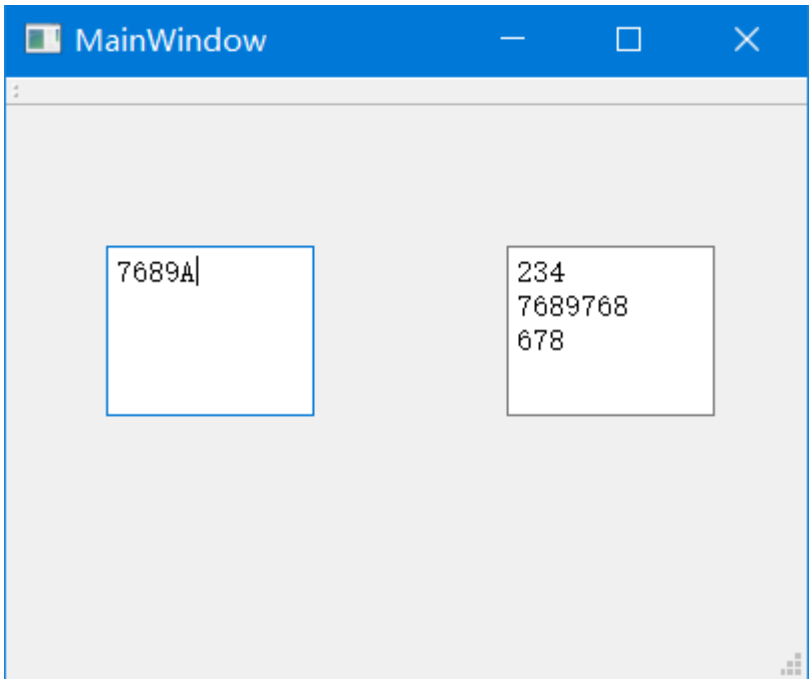
第五次作业文档

一.第一题

1.实验要求:

用Qt实现界面，在Edit1中输入文字后按回车键将文字添加到Edit2中并将Edit1清空

2.运行示例:



3.代码分析:

本题较为简单，使用了Qt Designer进行UI设计。主窗口为MainWindow类，使用，主要功 能实现采用了eventFilter方法。在该方法中对于textEdit对象注册了一个eventFilter用于过滤其事件，对事件的种类和具体按键进行了判定，当确定回车时，执行操作（写入textEdit2文本并清空自身文本）， 该方法代码实现如下:

```
bool MainWindow::eventFilter(QObject *obj,QEvent *event)
{

    if(obj==ui->textEdit)
    {
        if(event->type()==QEvent::KeyPress)             //判定事件种类
        {
            QKeyEvent *keyEvent =(QKeyEvent*)event;
            if(keyEvent->key() == Qt::Key_Enter || keyEvent->key() == Qt::Key_Return)
                                    //判定按键是否回车
            {
                ui->textEdit_2->append(ui->textEdit->toPlainText());
                ui->textEdit->setText("");
                return true;
```

```
            }
            else {return false;}
        }
        else {return false;}
    }else
    {
        return QMainWindow::eventFilter(obj,event);        //返回主程序
    }
}
```

