

北 京 林 业 大 学

2020 学年— 2021 学年第 2 学期 C++程序设计

大作业报告

专 业： 计算机类 班 级： 计算机类 20-1 班

姓 名： 王溪云 学 号： 201002105

任课教师： 徐艳艳

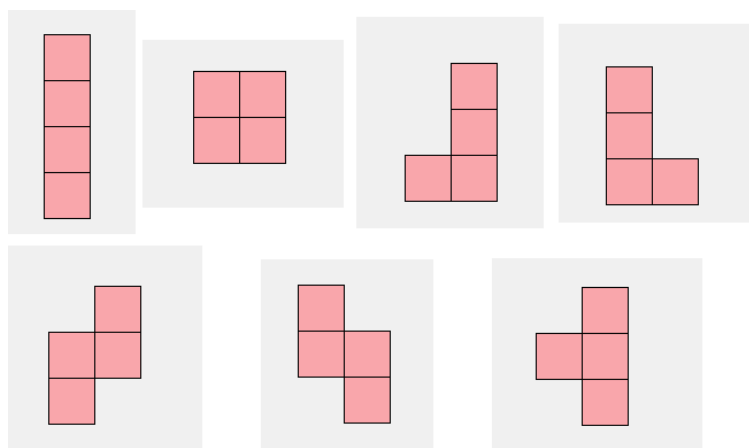
题目： 俄罗斯方块

实验环境： Qt Creator

内容：

1. 玩家在一个 10*20 范围内的空间进行游戏，所有图形不越界；
2. 包含 5 种图形，并可以向四个方向旋转：
 - A. “一”字形
 - B. 正方形
 - C. L 型（左，右）
 - D. “之”字形（左，右）
 - E. “凸”字形

其中 L 型和“之”字形包含左向和右向两种类型。如下图所示：



3. 七种图形随机掉落，并在游戏区域内持续下落直到停止；同时在游戏区域右侧对下一个图形进行提示，并显示当前分数和关卡；
4. 玩家使用键盘控制游戏进行（上、下、左、右、空格）：
 - A. “左”、“右”分别控制图形的左右移动；
 - B. “下”控制图形加速下落；
 - C. “上”控制图形旋转 90 度；
 - D. 当图形碰撞到底部边界或已经存在的图形导致无法继续下落时，下落停止，玩家无法继续操控此图形；
 - E. “空格”使游戏暂停；
5. 当一行内的所有方格都被填满时，该行自动消除，上方行随之下落；
6. 当方块堆叠到游戏区域最顶端时，游戏结束；
7. 玩家每次消除一行方块得 1 分，不设上限，其中同时消除 n 行得到 $n \times n$ 分；
8. 分数每增加 10 分，图形下落速度加快 200ms；当图片下落速度增加至 200ms^{-1} 时，下落速度不再加快；
9. 每个图形可以看到即将落到的最下方位置。

过程：

算法实现思路：

首先对主窗口进行构造，大致将其分为游戏区和信息区两类，游戏区响应用户操作，信息区响应游戏区产生的信息变化。

打开窗口后首先对游戏界面产生初始化，绘制背景边框和三类方块，然后主窗口对键盘事件进行响应并通过 `ui->GameArea->keyPressed(e->key())` 传递到主窗口中使当前下落方块移动；若玩家未使用键盘对方块行为进行操纵，获取当前关卡，使用定时器发出信号使方块进行匀速移动。

