

## 一、程序功能功能介绍

本项目基于传统的俄罗斯方块游戏，游戏目标是堆叠从游戏区域顶部掉下来的方块，填充游戏区域的整行。当一行被填满时会被消去，此时记录玩家消去行的数量，达到一定程度后能升入新的 level。左光标键将当前件向左移动一个空格，右光标键将其向右移动一个空格，向上光标键逆时针旋转片子 90 度，向下光标键顺时针旋转 90 度。如不想等待，可按“D”使方块立刻下落一行，或者按空格键使其迅速沉底。每下落一个方块都能获得分数，消去一行能获得额外得分。本项目在基础上增加了游戏的选择界面，提高了游戏的界面美观度，增强了游戏的可玩性。

参考开题报告已实现功能：

方块的旋转

加速下落

自动积分与显示

自动消去

显示下一个方块

背景音乐播放

难度微调

暂停游戏等

## 二、项目模块

1.tetrixpiece 类:由 tetrixpiece 头文件和 cpp 文件实现。包含每个方块的信息

成员函数与成员对象:

构造函数与析构函数

枚举形状常量 enum Tetrixshape

stddots 静态成员变量, 由 8 个形状构成, 代表各种形状所对应的坐标。

minX () /minY () /maxX () /maxY () 方块所占的 x/y 坐标的最大/小值。

setShape () 将指定形状的方块坐标写入

setRandomShape () 0~7 内随机生成一个整数, 再将该整数对应的形状写入。

rotatedLeft () 向左旋转

rotatedRight () 向右旋转

Pieceshape 成员变量, 代表其形状

Shape () 返回对应的形状

X () 返回对应的 x 坐标

Y () 返回对应的 y 坐标

2.tetrixboard 类:由 tetrixboard 头文件与 cpp 文件实现。包含游戏逻辑, 处理键盘输入, 并在游戏区域显示方块

成员函数与成员对象:

构造函数与析构函数 构造函数为游戏设置用户界面元素

Start() 将状态初始化, 启动新游戏, 重置游戏状态, 分数, 关卡等。

Pause () 暂停内部计时器, 并且将画面定格并屏蔽 (防止利用暂停来策划), 再次点击回到游戏状态

Paintevent () 画出事件

setNextPieceLabel () 函数用于将外部构建的 piece 传达。

drawSquare()函数绘制块, 使用不同的颜色对不同形状的方格构成每个方块

Sizehint () 等含有 size 的函数用于控制尺寸。

nextPiece () 将下一个可用方块放在顶部, 并创建一个随机形状的新方块 (与 showNextpiece 联动)

showNextPlece () 用于展示下一个方块

tryMove () 函数用于确定一个部位是否可以定位在指定的坐标上, 检查方块所要占用的空间, 如果已被其他方格占据, 返回 false, 表示移动失败。

removeFullLines () 每下落完一个方块后使用, 从上到下扫描游戏区域寻找空格, 若一行全满, 则由上一行复制下来, 清楚板上的顶行, 并增加已消去的行数 LinesRemoved () 和 ScoreChanged () 函数。

