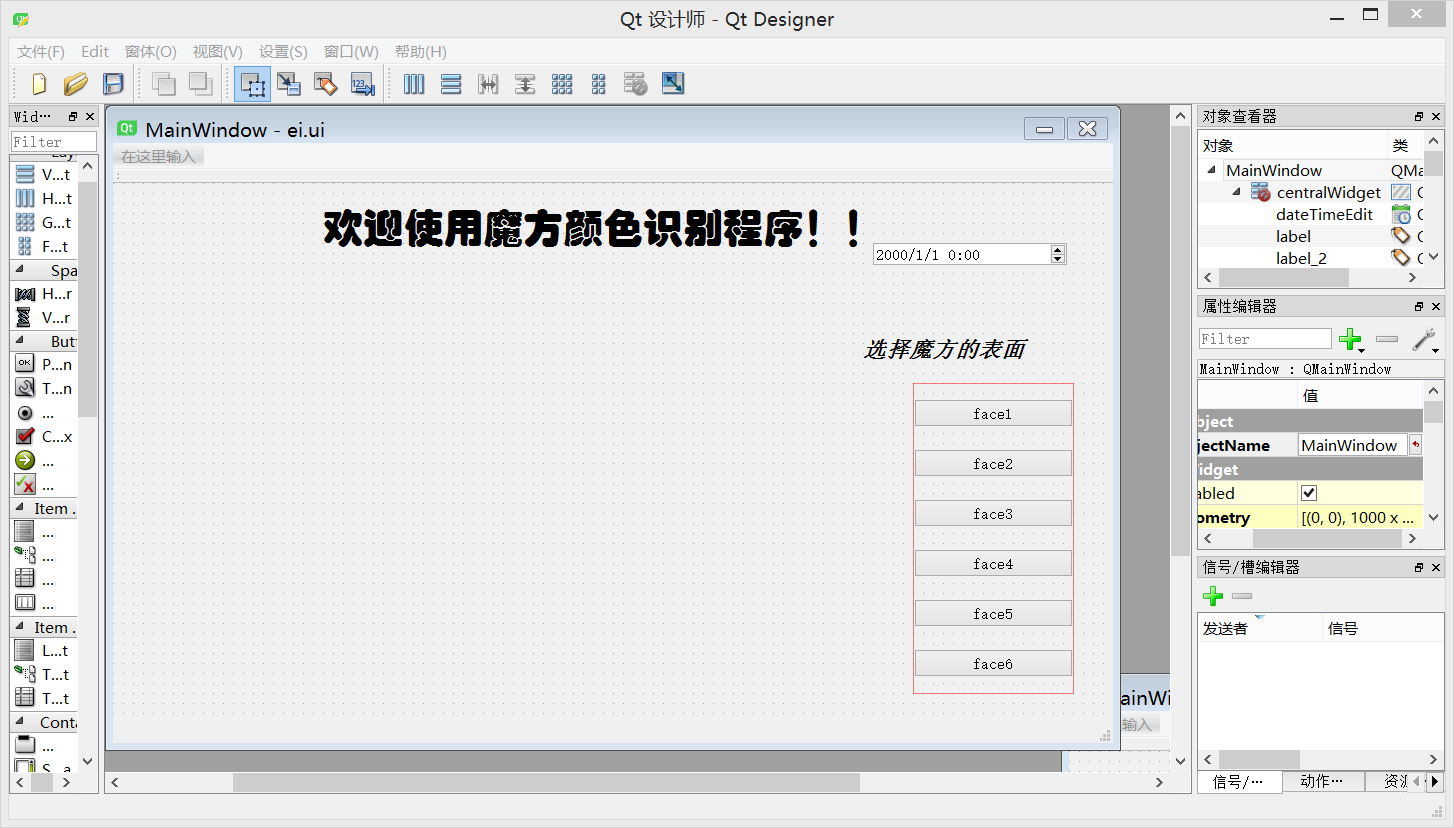
李彦瑞3140102743

这一周主要对魔方识别系统的GUI界面进行了设计，设计采用的环境为python+pyqt5。最终完成的效果图如下所示：



以下分步骤对GUI界面的建立进行介绍：

1. 用QT designer设计界面



如上图所示在qt中创建了widget，在上面放上了一些文字和六个按钮。

计划这每个按钮用来控制选择哪一个面读取了图片数据，然后通过上一个星期做的魔方颜色识别程序将每个面的9个方块的颜色返回一个列表，最后使用widget的paintevent函数调用图片处理返回的列表，将魔方的颜色显示在widget上。

