|  |  |  |  |
| --- | --- | --- | --- |
| **1** |  |  |  |
|  |  | **版本：v1.00** |  |

贪吃蛇小游戏

2018年 3月

软件设计说明

网易（互联网）C++开发工程师大作业

文档修改记录

|  |  |  |
| --- | --- | --- |
| 序号 | 版本 | 修改内容 |
|  | A | 建立 |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

# 1 范围

## 1.1 标识

本文档的标题为贪吃蛇游戏软件设计说明。

本文档适用于贪吃蛇小游戏。

## 1.2 概述

本软件主要针对网易互联网公司的新人培训计划中C++开发工程师部分的通关大作业而进行的开发。本人选择的是第一个UI类任务，开发一个贪吃蛇小游戏。本贪吃蛇游戏包含两种玩法一种是传统的贪吃蛇模式，为积分制。另外一种玩法是为了增加游戏的趣味性，单人贪吃蛇的基础上加入了双人对抗模式。

本游戏的开发人员是金强

本软件为新开发软件，当前版本为1.0。

## 1.3 文档概述

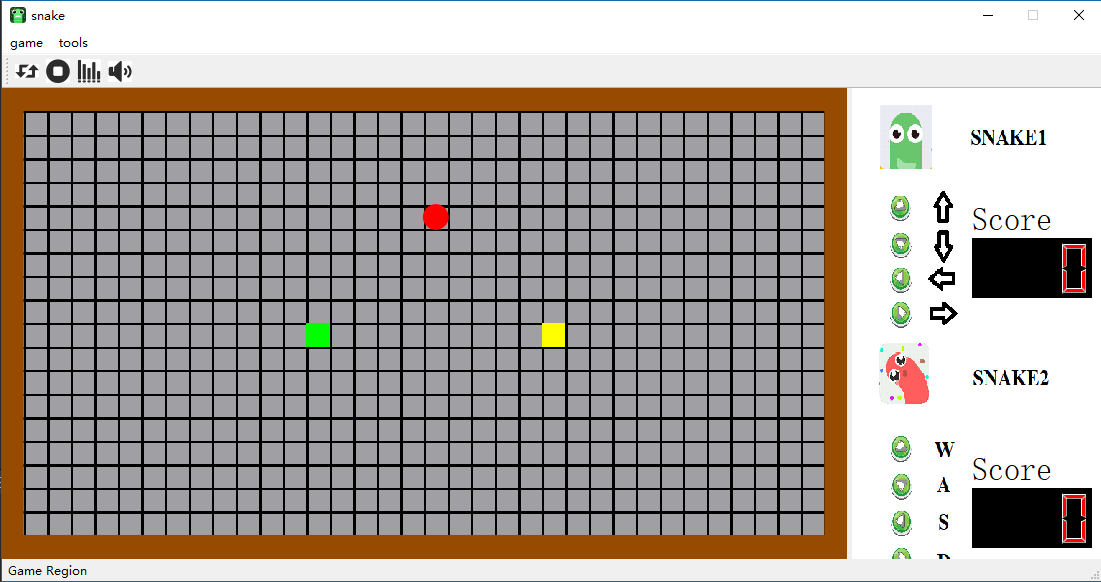
本文档适用于描述贪吃蛇小游戏的设计说明文档。

在本文档中，第2章列出了本游戏的界面设计，第3章描述了本游戏的详细设计策略。

# 2 界面设计

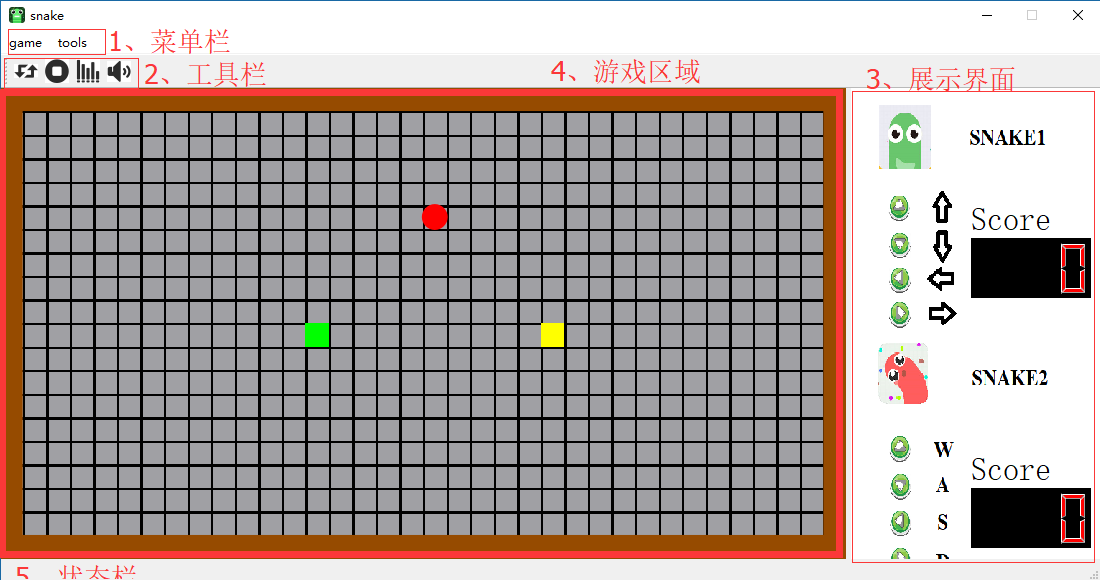
## 2.1 整体设计

本游戏整体包含两个界面，开始界面和游戏界面，如下所示，其中开始界面仅由两个按钮组成，游戏界面是一个经典的应用程序界面。

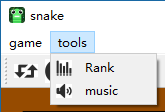
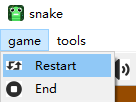
有些界面由五个部分组成，具体位置如下图所示：

1. 菜单栏，相关菜单按钮；
2. 工具栏，相关工具的快捷按钮；；
3. 展示界面，展示信息
4. 游戏区域；
5. 状态栏，显示当前软件状态。



## 2.1 菜单栏

菜单栏包括两组菜单，分别是“game”和“tools”：



“game”中有“Restart”选项，选择将重新开始游戏，“End”选项，选中后直接结束游戏；

“tools”中有“Rank”选项，选中后将显示游戏的排名，“music”选项，选中后开始/暂停游戏背景音乐。

## 2.2 工具栏

工具栏按钮为菜单栏选项的快捷显示，作用同菜单栏。

## 2.3展示界面

展示界面展示了玩家1和玩家2的操作方式，还有两个窗口显示对应玩家的得分。

## 2.4 游戏区域

游戏区域是游戏的主体界面，里面包含了咖啡色的墙体，红色的食物，绿色和黄色的小蛇。