移动过程中，应当判断方块是否可以继续移动、当前位置是否允许变化方向等。

方块即将下落位置的显示应当随正在下落方块位置变化。正在下落的方块的位置每次改变，重新设置“显示”的坐标。

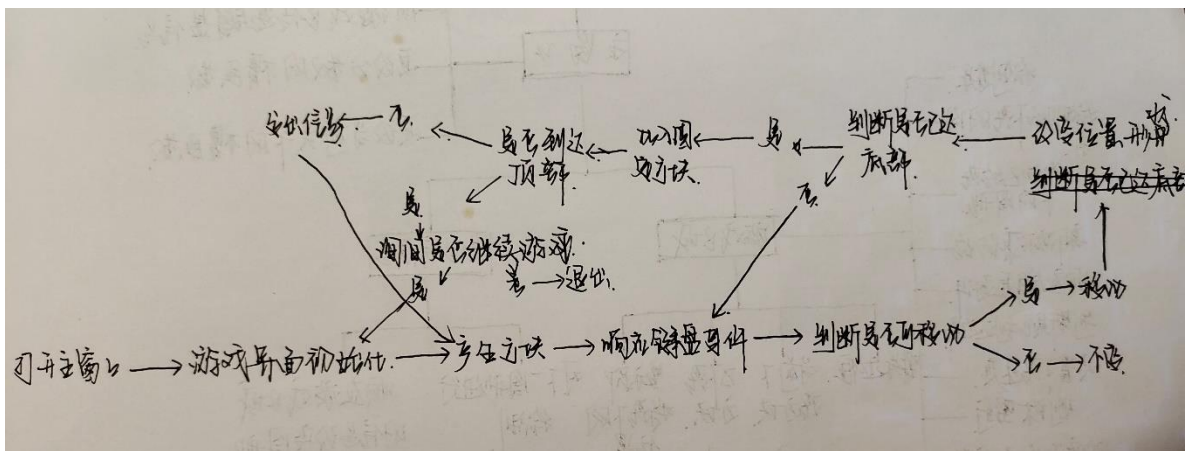
方块持续下落，每次下落判断是否与已固定的方块重合或者根据坐标判断是否已经到达底部，将当前方块加入到已落下的方块中（坐标）。

更新当前下落的方块、下一个将要下落的方块、已固定的方块，判断已落下的方块中是否有一整行已被填充满，若有，消除该行并使该行以上的方块整体下落，并加上相应的分数。最后对游戏区域进行重绘。

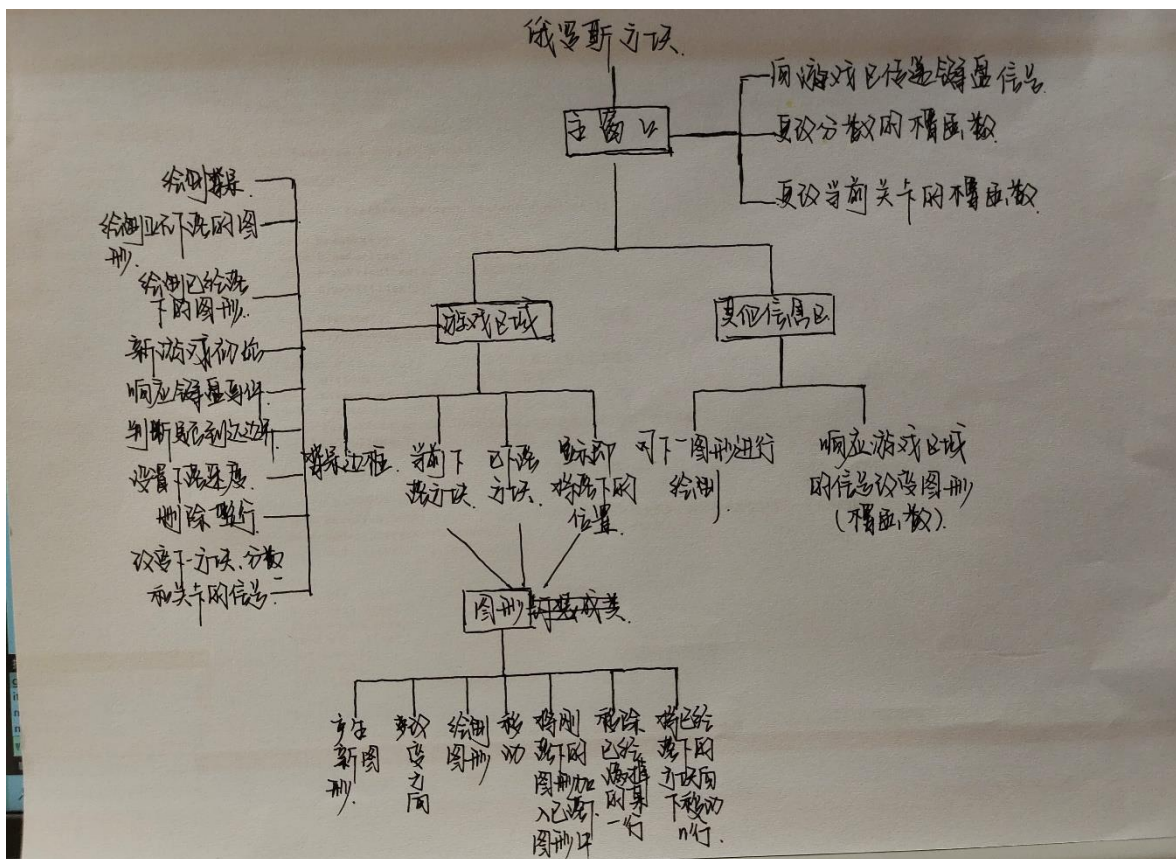
同时判断落下的方块是否到达顶部，若是，弹出提示框，重新游戏或退出游戏；若无，发出更新分数、更新下一方块的信号；若分数达到关卡变化的分数，发出信号更新关卡。

