Project2开发文档

**一、结构设计与分析**

**1.**

类体规定了当前方块的整数型二维数组（4×4）、下一方块的整数型二维数组（4×4）、提示下落位置的整数型二维数组（4×4）、各类不同方块的整数型二维数组（4×4）、主界面的整数型二维数组.（16×13）、提示下一方块的界面的整数型二维数组（4×4）、方块初始位置的横纵坐标、分数值以及图片、音乐、各种父子结点、一个舞台。

注：方块数组与主界面数组中的0、1为信号，0代表此位置无方块，1代表此位置有方块。主界面数组中的2为下落位置提示的信号。

**2.**

程序主体由start（）方法、addblock（）方法、run（）方法组成。start（）方法运行时显示初始场景，场景中有各种按钮，点击“开始游戏”即可切换场景并开始游戏，其余按钮见《用户手册》。addblock（）方法作用为向新场景中加入合适的结点，并完成主界面的初始化工作。run（）方法是运行主程序，相当于主方法，其中有Timeline对象以实现定时下落，游戏逻辑部分见3。

**3.**

游戏过程中，首先运用clear（）方法与cleradown（）方法清除上一次操作时留下的的方块痕迹；

如果上次操作后的堆积方块出现可以消除行的现象，则调用clearline（）方法消除行，同时记录相应的分数；

之后玩家用command（）方法对玩家输入的相应指令控制当前方块的移动与变换；

方块在移动时，有下落位置的提示，需要getdownblock（）方法，获得下落位置在主界面中的位置；

之后用gettable（）方法，将当前主界面赋值给已经初始化的主界面的二维数组（16×13）；

提示下一方块需要用到getsmalltable（）方块，将下一方块的信息赋值给已经初始化的提示下一方块界面的二维数组（4×4）；

最后用一系列打印方法：printsmalltable（）、printScore()、Printtable()将提示下一方块的界面、分数与主界面打印出来；

如果方块达到了停止运动的条件，即stop（）方法返回值为true，则将下一方块nextblock赋值给当前方块block，并且清空提示下一方块的界面，再产生随机产生下一个方块的提示。同时把方块位置的横纵坐标重新调回初始值；

**二、主要函数功能**

**1.gettable（）方法：**

遍历当前方块block的4×4的数组，若发现信号“1”，则将该信号反映到主界面中，类似于物理中的伽利略坐标变换，Bigtable.getblock [centerX + i][centerY + j]代表主界面坐标与当前方块坐标的叠加。结果是在主界面中，这个位置被赋予1（代表此处有方块）。注：getsmalltable（）方法与此类似，只不过是nextblock与smalltable之间的关系，用于提示下一方块的界面；getdownblock（）方法也类似，只是1的位置变为2，用于提示方块的加入。

**2.printtable（）方法：**

遍历16\*13的Bigtable（主界面数组），若发现1则加入黑色图片，若为0则加入绿色图片，若为2则加入黄色图片（图片大小一致）。加入到mainGridpane中，实现可视化。（printsmalltable（）方法同理）

**3.random（）与nextrandom（）方法：**

用随机数方法取0-6之间的任意整数，将各类型方块与数字对应起来，然后分别将随机对应所得方块赋值给block与nextblock。

**4. initialize()与initializesmall()方法：**

将table数组与smalltable数组分别初始化，若不初始化，则数组内全为0。初始化为无方块的界面，这两个方法用于重玩功能。

**5.clear（）与cleardown（）方法：**

前者遍历当前方块block数组，若检测到信号“1”，则将Bigtable上的对应位置变成0，起到了将主界面原来为方块的位置清除的效果；cleardown（）方法，遍历整个主界面，当搜寻到2时，将该处变为0，起到了将下落位置提示清除的效果。

**6.spin（）方法**：

实现旋转。首先判断方块类型是否为长条，若是，则引入tempI数组，旋转中心为（1.5，0），运用数学中旋转90度的坐标变换，将当前方块block数组的0和1做相应调整；若不是，则旋转中心为（1,0），同理。注：若不能旋转，即到达边界或者左右有方块，则不能旋转。，

**7.command（）方法：**

根据相应的键盘事件setOnKeyPressed，相应地做出变换：按下A向左移动一格，按下D向右移动一格，按下W逆时针旋转，按下S向下移动一格，按下X直接下落，按下C清除最后一行。其中，在判断是否旋转时，首先判断当前方块是否为正方形方块，若是，则旋转无变化，若不是，则调用spin（）方法。