## 2.5 状态栏

状态栏用于显示当前鼠标所处界面中不同区域的作用，用以辅助玩家操作

# 3 详细设计策略

## 3.1 开始界面类StartWidget

StartWidget类继承自QT的QWidget类，成员变量添加了两个按钮，用于选择游戏模式，一个主界面指针用于创建主界面；成员函数由三个槽函数组成，用于响应两个按钮的操作和重绘开始界面。



## 3.2 游戏主界面类MainWindow

主界面类继承与QMainWindow,并进行自定义设计。菜单栏和工具栏通过添加QAction添加了四个按钮，分别是restartGameAction、endGameAction、rankingAction、musicAction。主界面的主窗口使用QSplitter进行布局，分割成两部分，一个部分添加说明面板-自定义的类ControlPanel，另一个部分使用Q自带的Graphics View Framework 作为核心游戏框架，添加了一个QGraphicsScene，作为游戏发生的舞台；一个QGraphicsView，作为观察游戏舞台的组件；以及若干元素，用于表示游戏对象，比如蛇、食物以及墙体。为了增加趣味性，添加了一个QMediaPlayer类，播放背景音乐。该类的成员函数部分包括了一系列的窗口初始化、音乐初始化函数，以及一些在控制窗口绘制和游戏开始结束方面的信号函数和槽函数。具体的类设计如下图：



## 3.3 游戏排名对话框类RankDialog

游戏排名对话框类继承自QDialog，并进行布局设计。每条记录包括了名次、姓名、和分数。该类的成员变量包括了一个QTextEdit用于在界面上显示记录，一个QVector<item>用于保存成绩。成员函数部分包含了一个void writeTextWidget()，用于将成绩写往面板，一个void writeText()用于将成绩写往硬盘，一个int update(QString name,int score)用于更新成绩。



## 3.4 游戏说明面板类ControlPanel

ControlPanel类继承自QT的控件类QWidget，进行自定义的布局设计。该类的成员变量包括了两个用于显示分数的LCD面板player1LCD、player2LCD，和两个对这个面板进行标注的标签player1Score、player2Score。成员函数包括了用于初始化界面的initLCD()和initLabel()；重写了绘图事件void paintEvent(QPaintEvent \*event)用于将游戏的操作说明以背景的形式画入；两个两个用于设置LCD面板值的函数void setPlayer1LCD(int score)、void setPlayer2LCD(int score)。