产生下一方块循环执行上述过程。

同时实现游戏中暂停，按下空格键后由主窗口传递至 GameArea，停止定时器。



程序模块结构：



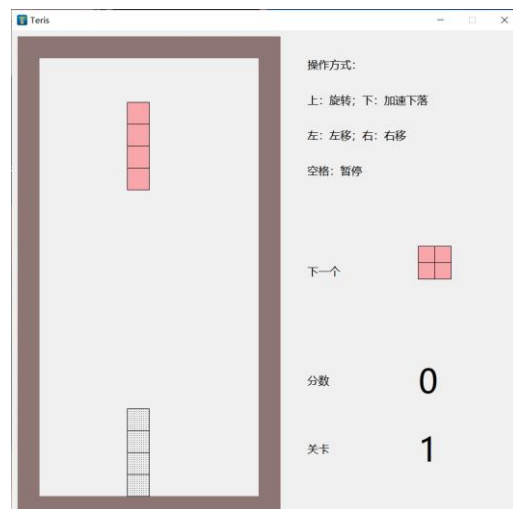
使用 Qt 框架进行图形界面设计，并由主窗口产生游戏区域 GameArea 和信息区域两个无边框子窗口，其他信息区构造 NextArea 子类用于生成下一个将要落下的方块。

子类 GameArea 中主要用于响应玩家操作，控制方块的下落及随之产生的位置变化，对游戏区域进行初始化，设置方块下落的速度，删除已经全部填充完的整行方块，并向其他区域发出信号。

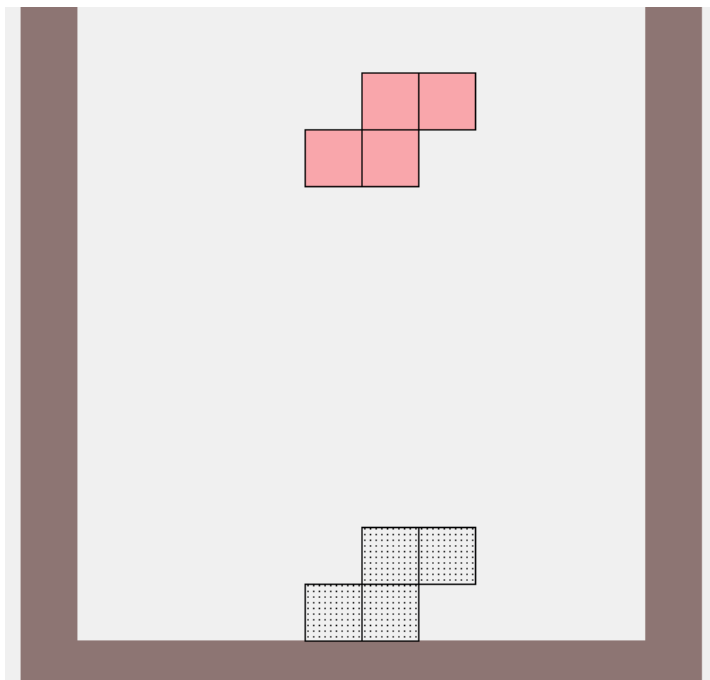
将方块封装为类，在 Item 类中定义方块的形状和位置，具体实现图形方块的方向变换，当前落下方块与已落下方块的合并，已落下方块整行的删除和移动等。

