实验四 嵌入式小游戏

一、 实验目的。

了解嵌入式系统下的图形化编程，在 ui 交互环境下完成小游戏开发。熟悉 qt 的基本框架，和简单控件的使用。通过游戏逻辑锻炼逻辑思维能力。

二、 实验器材。

iTop-4412 实验开发板一套、配套交叉串口线一根、PC 机一台（Windows 操作系统）、u 盘一个（FAT32），屏幕一块儿。

三、 实验原理（Qt 程序）。

1. Qt 简介。

   Qt 是一个 1991 年由 Qt Company 开发的跨平台 C++图形用户界面应用程序开发框架。它既可以开发 GUI 程序，也可用于开发非 GUI 程序，比如控制台工具和服务器。Qt 是面向对象的框架，使用特殊的代码生成扩展（称为元对象编译器(Meta Object Compiler, moc)）以及一些宏，Qt 很容易扩展，并且允许真正地组件编程。

2. Qt 常用组件介绍。

   QWidget：QWidget 类是所有用户界面对象的基类。通俗的来讲，Qt 基本上所有的 UI 类都是由 QWidget 继承出来的，而 QWidget 继承于 QObject，大家可以查阅 Qt source 即可发现一些微妙的写法，如这篇文章有详细介绍：Qt 库对象数据的声明和使用。窗口部件是用户界面的一个原子：它从窗口系统接收鼠标、键盘和其它事件，并且在屏幕上绘制自己的表现。每一个窗口部件都是矩形，并且它们按 Z 轴顺序排列的。一个窗口部件可以被它的父窗口部件或者它前面的窗口部件盖住一部分。 QDialog 是最普通的顶级窗口。不被嵌入到一个父窗口部件的窗口部件被叫做顶级窗口部件。通常情况下，顶级窗口部件是有框架和标题栏的窗口（尽管如果使用了一定的窗口部件标记，创建顶级窗口部件时也可能没有这些装饰。）在 Qt 中，QMainWindow 和不同的 QDialog 的子类是最普通的顶级窗口。一个没有父窗口部件的窗口部件一直是顶级窗口部件。非顶级窗口部件是子窗口部件。它们是它们的父窗口部件中的子窗口。你通常不能在视觉角度从它们的父窗口部件中辨别一个子窗口部件。在 Qt 中的绝大多数其它窗口部件仅仅作为子窗口部件才是有用的。（当然把一个按钮作为或者叫做顶级窗口部件也是可能的，但绝大多数人喜欢把他们的按钮放到其它按钮当中，比如 QDialog。）

QAction：Qt 使用 QAction 类作为动作,QAction 包含了图标、菜单文字、快捷键、状态栏文字、浮动帮助等信息,Qt 自己选择使用哪个属性来显示，无需我们关心。同时，Qt 能够保证把 QAction 对象添加到不同的菜单、工具栏时，显示内容是同步的。也就是说，如果我们在菜单中修改了 QAction 的图标，那么在工具栏上面这个 QAction 所对应的按钮的图标也会同步修改。

Event：QEvent 类是所有事件类的基类，事件对象包含事件参数。Qt 的主事件循环（QCoreApplication::exec()）从事件队列中获取本地窗口系统事件，将它们转化为 QEvents，然后将转换后的事件发送给 QObjects。一般来说，事件来自底层窗口系统（spontaneous() 返回 true），但也可以使用 QCoreApplication::sendEvent() 和 QCoreApplication::postEvent()（spontaneous() 返回 false）来手动发送事件。QObjects 通过调用它们的 QObject::event() 函数接收事件。该函数可以在子类中重新实现，来处理自定义的事件以及添加额外的事件类型，QWidget::event() 就是一个很著名的例子。默认情况下，像 QObject::timerEvent() 和 QWidget::mouseMoveEvent() 这样的事件可以被发送给事件处理函数。QObject::installEventFilter() 允许一个对象拦截发往另一个对象的事件。基本的 QEvent 只包含了一个事件类型参数。QEvent 的子类包含了额外的描述特定事件的参数。

信号和槽：信号和槽机制是 QT 的核心机制，要精通 QT 编程就必须对信号和槽有所了解。信号和槽是一种高级接口，应用于对象之间的通信，他是 QT 的核心特性，也是 QT 差别于其他工具包的重要地方。信号和槽是 QT 自行定义的一种通信机制，他独立于标准的 C/C++语言，因此要正确的处理信号和槽，必须借助一个称为 moc（Meta Object Compiler）的 QT 工具，该工具是个 C++预处理程式，他为高层次的事件处理自动生成所需要的附加代码。

