学号：20172146 姓名：梁存镔 班级：软件171班

# 俄罗斯方块的设计与实现

**游戏分析：**

**1.首先了解规则**

（1）一共有7种不同形状的方块从游戏操作区域自由向下移动一个小格，每种方块由4个正方形的小方块组成。



（2）在方块落下的过程中，玩家可以旋转90°、左右移动方块，每次以一个小格子为单位进行移动。

（3）当方块落到游戏操作区最下方或者落到其他的方块上，则方块停止移动，游戏产生一个新的方块从游戏操作区域落下。

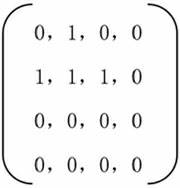
（4）当游戏操作区某一横行被方块全部填满，则该行可以消去，位于该行上面的所有行将落下，玩家可以得分，随着分数的增加，游戏级别上升，游戏级别越高，方块下落速度越快。

（5）如果固定区域的方块已经到达顶端而无法消除的时候，游戏就结束。

**2.方块分析**

每种方块由4种小的正方形格子组成，同时，在游戏运行过程中，玩家还可以旋转不同的方块。每种方块旋转后的形状也不一样，由于每个方块的形状最多为横向4个格子（如横向的I）以及纵向的4个格子（比如纵向的I），所以我们使用一个4×4的表格就可以描述所有的情况。

把地图划分为若干格子，有方块用1表示，其他没有方块的区域用0表示。使用4×4的二维数组表示，1表示方块的组成部分，0表示没有。下图是用 T 形描述。



这样子就比较浪费空间，所有我们把数组的每一行用十六进制数据描述，比如第1行0100可以表示成0x4，第2行表示为0xe，这样整个二维数组就可以使用一个十六进制数据0x4e00表示，每4位表示一行数据，从高到低分别表示1～4行的数据，这样，T型方块的所有状态可以使用4个数据表示：0x4e00、0x4640、0xe40、0x4c40。

**3.地图分析**

游戏操作区域也一样需要使用数据存储，我们使用20X12，即使用20X12列的二维数组来存储游戏操作区域。初始的时候，这个二维数组的所有数据都是0，表示没有方块，如果该位置的数据为1，则表示该位置已经有方块了。如下图可以使用000011100010来表示。同样地，如果直接使用二维数组存储则浪费空间，我们可以采用十六进制保存数据，可以使用0x0e2表示，于是，对于20×12的二维数组可以使用一个有20个数据的一维数组表示，每一个数据表示一行。接下来，我们需要处理一个方块在操作区域的移动了，在游戏进行过程当中，玩家可以使用按键移动方块。



**4.方块移动分析**

采取以下步骤：

（1）获取当前操作方块的第1行数据。

（2）判断该行和所在操作区域的行是否有重叠。

（3）如果有重叠则表示无法移动，退出。

（4）获取当前操作方块的下1行数据，重复步骤2～4，直到4行处理完毕。

这里有点不好搞的是第二步，如何判断该行和所在操作区域的行是否有重叠。如果满足条件：(data1|data2)!=(data1^data2)，即把两个数据分别按位或和异或，如果不相等则可以判断这两个数据是否有重叠，因为判断重叠，实质上是判断相同位置的数据是否都为1即可。

为了避免方块移动到操作区域两边边界处不好处理，所以我们使用16列表示每一行操作区域数据，即每一行的两边分别多加了两列，该处的数据用0xffff表示底部边界，所以最终操作区域的数据有22行16列组成。

**5.合并数据和产生新的地图**

当一个方块不能向下移动的时候，此时，需要把方块当前的数据和操作区域的数据合并以产生新的地图，同时，需要产生一个新的方块。

判断方块是否能继续往下移动，只需要调用canMove（方块的当前行+1，方块的当前列，方块数据）即可。

一旦判断方块无法向下继续移动，我们就需要采用以下步骤，合并当前的操作区域数据和方块数据。

（1）获取当前操作方块的第1行数据。

（2）把该行的数据和所在的操作区域行的数据按位相或，赋值给该行。

（3）获取当前操作方块的下1行数据，重复步骤2～步骤3，直到4行处理完毕。

接下来的工作就是产生一个新的方块了，我们可以在表示所有方块的二维数组中取出一个随机的数据就可以了。

最后就是当操作区域的某一行被方块全部填满时，该行被消除，只要判断操作区域的某一行数据为0xffff了，就表示该行被方块全部填满了，此时我们需要把该行的数据从数组中删除，然后在数组的第1行增加新的空数据就可以了