结果：

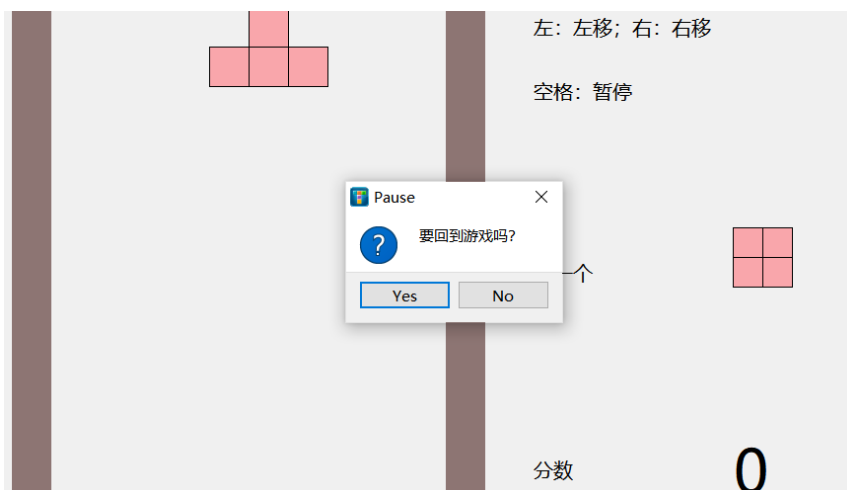
1. 当前图形下落时会对下一块图形进行提示。



2. 显示图形即将落到的最下方位置。



3. 按空格键可以对游戏进行暂停



分析（必须包括以下 3 点）：

1、大作业完成过程中遇到的问题及解决方法和心得体会

①此前未接触过图形界面设计，在终端命令行实现后产生了深深的不满足，但对于图形界面设计无从下手。询问有经验的学长学姐，鼓励我趁此机会学习 Qt、MFC、OpenGL 等跨平台的 GUI 框架，查阅资料进行比较后选择了 Qt 对俄罗斯方块进行实现。

