程序结构设计与分析

1. 本程序根据功能的不同分为Game类、ShuttleBoundary类和PlayRandomBlock类。

* Game类（基础类）
* ShuttleBoundary（穿越边界）
* PlayRandomBlock（随机方块）

1. 再由功能的不同抽象出不同的方块类。

* Block类为基础方块类，其中包括7种基础俄罗斯方块及每种的4种旋转状态，其中使用三维整型数组储存28种方块，第一维为方块类型，第二维为旋转状态，第三维为方块中小cell的具体位置（用0,1表示）。Block类为Game类及ShuttleBoundary类提供方块。
* RandomBlock类为随机方块类，随机方块在3 \* 5的矩形中产生，RandomBlock类为PlayRandomBlock类提供方块。

1. 而为了实现排行榜功能，创建Player类，记录玩家的名字及得分。
2. 方块类

**Class Blocks:**储存28种基础方块，分为7种类型及4种旋转状态

主要函数的功能

* getNumOfBlock():返回4x4cell方块中的小方块个数
* getRandomBlock():随机生成方块类型和旋转状态，返回1x12的方块
* nextRandomBlock():随机下一个方块的生成方块类型和旋转状态
* getNextRandomBlock():返回1x12的下一个方块
* applyNextRandomBlock():当前方块类型和旋转状态等于下一个方块的方块类型及旋转状态，返回1x12的新当前方块
* getNextRotatedBlock():生成下一种旋转状态，返回1x12的旋转90°后的方块
* rotateBlock():当前旋转状态等于下一种旋转状态，返回1x12旋转后的方块

**class RandomBlocks:**在3x5的矩阵中随机生成连通方块，判断旋转中心

* createMyBlock():当连通域不为1时，在3x5的矩阵中随机赋值0或1
* getCount():返回方块的连通域
* erase():判断方块是否连通
* actualBlock():得到3x5矩阵中方块的实际行列大小
* firstCell():得到3x5方块矩阵中第一个出现方块的行和列
* myBlock():生成实际大小的方块
* getCenter():得到生成方块的旋转中心
* setCell():生成随机方块，将生成的方块放入5x5的矩阵中，将方块的旋转中心与矩阵的旋转中心对齐或将方块的旋转中点与矩阵旋转中心的左上角点对齐，返回5x5矩阵（即随机方块模式中的单位方块）

1. 游戏类

**Class Game():**游戏的主界面设置，游戏模式的选择设置，基础模式的实现

使用25 \* 14的两个数组作为地图，分别储存当前正在下落的方块和已经落底的方块，其中数组前4行用于隐藏新的方块，使其最后一行能出现在地图中；左右分别有一列用于边界判断；最后一行用于判断是否落底。

* Start(Stage primaryStage):游戏的主界面入口；初始化游戏的背景音乐，游戏音效，背景音乐音量，游戏音效音量，方块的下落速度；添加按钮Start、Exit、Rank、Settings、Help、Location
* setGame(Stage stage):设置游戏界面，设置按钮Start、Quit、Continue、Pause，分别设置键盘监听；游戏结束时展示时间和分数的面板设置（该方法为游戏实现的主要方法）
* setStartTimerTask(Stage stage): 设置定时器控制方块自动下落,在模式2自动加速模式中，设置count控制加速的时间
* setContinueTimerTask(Stage stage):设置定时器控制方块自动下落，设置continue按钮被按下后的时间计算
* whetherLeave(Stage stage):当quit按钮被触发后，询问玩家是否确定要退出

设置confirm面板，设置按钮Yes（确认后回到主界面），No（继续游戏）

* inputName(Stage stage):如果玩家希望保留成绩到排行榜，则输入名字按ok按钮，添加玩家信息到PlayerList中排序，写入分数排行文件和时间排行文件
* setModePane(Stage stage):设置模式选择面板（基础模式、加速下落、随机方块、边界穿梭），添加按钮设置事件
* setGamePane():游戏界面设置
* setBoardPane():游戏背景界面设置
* setNextBlockPane():下一个方块预览界面设置
* setLblScore():设置分数Label
* setLblTime(Long time):计算时间，设置时间Label
* setLblTime():初始化时间label
* setButton(Button button,int X,int Y,int x,int y):设置按钮的位置、大小
* leave(Stage firstSceneStage):离开当前界面，回到开始界面
* initGameView():在gamePane中添加矩形，根据map的值选择颜色（地图）
* nextBlockView():在nextBlockPane中添加矩形，根据cell的值选择颜色（下一个方块预览）
* getFillColor(int i):根据不同的值返回不同的颜色
* copy(int[] block):将block类中返回的1x12矩阵赋给4x4的cell
* drawWall() 画地图边界
* location() 方块在地图中的位置
* createBlock() 初始化方块位置，并使方块最后一行出现于地图中
* newMap() 刷新储存当前下落方块的地图
* feasibility() 判断方块是否落到底
* countDown() 判断方块是否能下移
* countLeft() 判断方块是否能左移
* countRight() 判断方块是否能右移
* left() 方块左移
* right() 方块右移
* down() 方块下移
* bottom() 方块落底
* corresponding() 将落底的方块对应到map1中保持不动
* isGameOver():判断游戏是否结束，返回游戏是否结束
* afterGameOver(Stage stage):设置游戏结束界面，显示游戏时间和分数，设置按钮Back返回主界面
* rotate() 旋转方块
* bottomPlus() 提示方块落底位置
* newGames() 重新开始游戏时清除历史游戏数据
* destroyLine() 消行并算分
* playRankScore():将分数排序
* playRankTime():将时间排序
* read():读取txt文件中的玩家信息，并按分数排序添加到数组中
* readTime():读取txt文件中的玩家信息，并按时间排序添加到数组中

**Class ShuttleBoundary**

基于Game类，修改其中的location方法，将方块位置取模后对应到地图中以实现边界穿梭，其他方法基本与Game类相同。

**Class PlayRandomBlock**

基本与Game类相同，将Game类中4 \* 4的cell修改为5 \* 5，以包括3 \* 5随机方块的所有状态。

1. 设置类

**Class Settings**

设置设置界面，提供三种设置（背景音乐音量、游戏音效音量、背景音乐选择）

* Settings() 构造方法 初始化设置界面；添加组件（checkbox、slider、combobox）
* setDisPlay(int index) 根据combobox的值选择播放哪一首音乐

1. 帮助类

**Class Help**

设置帮助界面，提示玩家游戏操作方式

1. 玩家类

**Class Player**

玩家信息（名字、分数、游戏时间）

1. 音乐类

**Class GameSound** 游戏音效类

* playMusic() 方块移动时的音效
* playMenuSound() 点击菜单时的音效
* playGameOverSound() 游戏结束时的音效

**Class BGM** 游戏背景音乐类

* playMusic(String pathname) 游戏背景音乐

编程中遇到的问题

1. 对javafx还不熟悉，界面的设置包括stage的隐藏和显示、多个stage的组合、各类pane的不同特点、动画效果等还不够熟练
2. 继承的优势发挥的还不够大，代码有些重复
3. 存档读档还是不会写，I/O还不是很清楚