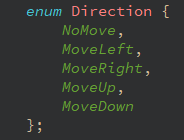
## 3.5 游戏的控件类Food食物类、Wall墙体类和Snake蛇类

本游戏的地图是建立在QGraphicsScene的基础之上的，所以，里面的对象应该是QGraphicsItem实例。我们会把所有的图形元素包括蛇、墙体和食物设计为QGraphicsItem的子类，在这个类中添加绘制自身的代码以及动画逻辑。

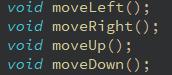
首先是较为固定的墙体类和食物类，食物我们使用圆形控件表示，墙体类使用长方形空间表示。由于Graphics View Framework框架中已经实现了继承自QGraphicsItem的QGraphicsRectItem和QGraphicsEllipseItem用于画长方形和圆形控件，同时为了方便后期的扩展和维护，我们使用装饰者模式，在现有的长方形和圆形控件的基础上实现食物类和墙体类。具体的类图如下图。实现一个装饰类GraphicsItemDecorator，通过调用其持有的对象实现继承来的一些虚函数；一个Wall类继承自GraphicsItemDecorator，添加一个墙体设置函数setColor(QColor color)，设置一些墙体的属性，并在构造函数中，添加其墙体的标记，该类通过添加一个长方形控件实现墙体；一个Food类继承GraphicsItemDecorator，添加一个墙体设置函数setColorPos(QColor color,qreal x, qreal y)，设置一些食物的属性，并在构造函数中，添加其食物的标记，该类通过添加一个圆形控件实现墙体；



游戏中的小蛇同样是继承自QGraphicsItem，小蛇由一个方块表示头部，一个list用于保存身体部分。蛇的移动方向通过一个枚举类型构成



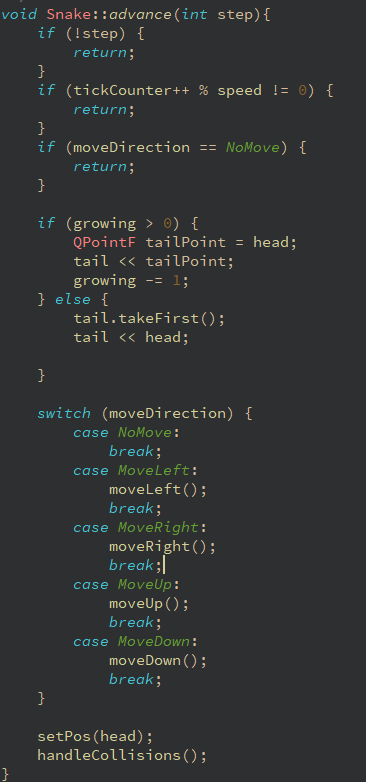
控制器为小蛇设置移动方向后小蛇在每个时钟周期里检测小蛇的移动方向调用相应的函数进行移动



在每个时钟周期结束后都要检测小蛇是否与场景中的物体发生了碰撞，碰撞检测函数为void handleCollisions()，在这个函数中通过调用基类的collidingItems()函数检测与所有与小蛇有交集的item，通过判断item的类型，调用相应的控制器中的函数，可能与小蛇发生碰撞的item类型有食物-吃掉食物、墙体-撞墙了、小蛇自己、其他蛇。其中与其他蛇相撞还要判断碰撞的位置，若是两条蛇头碰头，则看谁的分数高赢，若是自己碰别人的身体，则是自己失败。



每个时钟周期场景类都会调用重写的snake类的advance()函数实现蛇的移动：



其中tickCounter实际是内部的一个计时器。我们使用 speed 作为蛇的两次动作的间隔时间，两次运动的间隔时间越大，游戏越简单。这是因为随着 speed 的增大，tickCounter % speed != 0 的次数响应越多，刷新的次数就会越少，蛇运动得越慢。moveDirection显然就是运动方向，当是 NoMove 时，函数直接返回。growing是正在增长的方格数。当其大于 0 时，将头部追加到尾部的位置，同时减少一个方格；当其小于 0 时，删除第一个，然后把头部添加进去。

## 3.6 控制器类Controller

Controller用来初始化场景中的游戏对象，设置游戏时钟，通过将时钟结束的信号发给场景的advance()槽函数，开始游戏循环。





Controller类的成员包括一个时钟发生器，一个场景类，指向场景中小蛇的指针。成员函数包括了一个事件过滤器以便监听键盘事件，一系列控制蛇碰撞的函数以及一些槽函数和控制函数。