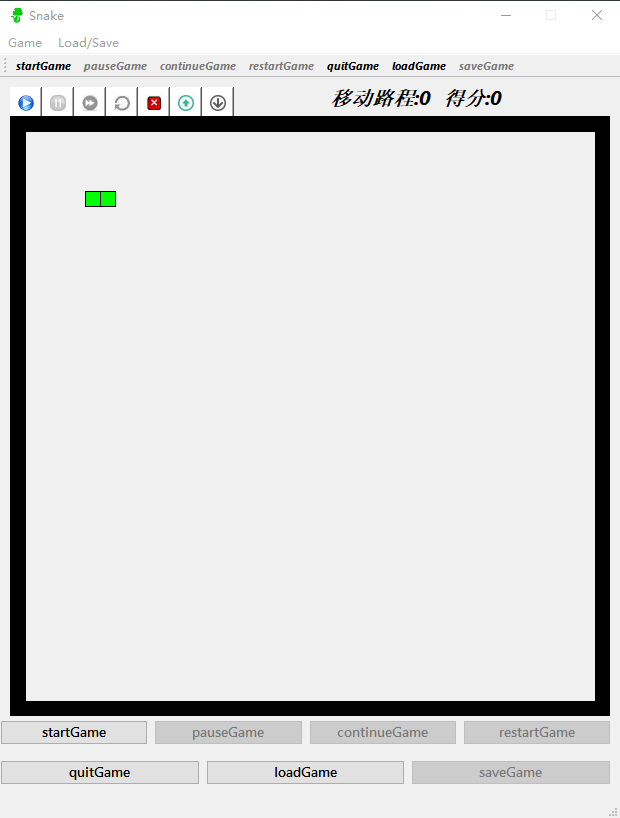
## 小学期第一周大作业报告

### 一、UI设计

首先用QtDesigner设计出交互界面，程序主窗口MainWindow是继承自Ui的子类，menuBar，toolBar，以及下方的七个QPushButton均来自Ui类。在MainWindow类中另外定义了一个QLabel对象用于显示游戏路程与游戏得分。MainWindow类还有一个成员gameWidget，是paintWidget类的实例化。其中paintWidget是继承自QWidget的子类。该类重写paintEvent函数用来画出地图上的边界，蛇身，果实等元素。游戏的主要逻辑都是在paintWidget这个类中实现（范围是图中黑色框部分）。

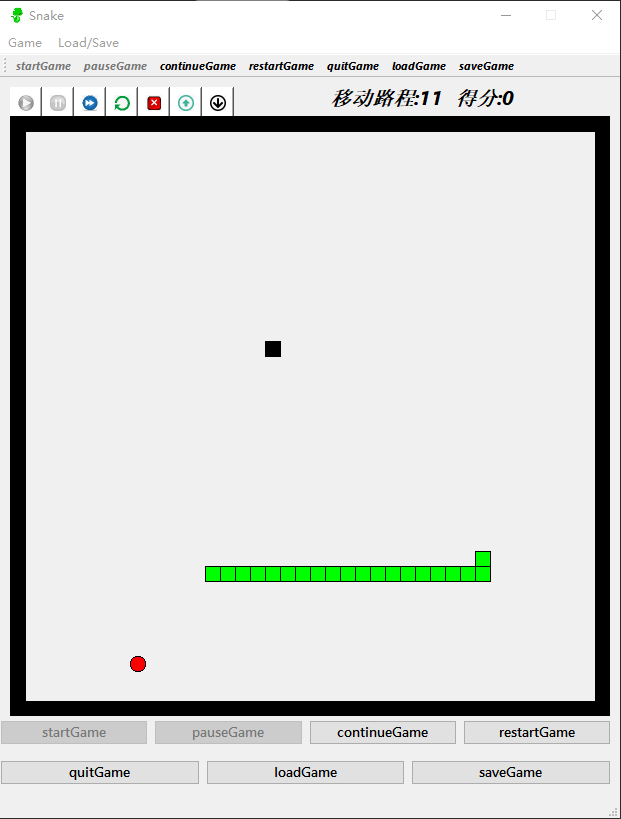


### 二、游戏逻辑

将地图上每一个点看作一个Node结构体，结构体中对应记录了每一个节点的横、纵坐标，以及节点类型(0=empty, 1=border, 2=snake, 3=food)。gameWidget包含一个40\*40的节点二维数组记录所有节点信息。地图上所有的障碍物，蛇，果实信息都记录在了三个QList(border, snake, food)中，通过QList的remove, append函数结合update()调用的paintEvent()函数实现地图外观的动态变化。

gameWidget通过mousePressEvent（左键添加障碍物，右键取消障碍物），keyPressEvent（上下左右键改变dx,dy，空格键暂停游戏）来响应鼠标与键盘的操作。

初始化前，可以通过鼠标点击将空白节点设置为障碍物，蛇位于地图的左上角位置。游戏开始后，首先会随机生成一个食物。通过QTime类的timeout()信号触发gameWidget的moveSnakeSlot()槽：进入槽函数时，由于一个果实对应蛇长度伸长三，首先通过判断是否还没有完全伸长（通过int型成员bonus记录）判断尾巴是否需要“消失”。之后结合dx，dy判断蛇头下一个位置对应的节点类型：border与snake对应游戏结束，empty对应bonus不变，food对应bonus加3。更新地图信息与地图外观之后进入下一个moveSnake槽函数，从而实现了地图的动态变化。



### 三、信号与槽分析

MainWindow类组件中，七个menuBar, 七个toolBar, 七个QPushButton本质上是一样的，分别触发gameWidget的startGameSlot, pauseGameSlot, continueGameSlot, restartGameSlot, saveGameSlot, loadGameSlot, quitGameSlot。MainWindow的槽主要是用于更改按钮的Enable状态，比较简单。主要分析gameWidget的七个槽：

startGameSlot: 创建第一个食物，同时QTime对象timer->start(); pauseGameSlot: 游戏暂停，timer->stop(); restartGameSlot: 初始化自身状态，同时timer->stop(); continueGameSlot: 游戏继续，timer->start(); quitGameSlot: 调用系统函数，关闭窗口。

saveGameSlot, 通过QT自带的json文件相关库，将游戏状态（dx, dy, roadLen, score, 节点信息map, bonus等）写入一个json文件并通过QFileDialog::getSaveFileName调用存储文件对话框选择存储文件位置与命名文件。loadGameSlot, 是saveGameSlot的逆过程，解析json文件并赋值，使得能恢复之前的游戏状态。