# 编程接口说明

## （一）、运行与使用

1、运行环境：Windows(w64)。

2、解压缩“俄罗斯方块编程比赛\_1.0.0.zip”运行“启动俄罗斯方块.exe”。

3、操作界面及主要功能说明如下。



4、自行编写程序，调用接口控制俄罗斯方面左、右、下移动，旋转，落底销行结算。俄罗斯方块编程比赛会实时显示控制效果，计算时间，消除行，计算得分，显示相关信息和日志等。

5、俄罗斯方块编程比赛工具可以用W（旋转）S（下）A（左）D（右）空格（落底）P（暂停）来控制。

## （二）、接口说明

**此次俄罗斯方块编程比赛由“俄罗斯方块编程比赛.exe”程序提供RestFul接口，参赛选手通过调用接口控制俄罗斯方块操作，当前俄罗斯方块图形不会自动下落。**

**所有提供的RestFul接口，会按照请求到达时间先后，单线程顺序执行。**

**编程时应先调用重新开始新游戏接口进行复位。**

**如果需要使用浏览器调用接口应使用火狐或谷歌浏览器。**

所有接口说明如下：

### 1、顺时针旋转接口

接口地址：<http://127.0.0.1:8000/rotate>

参数：无

返回值：返回旋转后的数据结构，参见（三）数据结构部分返回值说明。

功能：控制当前图形顺时针旋转90度。

### 2、左移接口

接口地址：http://127.0.0.1:8000/left

参数：无

返回值：返回左移后的数据结构，参见（三）数据结构部分返回值说明。

功能：控制当前图形向左移动一格。

### 3、右移接口

接口地址：http://127.0.0.1:8000/right

参数：无

返回值：返回右移后的数据结构，参见（三）数据结构部分返回值说明。

功能：控制当前图形向右移动一格。

### 4、下移接口

接口地址：http://127.0.0.1:8000/down

参数：无

返回值：返回下移后的数据结构，参见（三）数据结构部分返回值说明。

功能：控制当前图形向下移动一格。

### 5、落底结算接口

接口地址：http://127.0.0.1:8000/apply

参数：无

返回值：返回落底结算后的数据结构，参见（三）数据结构部分返回值说明。

功能：控制当前图形按直线向下落到碰撞或触底为止，并同时进行销行、结算等处理。调用此接口意味着当前图形处理完毕，接口调用成功后，会自动将下一图形作为当前图形。

### 6、获取当前数据结构接口

接口地址：http://127.0.0.1:8000/data

参数：无

返回值：返回当前状态的数据结构，参见（三）数据结构部分返回值说明。

功能：查询当前俄罗斯方块所有数据信息。

### 7、重新开始新游戏接口

接口地址：http://127.0.0.1:8000/reset\_all

参数：无

返回值：返回重新开始新游戏时的数据结构，参见（三）数据结构部分返回值说明。

功能：重置所有信息和状态，并重新开始新游戏。

## （三）、数据结构

### 1、返回值

返回值数据结构是JSON格式，结构如下。

{

"end\_game": 0, // 游戏是否结束 ( 1 代表 GAME OVER )

"score": 0, // 当前分数, 统计单次消行数的平方之和

"line\_feed": 0, // 累计消除行数

"ground": { // 背景场地中有无方块的矩阵(坐标从上到下,自左而右).

"mat2d": [ // 当前场地的矩阵行列式 (10\*20)

[ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 ] , // 0 代表对应位置没有方块占据，1 代表有对应位置方块占据

[ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 ] ,

[ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 ] ,

[ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 ] ,

[ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 ] ,

[ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 ] ,

[ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 ] ,

[ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 ] ,

[ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 ] ,

[ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 ] ,

[ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 ] ,

[ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 ] ,

[ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 ] ,

[ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 ] ,

[ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 ] ,

[ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 ] ,

[ 0, 0, 0, 1, 0, 0, 0, 0, 0, 0 ] ,

[ 0, 0, 0, 1, 1, 0, 0, 0, 0, 0 ] ,

[ 0, 0, 0, 1, 1, 1, 1, 0, 0, 0 ] ,

[ 0, 0, 0, 1, 1, 1, 1, 1, 0, 0 ]

]