在我们所熟知的非常多 GUI 工具包中，窗口小部件(widget)都有一个回调函数用于响应他们能触发的每个动作，这个回调函数通常是个指向某个函数的指针。不过，在 QT 中信号和槽取代了这些凌乱的函数指针，使得我们编写这些通信程式更为简洁明了。信号和槽能携带任意数量和任意类型的参数，他们是类型完全安全的，不会像回调函数那样产生 core dumps。

所有从 QObject 或其子类(例如 Qwidget)派生的类都能够包含信号和槽。当对象改动其状态时，信号就由该对象发射(emit)出去，这就是对象所要做的全部事情，他不知道另一端是谁在接收这个信号。这就是真正的信息封装，他确保对象被当作一个真正的软件组件来使用。槽用于接收信号，但他们是普通的对象成员函数。一个槽并不知道是否有所有信号和自己相连接。而且，对象并不了解具体的通信机制。

你能将非常多信号和单个的槽进行连接，也能将单个的信号和非常多的槽进行连接，甚至于将一个信号和另外一个信号相连接也是可能的，这时无论第一个信号什么时候发射系统都将即时发射第二个信号。总之，信号和槽构造了一个强大的部件编程机制。

声明信号：

```
void mySignal();
```

声明槽：

```
public slots:  void mySlot();
```

绑定信号和槽：

```
QObject::connect( scroll, SIGNAL(valueChanged(int)),
label, SLOT(setNum(int)) );
```

四、 实验内容。

实例代码中对于按钮点击的控制部分已被删掉，同学们需要完成推箱子游戏的逻辑控制部分。下面是小游戏的简单介绍。

1. 工程组成介绍。

   Mainwindow.c, Mainwindow.h: 工程的主界面，主逻辑类。学生需要完成的任务是补充该类中缺失的部分。

   Mainpaint.c,mainpaint.h：游戏界面的绘制类，为了简化实验内容，该部分是已经封装好的，只需要在主类中调用即可，不许修改。

2. 主要变量，函数介绍。

   Mainwindow:

   Q 开头的宏定义：对应的是墙体。

   Box：箱子

   Goal：目标点。

   Get：箱子到达目标点。

   Ground:地面

   MAN：人物。

   MapH：地图宽度 MapV：地图高度 int map[4][MAPH][MAPV]; 初始地图，0 维表示关卡树，1 二维表示地图坐标。

   Int man[4][2]: 人物的初始坐标。

   Int box[4]；每关箱子的个数。

   Int boxNum:记录当前已经完成的箱子个数，用于校验是否完成关卡。

   StartGame（）；开始游戏并将初始地图元素付给绘制类。

   Keyup();

   Keydown();

   Keyleft();

   Keyright();

   需要在以上四个函数中补充确实代码完成游戏。

Mainpaint:

Int stepNow:当前关卡已经走过的步数。

Int levelNow: 当前的关卡。 Int

Man_h,man_v:当前任务的横纵坐标。

int mapNow[MAPH][MAPV]:记录当前的地图元素

3. 控制逻辑。

以点击向上按钮为例:

int manx=mainmap->man_h;

int many=mainmap->man_v;

分别为人物的当前横纵坐标。

需判断:

1.如果人物上方是箱子，或者人物上方是已经得分的箱子。

if((mainmap->mapNow[manx][many+1]==BOX)||(mainmap->mapNow[manx][many+1]==GET))。

2.如果人物上方是得分点或者是地面。

if((mainmap->mapNow[manx][many+1]==GOAL)||(mainmap->mapNow[manx][many+1]==GROUND))

4. 界面弹窗

采用 QMessageBox 实现。运行编译测试的程序，点击游戏界面内的“开始游戏”，会弹出一个对话框，供大家使用参考。使用时请引用 QMessageBox 头文件，并依葫芦画瓢完成弹窗显示游戏失败的功能。

5. 游戏失败逻辑。

最简单想到的是，当箱子推入非得分点死胡同，游戏失败。死胡同可以理解为是箱子的相邻两边都为墙，这种情况下，玩家是不可能再将这个箱子推开的，因此游戏失败。

五、 实验步骤。

1. 在编译测试目录中敲入 make 编译，查看是否编译生成了 tuixiangzi 二进制文件。将此文件和 game 文件夹放到同一个目录下，在板子上运行，
2. 根据注释和代码阅读把空缺的逻辑控制代码补全，并在服务器上编译产生可执行文件。
3. 把可执行文件放至 u 盘挂载至嵌入式设备，并将 game 文件夹复制到可执行文件同目录下。通过超级终端运行，并通过第一关验证逻辑正确。若运行失败，重复第一步。

4. 加入游戏失败的判断。当箱子推入死胡同时，弹出消息框显示“Game over”。
5. 提高部分：加入游戏关卡记忆功能。本游戏共有 3 关，例如，通关第一关后，退出游戏后再次进入，直接进入第二关。当第三关通关后再进入，直接进入第 1 关。