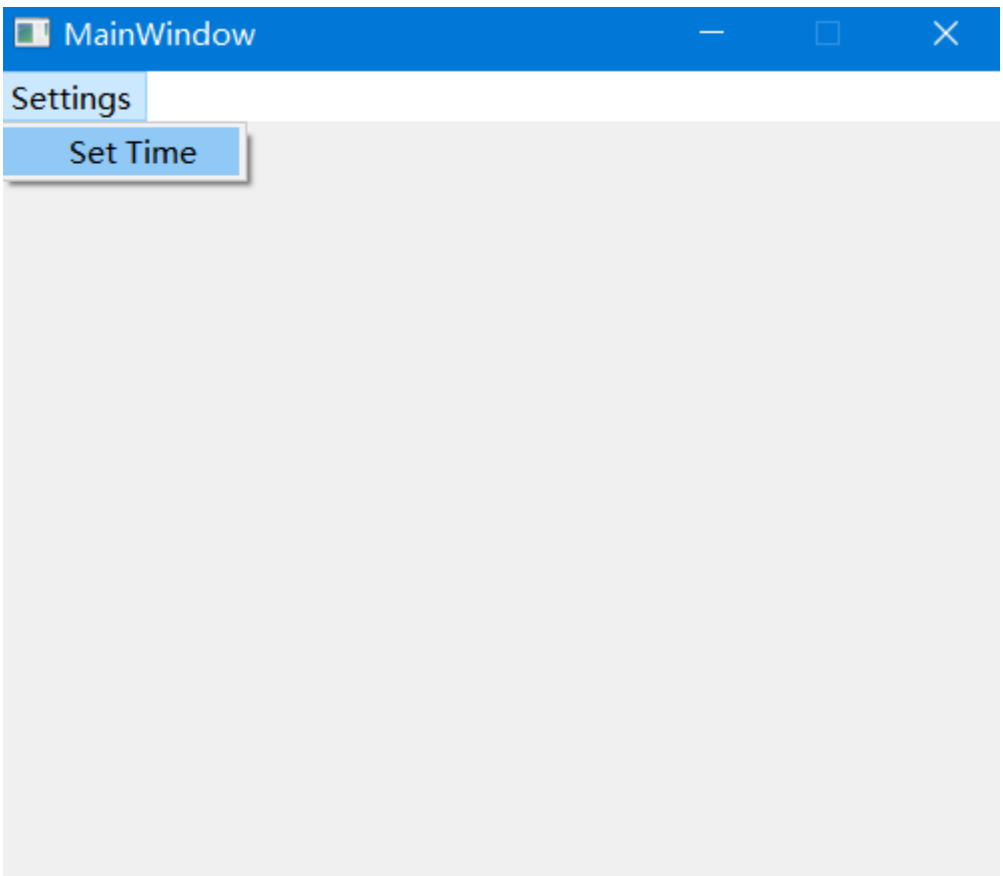
二.第二题

1.实验要求：

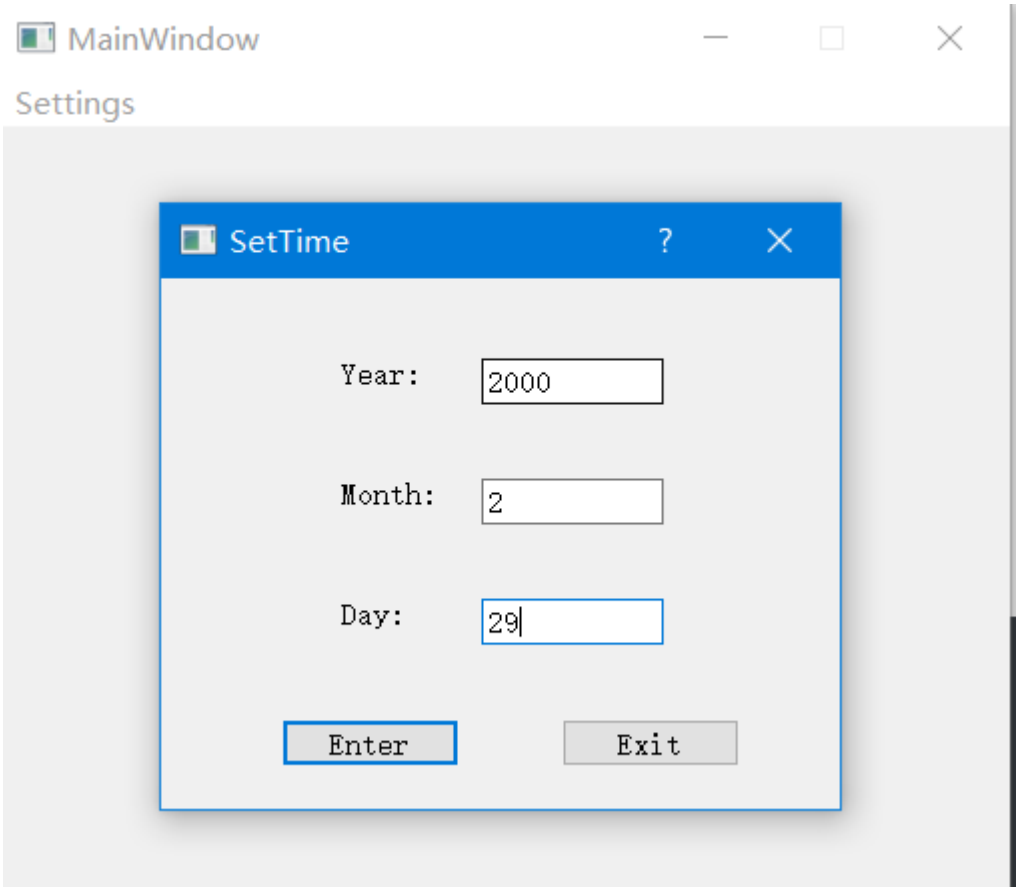
使用Qt实现界面，通过命令菜单弹出对话框并设置日期，用QMessageBox显示输入的日期，并对输入和日期合法性进行判断