1. 将qt得到的ui文件转化为py文件

在命令行中输入： pyuic5 –o 输出文件名.py 输入文件名.ui

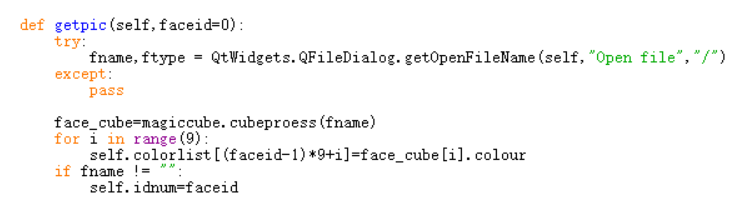
可以将画出的ui文件自动生成一个.py的文件，之后对生成的文件中的类进行实体化以及显示，可以显示出窗口页面。

1. 设置按钮的信号槽

self.face1.clicked.connect(lambda:self.getpic(1))

这行程序将face1按钮被按下的时候触发函数：self.getpic并将面的编号1传入函数中

1. 编写getpic函数



这段函数首先用

fname,ftype = QtWidgets.QFileDialog.getOpenFileName(self,"Open file","/")

创建了一个打开文件的窗口，选择对应的图片文件，将文件地址返回到fname

之后调用了图片处理函数，并将fname作为参数传入了调用函数中

之后调用函数调用该地址打开图片文件进行处理，并返回包含每个面颜色的列表赋给变量face\_cube。

最后通过简单的处理，将颜色信息放到self.colorlist中。

1. paintevent（）函数编写

paintevent函数是pyqt5下的类QWidget比较特殊的一个函数，这个函数在类的内部被循环调用，实时刷新，因此可以把画图的功能都写在这个函数中。

我是通过以下的方法来定义画图的位置的：

首先每个面由9个大小为40\*40像素的小格组成，整个图像有六个面，设这六个面的编号分别是123456，那么定义了

startx=[100,220,340,460,220,220]

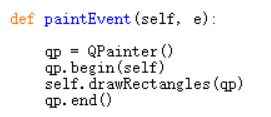
starty=[300,300,300,300,420,180]

xx=[0,0,0,40,40,40,80,80,80]

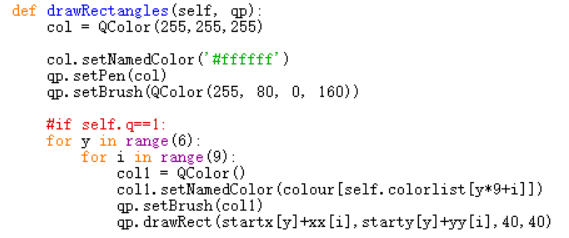
yy=[0,40,80,0,40,80,0,40,80]

其中startx和starty定义了每个面开始画图的坐标

之后的xx和yy定义了每个面中每个小方格的相对坐标，最后在画图时只需要循环6\*9次就可以吧总共的54个小方块都画出，具体的程序如下



主要在程序中调用了drawRectangle函数，该函数如下图所示



用qp.drawRect(startx[y]+xx[i],starty[y]+yy[i],40,40)在widget上画图

其中的startx[y]+xx[i],starty[y]+yy[i]确立了小方格的位置，40,40确立了大小

col1.setNamedColor(colour[self.colorlist[y\*9+i]])

qp.setBrush(col1)

这两行程序确定了画出的小方格的颜色，可以保证在识别的颜色改变之后画出的颜色也能改表。

至此，界面的简单设计与功能已经完成，通过打开图片可以获得每个小方格的颜色并在窗口中表达出来。之后会根据实际要求再做修改。

