俄罗斯方块设计

——1152745 邱峰

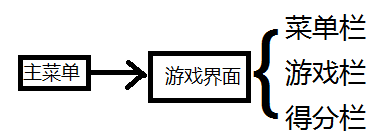
**程序功能：**

经典的俄罗斯方块游戏。

**游戏特色：**

* 难度级别选择
* 随着得分越高，方块落下速度越快
* 在困难模式下，速度没提升一次，最底层会多出一排方块
* 在游戏的左边有设置栏选项，在不暂停游戏时，设置属性不影响游戏进程
* 游戏设有高分榜，程序关闭后，高分榜不会消失
* 游戏设有背景音乐的选择
* 使用强大的位运算

**设计样式**



**方块类的设计**

任何一个方块基于一个4\*4的小方块block构成，如果block[i][j]处有方块，则new一个TextField，否则，该处指针为null

父类Block(抽象类)

该类中提供一系列对方块操作的函数

**public** **enum** BlockShape //枚举出了所有方块的类型

{

*LONG\_FOUR*, //----

*LEFT\_ONE\_THREE*, //|\_ \_ \_

*MIDDLE\_ONE\_THREE*,

*RIGHT\_ONE\_THREE*, //\_ \_ \_|

*LEFT\_TWO\_TWO*,

*RIGHT\_TWO\_TOW*,

*POINT*,

*SQUARE*

}

**public** **class** BlockPosition

//该类中提供接口：根据方块左下角的坐标定位到实际panel的坐标

{

**int** row;

**int** column;

**}**

**boolean** checkLeftMove() //检查能否左移

**void** leftMove() //向左移动

**boolean** checkRightMove() //检查能否右移

**public** **void** rightMove() //方块右移

**public** **boolean** checkDownMove() //检查能否下落

**public** **void** downMove() //下移操作

**public** **boolean** checkForm(**int** view) //检查是可以变化

由于所有的方块都是继承自父类Block，方块的操作都提供好了接口。因此，创建一个新的方块很简单，只需要设定好方块的形状即可。

强大功能——位运算

* 游戏每一层为20个方块，一共25层。于是，需要一个bitmap的数据结构存储游戏每一层的状态。
* 容易想到用一个boolean view[25][20]的数据结构表示目前已经堆好的方块的状态。
* 位运算的优化：一个int 类型占4个字节，有32位0/1数字。取最后20位用来表示每一层的状态，例如，当一个int x=(0100 0101 1001 1101 0001)B时，1的位置表示这个地方有方块，0的地方表示这个地方没有方块。因此，bitmap数据结构可以被优化到int view[25]

位运算与普通方法比较（1）

判断第i行是否可以消去

普通方法

采用boolean view[25][20]的数据结构

For (int j=0; j<20; j++)

{

If (!view[i][j]) return false；

}

Return true；

位运算

采用 int view[25]的数据结构

If (view[i]==(1<<20)-1)

return true;

Else return false;

* (1<<20)-1=1111 1111 1111 1111 1111， 即一行如果是满的时候的状态。

位运算与普通方法比较（2）

判断第i行哪些位置有方块

普通方法

采用boolean view[25][20]的数据结构

For (int j=0; j<20; j++)

{

If (view[i][j])

第i行第j列有方块

}

循环20次

位运算

采用 int view[25]的数据结构

Int x=view[i];

While (x!=0)

{

int column=x & (x ^ (x-1)) ;

则log2(column)列有方块

x=x-column;  
}

x&(x ^（x-1）)的意思是取出右数第一个1，例如110100，经过这个运算后得到100

第i行有多少个方块循环多少次

位运算与普通方法比较(3)

创建一个方块（这里创建的意思是设定好一个方块的形状，具体由另外一个接口创建)以田字型方块为例

普通方法

Boolean newBlock[4][4];

newBlock[2][2]=true;

newBlock[2][3]=true;

newBlock[3][2]=true;

newBlock[3][3]=true;

位运算

Int newBlock;

newBlock=102;

//102=(0110 0110)b

综上：位运算特色：减少代码量 以及优化代码复杂度，提升程序运行效率

**项目遇到的困难**

* 在刚刚开始做完俄罗斯方块的基本功能时，运行游戏，玩10分钟会崩溃一次。。。。
* 原因是在多线程处理的时候，比方说，方块正在下落的时候，用户按下了移动方块或者改变方块按钮的键。。这时候，出现很大的问题，导致坐标混乱，出现崩溃或者其他现象。

解决方法：后来加入了一个static变量 isBlockMove判断方块是否正在移动，如果在移动直接return。