2.运行示例

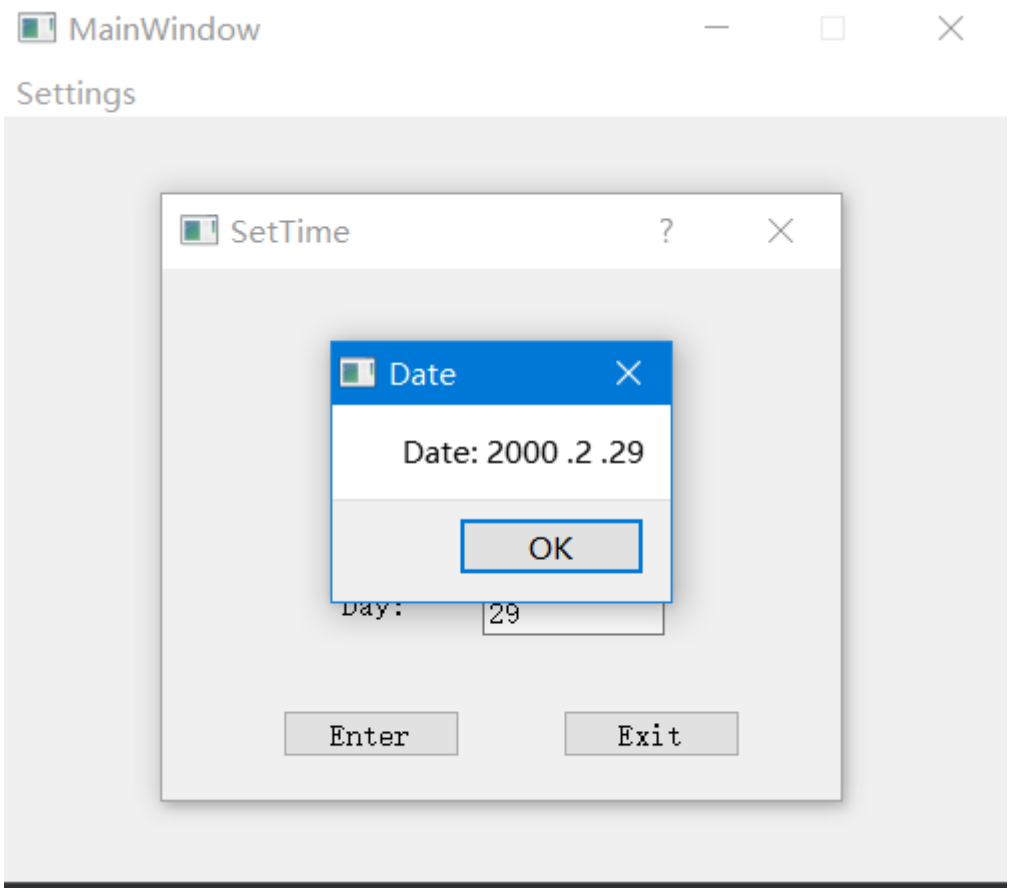
主界面：



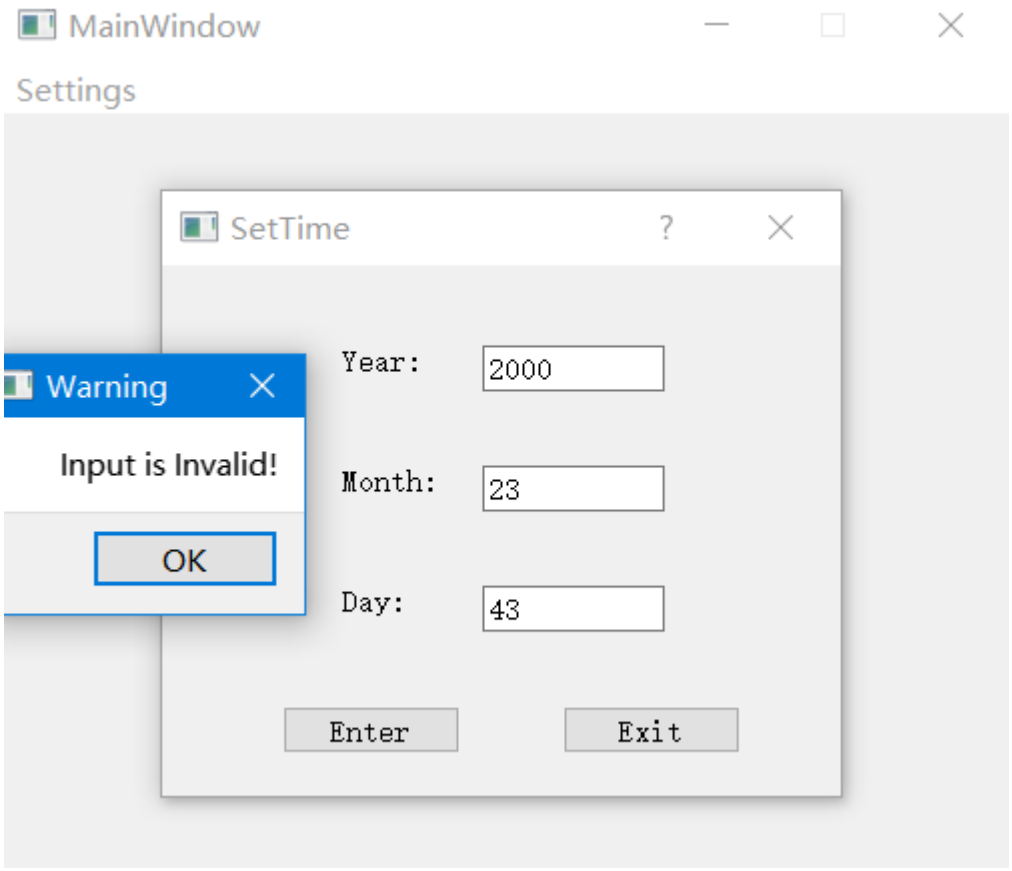
对话框设置日期：

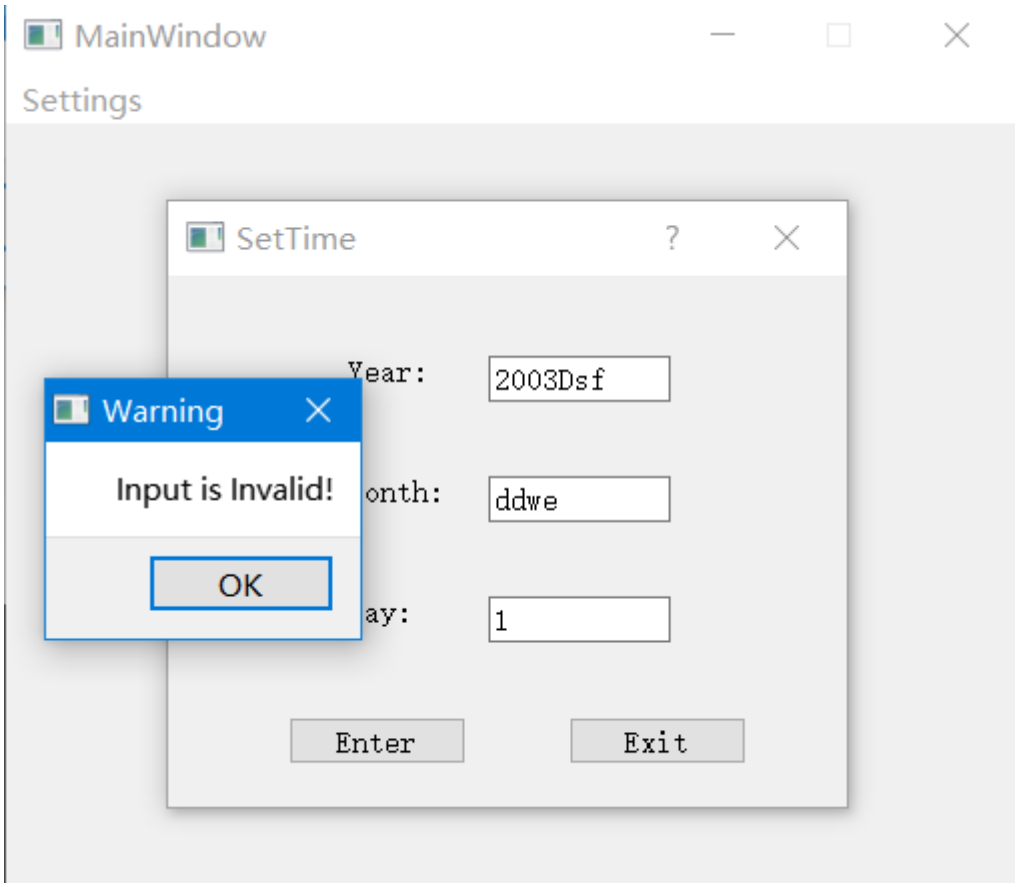