PieceDropped () 负责向玩家授予积分与定位当前方块,

检查是否有整行等。

oneLineDown () 用于将当前方块下降一行，无论是按“D”时还是计时器到时方块自行下落。

dropDown()函数尽可能在棋盘上向下移动当前棋子，直到它接触到播放区域的底部或堆叠在另一个棋子上。

clearBoard()函数只需用 TetrixShape::NoShape 值填充板，用于将版清空。

timerEvent()用于处理每个时刻的事件，无事发生则方块下落一个，若一行刚刚填满，则创建新方块并重置计时器。

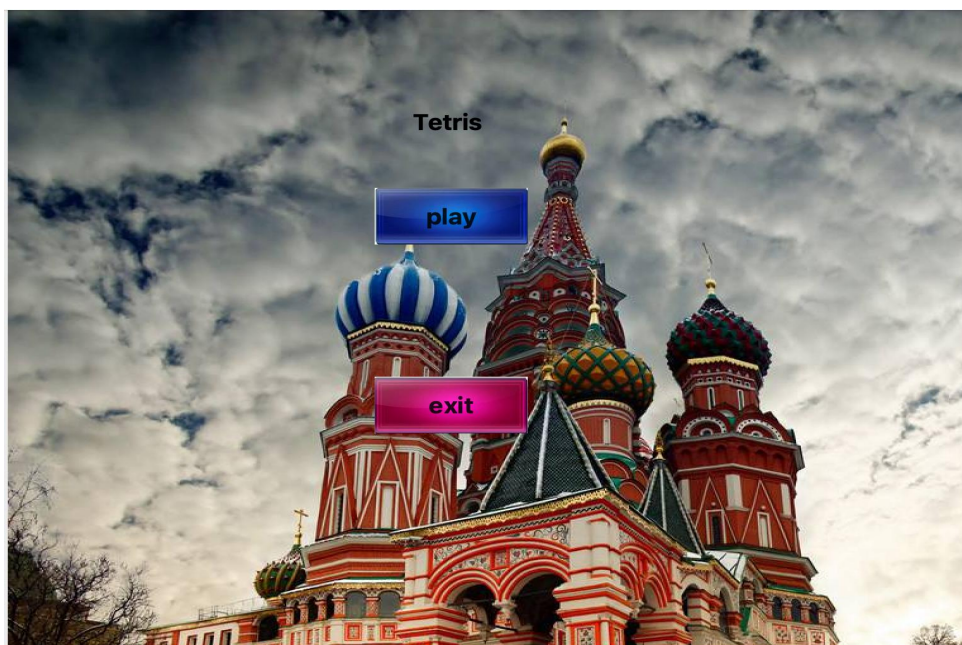
keyPressEvent () 每当玩家按下键盘都会调用，具有键盘焦点，会检查按键是否与上下左右空格D有关，无关则不管。

### 3.tetrixwindow 与 ui 界面

利用

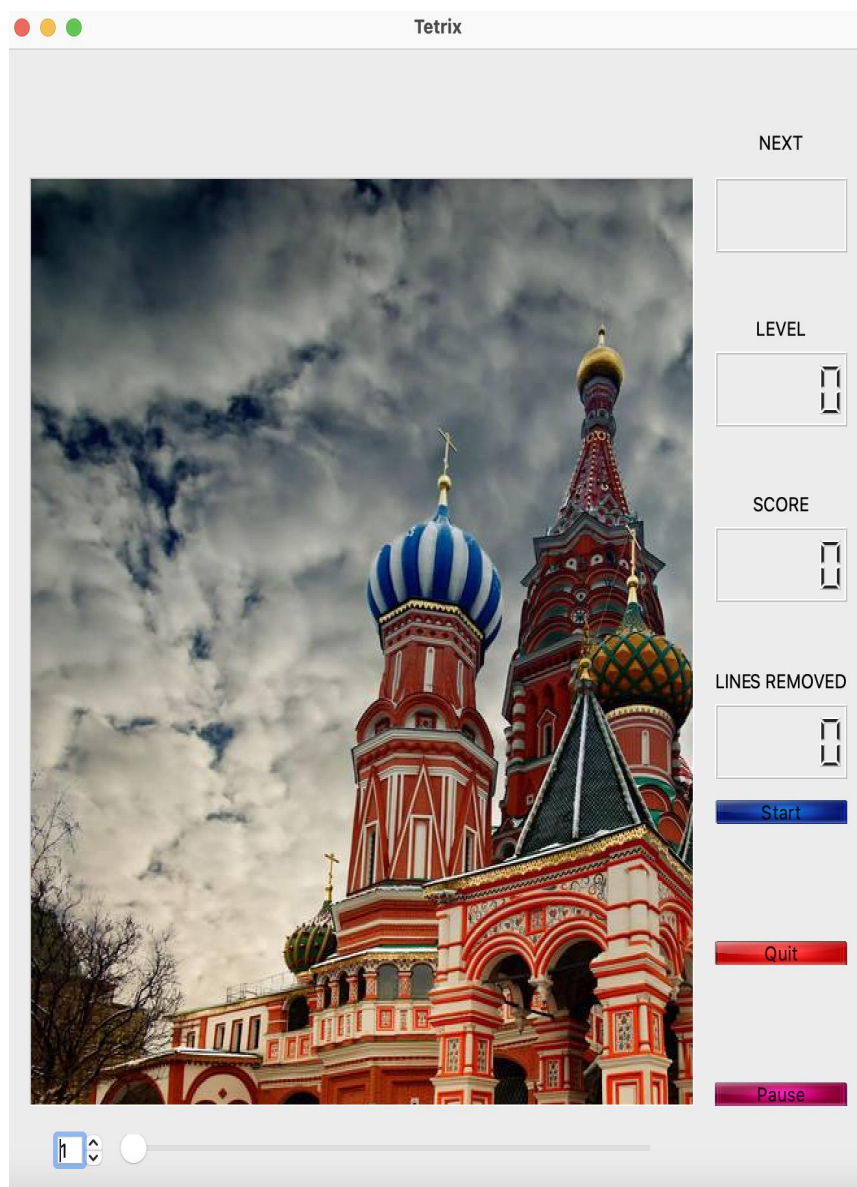
QMainWindow,QTime,QTimer,QMenu,QMediaPlayer,QFrame,QWidget,QSlider,QSpinBox,QSpacerItem 等一系列头文件并且将控件提升为已有的类。