},

"spirit\_this": { // spirit\_this 字段指示当前形状

"x": 3, "y": -1, // 当前形状所在坐标为 (3,-1)，该坐标是指图形4×4矩阵中左上角格子的坐标

"type": 3, "rotate": 0, // 类型为 2（0到2）, 角度类型为（0到3）

"mat2d": [ // 当前形状矩阵行列式 (4\*4)，位置为1的部分形状是

[ 0, 0, 0, 0 ] ,

[ 0, 1, 1, 1 ] ,

[ 0, 1, 0, 0 ] ,

[ 0, 0, 0, 0 ]

]

},

"spirit\_next": { // spirit\_next 字段指示下一形状

"type": 0, "rotate": 0, // 类型为 0, 角度类型为 0

"mat2d": [ // 下一形状矩阵行列式 (4\*4)

[ 0, 0, 0, 0 ] ,

[ 0, 1, 1, 0 ] ,

[ 0, 1, 1, 0 ] ,

[ 0, 0, 0, 0 ]

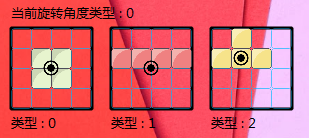
]

}

}

### 2、各种形状类型及数据结构说明

此次俄罗斯方块比赛中共有7种形状，每种形状对应的类型标号如下图所示。



#### 1、类型0，方块形状数据结构示例

旋转角0度

{ 0, 0, 0, 0 },

{ 0, 1, 1, 0 },

{ 0, 1, 1, 0 },

{ 0, 0, 0, 0 },

旋转角90度

{ 0, 0, 0, 0 },

{ 0, 1, 1, 0 },

{ 0, 1, 1, 0 },

{ 0, 0, 0, 0 },

旋转角180度

{ 0, 0, 0, 0 },

{ 0, 1, 1, 0 },

{ 0, 1, 1, 0 },

{ 0, 0, 0, 0 },

旋转角270度

{ 0, 0, 0, 0 },

{ 0, 1, 1, 0 },

{ 0, 1, 1, 0 },

{ 0, 0, 0, 0 },

#### 2、类型1，大棍数据结构示例

旋转角0度

{ 0, 0, 0, 0 },

{ 1, 1, 1, 1 },

{ 0, 0, 0, 0 },

{ 0, 0, 0, 0 },

旋转角90度

{ 0, 0, 1, 0 },

{ 0, 0, 1, 0 },

{ 0, 0, 1, 0 },

{ 0, 0, 1, 0 },

旋转角180度

{ 0, 0, 0, 0 },

{ 0, 0, 0, 0 },

{ 1, 1, 1, 1 },

{ 0, 0, 0, 0 },

旋转角270度

{ 0, 1, 0, 0 },

{ 0, 1, 0, 0 },

{ 0, 1, 0, 0 },

{ 0, 1, 0, 0 },

#### 3、类型2，T型数据结构示例

旋转角0度

{ 0, 1, 0, 0 },

{ 1, 1, 1, 0 },

{ 0, 0, 0, 0 },

{ 0, 0, 0, 0 },

旋转角90度

{ 0, 1, 0, 0 },

{ 0, 1, 1, 0 },

{ 0, 1, 0, 0 },

{ 0, 0, 0, 0 },

旋转角180度

{ 0, 0, 0, 0 },

{ 1, 1, 1, 0 },

{ 0, 1, 0, 0 },

{ 0, 0, 0, 0 },

旋转角270度

{ 0, 1, 0, 0 },

{ 1, 1, 0, 0 },

{ 0, 1, 0, 0 },

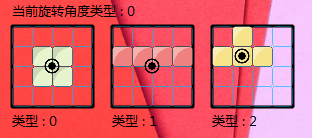
{ 0, 0, 0, 0 },

## （四）、关于随机数

此次俄罗斯方块编程比赛程序每次启动时，会按随机顺序生成1兆个图形，并顺序出现。调用重新开始游戏接口，不会改变图形出现顺序。可以使用菜单功能保存和加载图形序列，图形序列可以保存为文件形式。

## （五）、关于旋转

当前俄罗斯方块图形在旋转时，如果旋转后的状态与背景已有方块冲突，则不能旋转。每个图形出现时，初始旋转角度是1度。不同俄罗斯方块图形旋转轴不同，如下图所示各类型旋转轴用黑色点标记。



## （六）、关于图形

每次新俄罗斯方块图形出现时，初始坐标为（3，-1）。