MainWindow
Settings
SetTime
Year:
2000
Month:
2
Day:
29
Enter
Exit

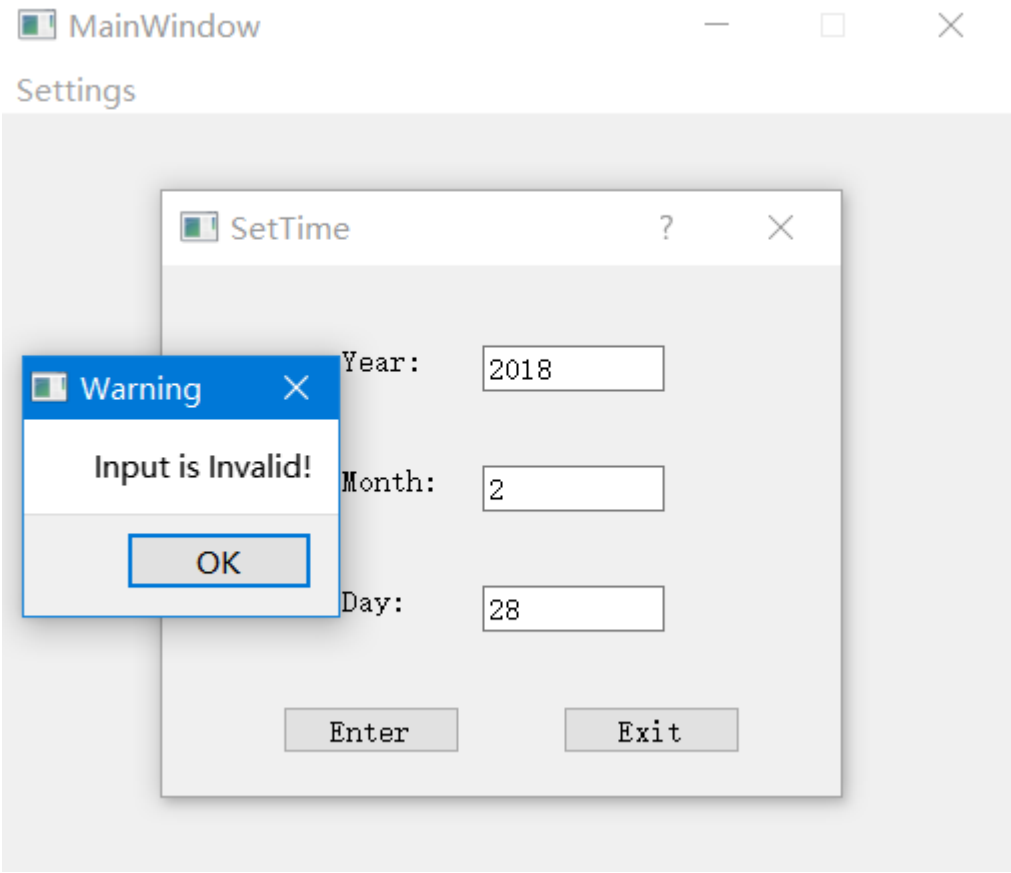
显示日期：


MainWindow
Settings
SetTime
Date
Date: 2000 .2 .29
OK
Day:
29
Enter
Exit

非法输入检测：


MainWindow
Settings
SetTime
Year:
2000
Month:
23
Day:
43
Enter
Exit
Warning
Input is Invalid!
OK