**8.over（）方法：**

判断当次游戏结束。引入一维boolean类型的数组over【15】，检测每i行是否有“\* ”，若有，则over【i】为true，当且仅当整个over数组的每个元素均为true时，代表游戏结束。

**9.stop（）方法：**

判断方块停止运动。检测到当前方块最下端为下边界或者还有方块时，方块停止运动。判断当前方块最下端还有没有方块时，需要调用thelastdig（j）方法，thelastdig（j）方法的返回值是当前方块最下端位于主界面上的行数，即横坐标。该方法需要传入参数j，即block数组中的的第几列，随后遍历当前方块block数组的每一行，若有信号“1”，则返回这个信号所在主界面的行数。由此，*table*[*thelastdig*(j)+ 1 ][*centerY* + j]代表了主界面上当前方块的最下端再下一行所在的位置，再用equals方法判断此位置是否为“\* ”或边界，以判断是否停止。

**10.judgement（）方法**：

返回值为-1或者1，用于“逆操作”：当方块无法再向左移时，会返回1；当方块无法再向右移时，会返回-1。若无法左移了，玩家还输入a，则在command（）方法中有centerY+=judgemendt（），相当于不移动。右移同理

**11.** **leftboundary（）方法**：

判断当前方块能否向左移动。检测到方块最左端到达左边界或者还有方块时，该方法返回true，供judgement（）方法选择返回1。theleftdig（i）方法的返回值是当前方块最左端方块位于主界面上的列数，即纵坐标。此方法需要传入参数i，即block数组中的第几行，随后遍历随后遍历当前方块block数组的每一列，若有信号“1”，则返回这个信号所在主界面的列数。由此table[centerX+i][theleftdig(i) - 1] 代表了主界面上当前方块的最左端再左一行所在的位置。

**12.rightboundary（）方法**：

判断当前方块能否向右移动。原理同leftboundary（），这里不解释。

**13.fall（）方法：**

返回当前方块离停止向下运动还需的的行数。运用while循环，调用stop（）方法判断是否停止运动，将当前位置的centerX保存在beforecenterX中，引入整数fall，若没有停下，则每次centerX加一，fall也加一。循环结束后，将beforecenterX赋值给centerX，相当于回到原位，然后返回fall，fall即当前方块离停止向下运动还需的的行数。此方法既可以实现方块的快速到达底部，又可以用于方块下落位置的提示。

**14.getsecond（）方法：**

该方法用于计时。首先用Calendar类创建一个对象，读取他的小时、分钟、秒，并计算此时对应的秒数，返回3600 \* hour + 60 \* minute + second。开始游戏时，调用一次该方法，计下开始游戏时的秒数，然后在Timeline中每次都调用该方法，用每次得到的秒数减去游戏开始时的秒数，得到已经游戏的秒数。

**15. clickbthelp()方法：**

该方法是点击“游戏帮助”按钮时的方法。利用JOptionPane类弹出一个弹窗，上面写游戏规则，实现游戏帮助。

**16.choose()方法：**

该方法用于游戏结束后供玩家进行选择会否继续游戏。若over（）方法判断到了该次游戏已经结束，则利用JOptionPane.showConfirmDialog弹出一个弹窗，写有“本次游戏结束，您的得分为 分，是否再次进行游戏？"。提供两个选项“是”与“否”：若选择是，则游戏重新开始，若选择否，则退出程序。

**三、遇到的问题及解决办法**

**问题1：**刚开始无法得到已经游戏时间，只能得到当前时间，并每次都更新。

后来查阅资料发现，有Calendar类可以返回当前时间的小时数、分钟数、秒数。由此想到可以采用在时间线中加如getsecond（）方法，以便每次都减去开始游戏时的秒数，得到的结果即已经游戏的时间。

**问题2：**刚开始时imageView将作为公共变量，然后加入图片时每次都加同一张图片，导致界面显示不出来。

查阅了书本，发现imageView不能每次都加同一张图片，要想循环加入同一张图片，必须在循环体中，每次循环都加入一个图片对象，每张图片都是一个对象，这样才能正常显示。

**问题3：**上述问题解决后发现游戏十分卡顿。

问题出在了每次都要去寻找图片的来源（getResource），速度慢。于是想到优化方法：将image作为公共变量，而循环时，只是加入imageView对象，这样子不比每次都去获取图片来源，加快了效率。

**问题4：**游戏音效刚开始反应不灵敏，按下相应键游戏音效响应慢。

与问题3类似，之前每次都去寻找音效的来源，反应很慢。后来把medieplayer作为公共变量，每次按键只是控制公共变量mediaplayer的play（）与stop（），这样子反应快很多。