②对具体算法思路略混乱。网上寻找了一些实现比较好的代码进行阅读和梳理。

③接触新的领域（Qt 框架），利用网络优势在 B 站，慕课以及其他相关网站查询资料，询问有经验的前辈，初步踏入 C++ 图形界面开发的大门。

④遇到不懂和困难要多询问，首先要自己试着在网上查询解决再询问其他人，培养自己独立解决问题的能力。

⑤工程项目调试相对困难。向有经验的前辈询问经验，逐渐熟练对代码较多的工程的 Debug。

2、查阅资料

为了理清代码思路，找到一些已经写好的代码进行阅读：

界面设计主要参照 <https://zhuanlan.zhihu.com/p/384838972>，并参照了部分代码思路。

<https://github.com/shiyicode/project/tree/master/eluosi> 阅读代码理清思路。

前期对 Qt 和 MFC 的优劣进行了查询对比：

<https://blog.csdn.net/liubing8609/article/details/86526406>

<https://www.cnblogs.com/tsingke/p/10104276.html>

并对 OpenGL 进行了一些阅读（虽然没有使用）：

英文版教程：<https://learnopengl.com/>

中文版教程：<https://learnopengl-cn.github.io/>

查阅的主要英文资料使 Qt 文档以及 Github 上对于代码的英文描述（过于杂乱略过不举），Qt 文档通过 QtCreator 直接查询。

查询 QWidget、QDialog、QMainWindow 三种类型窗口的特点：

<https://doc.qt.io/qt-5/qwidget.html>

<https://doc.qt.io/qt-5/qdialog.html>

<https://doc.qt.io/qt-5/qmainwindow.html>

了解 Qt 中的数据类型

<https://doc.qt.io/qt-5/qmainwindow.html>

比较 QByteArray 和 QString:

<https://doc.qt.io/qt-5/qbytearray.html>

<https://doc.qt.io/qt-5/qstring.html>

了解 Qt 中的 QVariant 类的 API, 需恶习使用 QVariant 对 Qt 中标准数据类型和自定义数据类型进行处理:

<https://doc.qt.io/qt-5/qvariant.html>

学习使用 Qt 中的坐标类, 直线类, 尺寸类, 矩形类, 日期时间类等:

<https://doc.qt.io/qt-5/qpoint.html>

<https://doc.qt.io/qt-5/qline.html>

<https://doc.qt.io/qt-5/qrect.html>

.....

信号和槽的使用:

<http://doc.qt.io/qt-5/signalsandslots.html>

QTimer 类定时器的使用:

<https://doc.qt.io/qt-5/qtimer.html>

QDialog 类中对话框:

<http://doc.qt.io/qt-5/qmessagebox.html> 消息对话框

<https://doc.qt.io/qt-5/qfiledialog.html> 文件对话框

还有字体对话框、颜色对话框、输入对话框等。

窗口布局 Layout:

<https://doc.qt.io/qt-5/qml-qtquick-layouts-layout.html>

绘图事件:

<https://doc.qt.io/qt-5/qpainter.html>

<https://doc.qt.io/qt-5/qpaintevent.html>

.....

3、对软件维护的认识

软件的可视性不强，如何维护以及维护的工作量都难以确定，维护的效果不易看清。这些问题造成维护工作存在很多困难。

软件维护的目的大致可以分为三个方面：

- 1, 对各种 bug 的修复, 优化一些体验问题, 提升版本稳定性和使用体验, 可能修复了一些问题又会有新的 bug 出现, 所以就需要不断修复
- 2, 为了留存用户, 根据消费者的反馈需要不断的增加一些新的功能或改善 UI 体验, 又或者阉割一些用处不大没有意义的功能
- 3, 有些软件也需要不断的维护才能保证软件的安全

维护比开发更花时间和经历, 要在后期想要很好的维护, 就必须在软件开发的时候做到代码规范易读, 有说明文档, 有注释, 命名要规范, 才能降低后期维护难度。

大作业评分标准

设计与实现（满分 80 分）：

70-80：图形界面实现，代码风格良好，运行正确，文档完备

55-70：非图形用户界面实现，代码运行正确，文档完备

40-55：基本实现了功能，代码及文档较为完备

40 以下：代码及文档有明显问题

英文文献（满分 15 分）：

10-15：查阅了大量英文文献

5-10：查阅了一些英文文献

0-5 分：几乎没有查阅英文文献

软件维护（满分 5 分）：

教师对学生报告里这一部分的描述酌情打分

提交方式：

2021 年 7 月 20 日之前把源代码（文件夹）和大作业报告书（单个文件）打包为压缩文件“学号姓名_大作业.rar”，上传至 [ftp://211.71.149.53/徐艳艳/课程作业/C++程序设计\(大作业\)](ftp://211.71.149.53/徐艳艳/课程作业/C++程序设计(大作业)) 文件夹中。如果上传后需要修改，则新传文件命名为“学号姓名_大作业_修改 1.rar”，以此类推，重新上传即可（教师只批改最后上传的文件）。