MainWindow
Settings
SetTime
Year:
2003Dsf
onth:
ddwe
ay:
1
Enter
Exit
Warning
Input is Invalid!
OK



3.代码分析

本项目有两个界面文件`mainwindow.cpp`和`dialog.cpp`，在主界面Mainwindow类中定义QMenu对象和QAaction对象，并通过信号槽的连接实现点击菜单弹出dialog界面。在Dialog类中，两个QButton对象分别实现确认和退出功能，三个文本框供使用者输入日期。其中核心功能实现由`Dialog::SetDate()`函数实现，其代码如下：

```
void Dialog::setDate()
{
    //从输入获取日期
    year=ui->lineEdit->text().toInt();
    month=ui->lineEdit_2->text().toInt();
    day=ui->lineEdit_3->text().toInt();

    //如果输入合法设置QMessageBox输出格式
    if(isReg(year,month,day))
    {
        QString str="Date: "+QString::number(year)+" ."+QString::number(month)+"
."+QString::number(day);
        char*  date;
        QByteArray ba = str.toLatin1();
        date=ba.data();
        QMessageBox msgBox(QMessageBox::NoIcon,"Date", date);
        msgBox.exec();
    }else    //如果输出不合法 弹出警告
    {
        QMessageBox msgBox(QMessageBox::NoIcon,"Warning", "Input is Invalid!");
        msgBox.exec();
```

```
    }
}
```

而对于输入合法性和日期合法性的判断由`Dialog::isReg()`函数完成，当日期符合格式但不符合现实规则也判定为不合法日期。

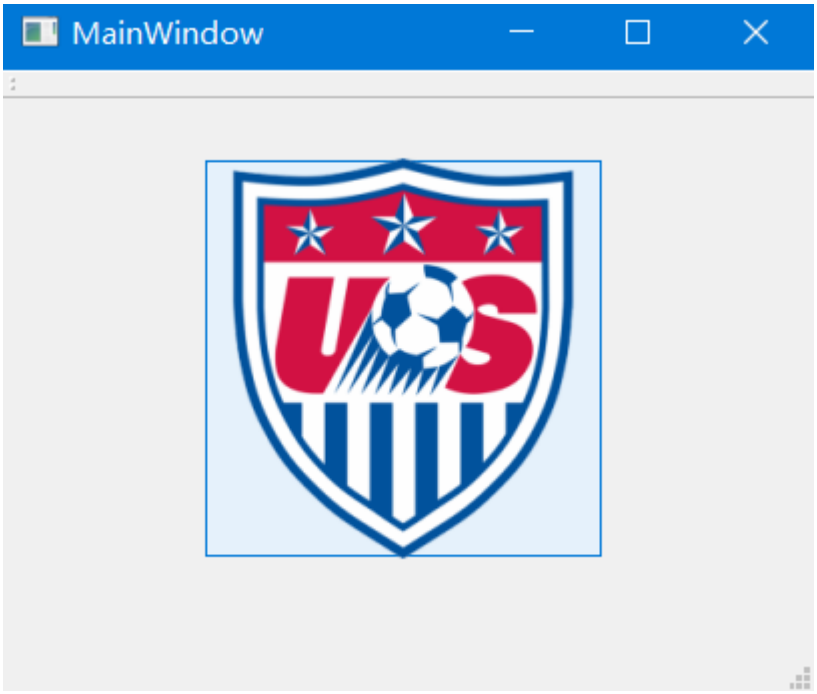
```
bool Dialog::isReg(int year, int month, int day)
{
    if(year>=0&&month>=1&&month<=12&&day>=1&&day<=31)      //判断输入日期格式是否合法
    {
        if((year%4==0&&year%100!=0)||(year%400==0))            //判断闰年
        {
            qDebug()<<"闰年"<<endl;
            if(month==2&&day>=30){return false;}              //闰年2月有29天
            else{return true;}
        }
        else
        {
            qDebug()<<"不是闰年"<<endl;
            if(month==2&&day>=28){return false;}              //非闰年2月有28天
        }

        //判断大小月
        if((month==4||month==6||month==9||month==11)&&day==31){return false;}
        return true;
    }else {return false;}
}
```