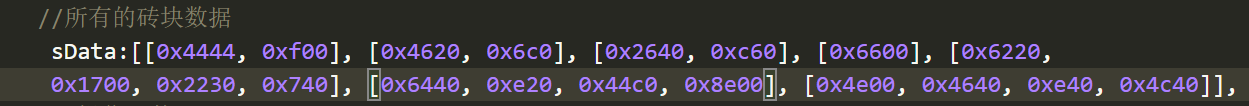
**游戏设计：**

1. 界面设计

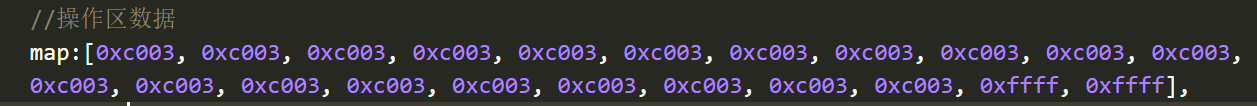


1. 方块设计

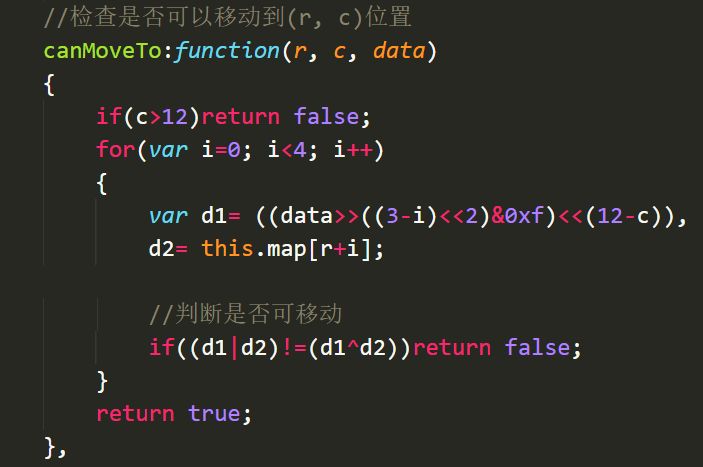
使用一个二维数组描述7种方块的所有形态。



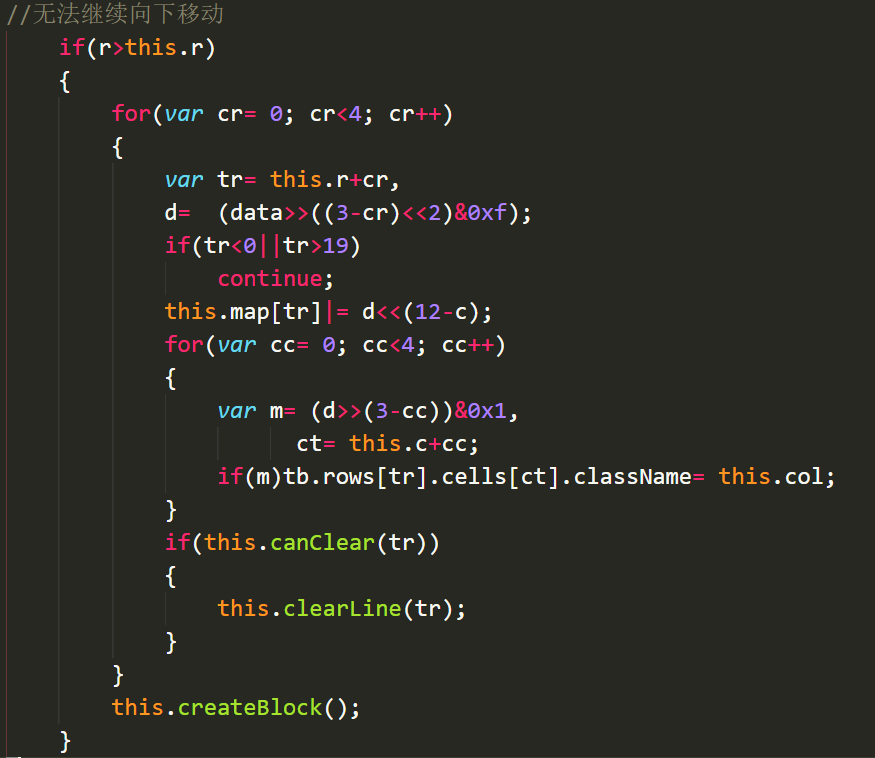
1. 操作区设计



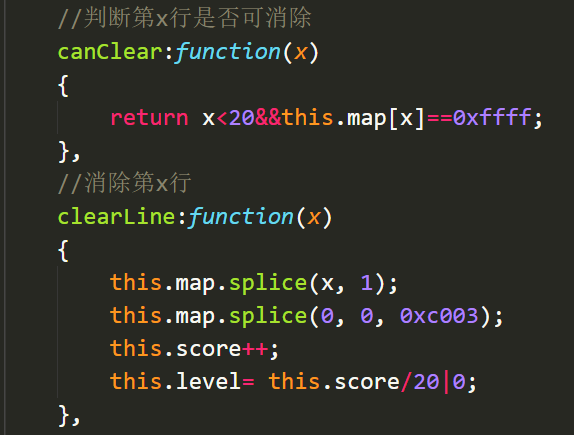
1. 对于任意一个方块是否可以移动到操作区域的某一行某一列。



1. 无法向下移动



1. 判断是否可以消除和消除



**总结：**

选做的游戏较为简单，但从分析到设计最后到实现的过程是不简单的。在这个过程中，很多时候都无从下手，我刚开始是一分析就动手，但一到动手的时候总会出现很多问题，这是我没有思考全面的原因，也很多时候想好怎么做了，一到敲代码实现的时候总会出现问题，不是代码实现很繁琐，难以后面的实现，就是容易缺斤少两，等后面用的时候就得重新改前面的代码。整个下来，很烦也很慢。这是我分析的时候没有分析好的问题。以后在设计的时候就得用伪代码形式处理一下。