开始点击程序后进入如下界面



点击 play 进入程序

点击 exit 退出。进入后，如图



其中 下方 QSpinBox 可以对下落速度进行微小调整

start 控件对应 tetrixboard 类的 start () 函数

Pause 控件对应 tetrixboard 类的 pause () 函数

并通过资源库进行了颜色的填充

next 框内负责显示下一个方块的形状，对应 tetrixboard 类的 showNextPiece () 函数

level 表示等级，与分数相对应，score 负责记录分数，分数由已经下落的方块数和难度共同决定，lines removed 记录已经消去的行数。

利用 qsound 头文件加入了程序的背景音乐，并采用了单曲循环模式 setloops (-1)

上方也通过 QMenu 设置了帮助栏，内有对该游戏的介绍“《俄罗斯方块》的基本规则是移动、旋转和摆放游戏自动输出的各种方块，使之排列成完整的一行或多行并且消除得分”

#### 4.程序图标

从互联网上下载 jpeg 图片并转成.ico 文件，在 pro 文件上添加 RC\_FILE+= tetrix\_ico.rc 并且创建 tetrix\_ico.rc 写入 IDI\_ICON1 ICON DISCARDABLE "tetrix\_ico.rc"





### 三、成员分工

尹哲晖：组长，负责分配任务检查进度、tetrixpiece 类的代码实现，代码 debug，调整组员代码风格一致，ico 图标文件的制作，项目报告的撰写。

韩卓衡：组员，负责 ui 界面与 tetrixwindow 的代码实现与界面设计，代码 debug 与调试。

徐轶：组员，负责 tetrixboard 的代码实现，代码 debug 与调试。

### 四、项目反思与总结

整个过程中，首先决定立项，参考项目代码量限制和人员水平，我们首先确定了方向偏向于做小型游戏，一段讨论后确定了做俄罗斯方块并撰写了开题报告。

此后经过一系列的讨论，我们决定将俄罗斯方块分为 3 个部分，主要负责功能实现的 tetrixboard，ui 界面与 tetrixwindow 的实现与 tetrixpiece。根据任务量的多寡，我们进行了分工。

此后在 6 月 1 日组员的代码依次完成，但由于每个人代码习惯的不一样导致要综合在一起有很大出入，各位在经过一系列的讨论后决定精简 tetrixpiece 将功能都放在 tetrixboard 里实现。

期末周并未进行讨论与整理，期末结束后各位组员依次将工作完成并上交。

总的来说,通过这次项目,组内各成员的编程能力与协作能力都得到了一定的提升,上网学习新技术,调试与 debug 的能力均得到了较大增强。

在项目管理方面,小组成员之间的沟通很重要,比如一开始 `tetrixpiece` 和 `tetrixboard` 类里存在很多重复的功能,造成了很多不必要的麻烦。如果写代码的过程中能及时沟通,大概率能避免此类情况。以及时间的把控上,把后续 debug 工作全放在期末考试后未免有点赶,小组成员大部分报名了招生工作,时间不是很充裕,如果存在着比较难找出的错误可能就不一定能按时完成任务了。