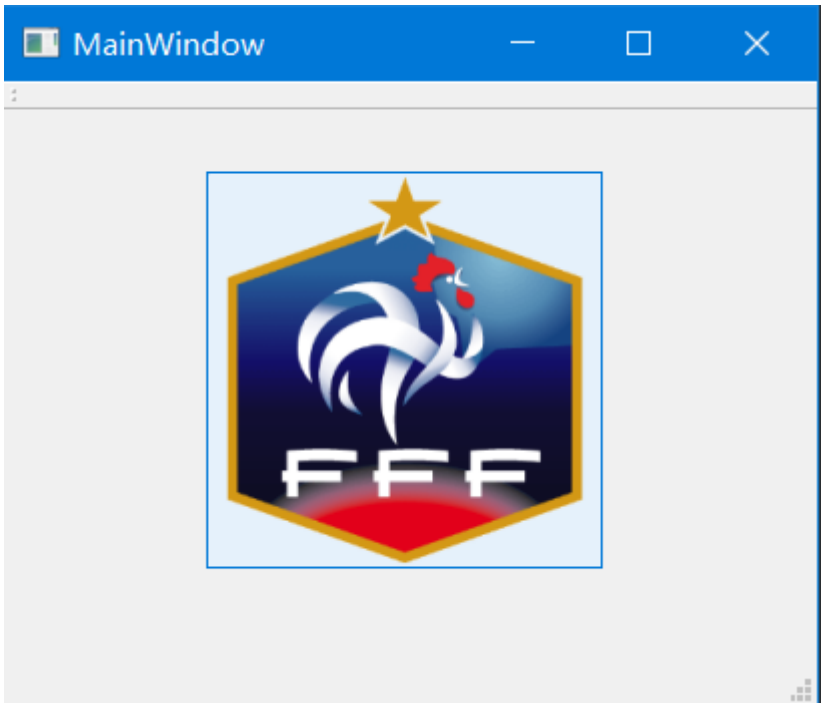
三.第三题

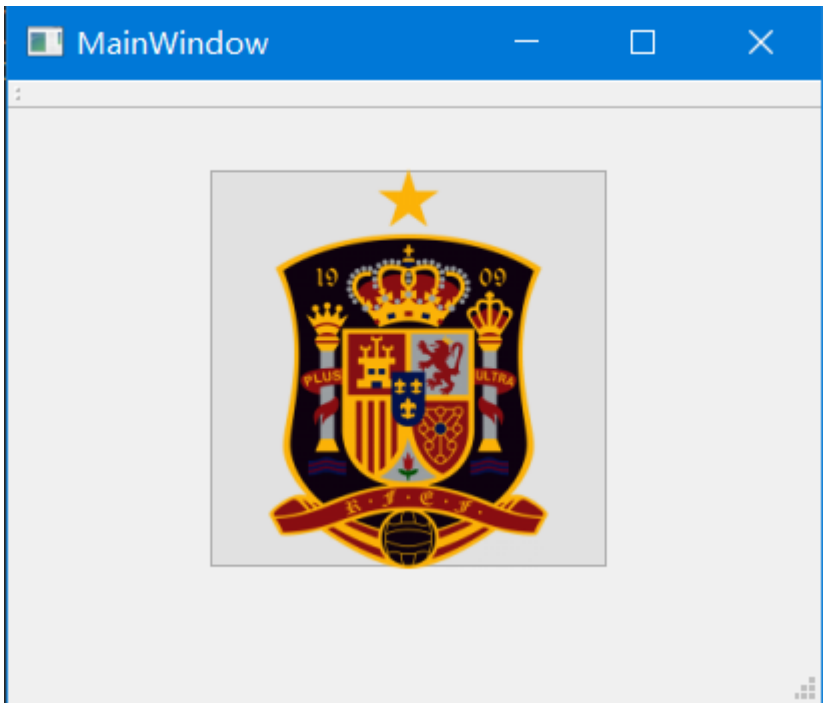
1.实验要求：

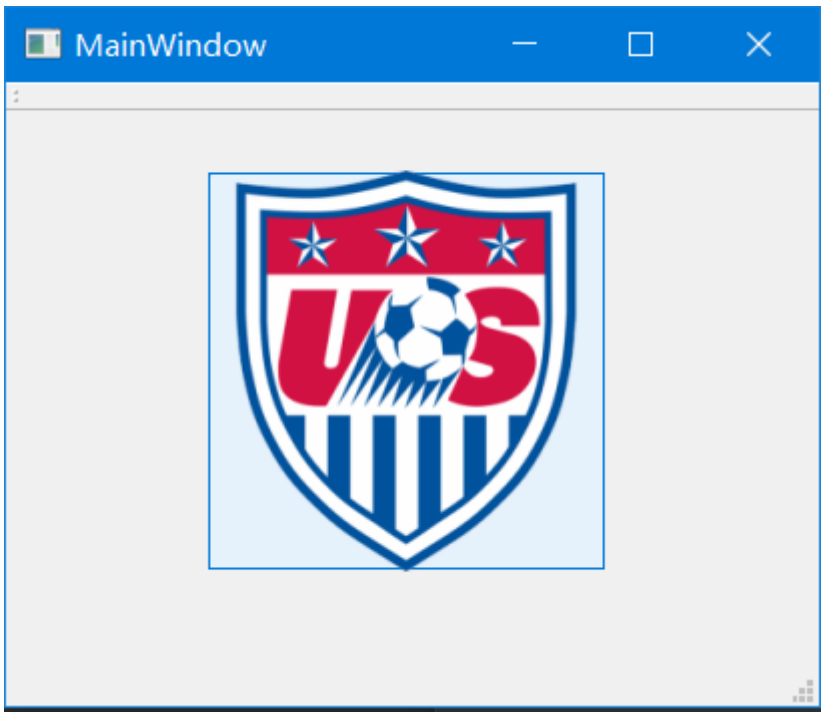
使用Qt实现界面，添加一个图片按钮并在每次按下按钮后更换图片

2.运行示例：


MainWindow


MainWindow





3.代码分析

本项目较为简单，主要实现了图片载入操作更换icon的一个槽函数。

在Qt中，使用`QIcon`类将png格式的图片初始化为对象，本次实验中我选择了2014年世界杯参赛球队的队徽12张，使用相对路径进行读取，并将它们加载在一个数列中方便用下标访问。主要的实现在`MainWindow::Icon_Change()`函数中，代码如下：

```
void MainWindow::Icon_Change()
{
```

```
    index++;
    QIcon button_icon1(":/rc/1.png");
    QIcon button_icon2(":/rc/2.png");
    QIcon button_icon3(":/rc/3.png");
    QIcon button_icon4(":/rc/4.png");
    QIcon button_icon5(":/rc/5.png");
    QIcon button_icon6(":/rc/6.png");
    QIcon button_icon7(":/rc/7.png");
    QIcon button_icon8(":/rc/8.png");
    QIcon button_icon9(":/rc/9.png");
    QIcon button_icon10(":/rc/10.png");
    QIcon button_icon11(":/rc/11.png");
    QIcon button_icon12(":/rc/12.png");
    QIcon Array[]={button_icon1,button_icon2,button_icon3,button_icon4,
                   button_icon5,button_icon6,button_icon7,button_icon8,
                   button_icon9,button_icon10,button_icon11,button_icon12};
    ui->pushButton->setIcon(Array[index%12]);          //下标12个一循环
    ui->pushButton->setIconSize(QSize(200,200));
}
```

四.一些收获

在本次的三个作业完成后我对于Qt的信号槽机制有了更深的体会和梗熟练的应用，并且通过第二个作业了解了多窗口程序的开发逻辑，通过第三个作业熟悉了图片的加载和资源管理器的使用。