

## 【实验题目】

### 3D 俄罗斯方块 Tetris

## 【小组成员】

杨曦华	10383039	10 计算机科学与技术 A 班	303258894@qq.com
马梓扬	10383003	10 计算机科学与技术 A 班	754142661@qq.com
刘旭峰	10383010	10 计算机科学与技术 A 班	296235396@qq.com
罗嘉文	10383058	10 计算机科学与技术 B 班	924878969@qq.com
杨智健	10383037	10 计算机科学与技术 A 班	695300838@qq.com

## 【实验内容】

使用 OpenGL 和 C++，实现 3D 版本的俄罗斯方块。除了实现俄罗斯方块的基本功能，还引入了新的游戏模式，同时采取令人舒适且可调节的观察视角，增强用户体验。

## 【实验目的】

1. 熟悉 OpenGL 的使用
2. 掌握一定的图形学算法，如：旋转、碰撞、着色、光照、消隐等
3. 提高结合图形学算法编程的能力

## 【开发环境】

1. 系统：Ubuntu
2. 程序语言：C++
3. 编译器：G++ compiler
4. 版本控制：GitHub

## 【实验展示】

## 【实验讨论与分析】

本次实验使用 OpenGL 库结合 C++ 编程，实现 3D 版本的俄罗斯方块。我们小组的项目主要分为以下几个阶段实现：

规划设计：对项目要实现的目标进行初步规划和模块分割，包括设计俄罗斯方块要具有的功能和体验、确定项目实现的架构和开发平台等；

算法实现：使用 C++ 实现俄罗斯方块相关的各类算法，包括边界碰撞、积木旋转、积木移动、行消除等；

2D 图像显示，使用 OpenGL 库对 C++ 函数计算出的点进行 2D 显示，先实现 2D 的俄罗斯方块，本阶段主要用于对项目的基本功能进行调试；

3D 图像显示，主要通过 OpenGL 库及数学运算，计算积木及单元方块的 3D 坐标并显示，本阶段主要对用户体验和游戏功能进行改善，引入了暂停、重新开始游戏、切换模式、旋转视角等高级功能，增强可玩性和体验。

由于俄罗斯方块本身的算法并不复杂，因此这个项目的难点主要集中在后两个部分。由于我们开发时使用的是最基础的 OpenGL 库结合 C++ 的方式，因此我们的项目在外观上并不出彩，但也正由于我们采用的工具更加接近底层，我们的组员通过这一项目的开发对图形学的各种算法也有了更直观的了解。例如，在最终的版本中我们实际采用的图形学算法就包括了图像消隐、几何坐标变换、区域填充、点阵字符显示等，而在开发过程中曾经尝试过，但由于效果不佳而没有出现在最终版本里的还包括了光照、纹理映射、位图读取与显示、显示列表等算法。

在游戏体验方面，我们引入的改进主要包括视角变换和重力模式。视角变换分为水平视角变换和垂直视角变换。在编写代码的过程中，我们注意到由于在 3D 显示中引入了视点，因此视角的变换会使得图像相对于视点的远近、方向等出现变化，于是我们加入了反向的移动抵消了这一波动；同时，视角变化会使得键盘操控出现变化，原方向的向前可能在水平变换后变为向左或向右移动，因此我们也加入了让键盘操控随着视角变换而改变的代码。重力模式是较为新颖的模式，当一个积木落地后，组成该积木的各方块不会留在原位置不动，而是会像受到重力的牵引一样继续下落到底；3D 俄罗斯方块的经典模式十分考验玩家的空间想象能力，玩一局十分耗费精力，也很容易出现操作失误，这个模式无疑降低了游戏难度，增加了可玩性，而重力模式与经典模式可以随时切换，这也使得想要锻炼空间想象力的玩家和希望通过这个游戏来放松身心的玩家都可以满足各自的需求。同时，保持显示在屏幕周围的各种提示也让刚接触的玩家就可以很快熟悉游戏的操作。

当然，如果时间充足，那么我们的项目还有一些改进的空间，例如引入更多更有创意和乐趣的模式，使用美观的素材并应用于纹理映射和光照，或结合 Qt 等图形界面库进行开发，从而扬长补短，使项目整体更加完善，也让自己有更多的机会接触和熟悉图形学的知识。

## 【小组成员实验心得】

**杨曦华：**

我们小组这次项目做的是 3D 俄罗斯方块。这次项目的难点主要不在俄罗斯方块本身的算法，而集中在如何用 OpenGL 对游戏进行合理的显示，增强用户体验。3D 的俄罗斯方块会互相遮挡，同时由于远小近大的影响玩家有可能难以判断当前方块的准确位置，因此我们运用了半透明、预测投影等技术，降低了玩家进行游戏的难度。同时，我们也包含了分数计算系统和重力模式，玩家在进行游戏的时候既可以提升难度锻炼自己的大脑，也可以降低难度放松心情。这次项目的主要收获是，要做好一个项目，不但要注重算法的应用，也要注意如何与用户进行交互，增强体验。

**刘旭峰：**

本来我们小组是打算做 2D 的俄罗斯方块并通过增加游戏模式来增强其游戏性的，但是由于老师后来的题

目要求便最终制作成了 3D。制作 3D 要比 2D 困难许多，我觉得这主要是难在如何让 3D 俄罗斯方块变得“可玩”。因为 3D 的方块会阻挡玩家的视线，移动时或堆叠起来后方块的形状也比较难分辨。所以，我们把方块做成了半透明，增加了各种方向的视觉旋转，还增加了方块的投影。之后在游戏的背景线条上也做了许多尝试。最后，我感觉我们做出来的这款俄罗斯方块是非常有趣的。这次的制作过程也让我深深体会到除了算法实现上的困难以外，如何提高游戏性与交互感也是十分巨大的困难。

### 罗嘉文：

我们小组的项目是 3D 俄罗斯方块，区别于 2D 俄罗斯方块，3D 俄罗斯方块不管在设计实现还是游戏操作上难度都更大。3D 视图中，方块的位置会相互遮挡，使得位于屏幕内部位置的空间无法正常显示，通过旋转视图后，可以一定程度上缓解这个问题。但是相应的就出现了方向键与当前视图不匹配的问题，因此方向键的设定需要做一些调整适应视图的旋转。其次，利用投影定位方块当前底部的位置，以方便玩家可以知道方块下落后的位置。项目使用 OpenGL 接口，实现方块的颜色跟立体显示以及透明处理等。

通过这次的项目，我学到了一些关于 OpenGL 的简单用法，进一步了解到游戏开发的难度和意识到 OpenGL 的强大。为了能利用 OpenGL 编写赏心悦目的真实感图形，还得系统学习以下 OpenGL 的使用以及一些图形学算法。

### 杨智健：

本小组选择的项目是做一个 3D 版的俄罗斯方块，俄罗斯方块大家都不陌生，2D 版的基本上都玩过，因而我们选择了有挑战性的 3D 版，同时也会把很多 2D 版上有趣的模式也移植到 3D 版上，如重力模式。

简单介绍一下我们的项目。在模式上，我们分有普通模式与重力模式，普通模式就跟一般的玩法一致；重力模式则是方块的每个小方块并不是固定的，会一直往下移动直到到达底层。因为做的是 3D 效果，所以会有视角转换功能，不会被最前方的方块挡住视野，提高可玩性。

项目使用 OpenGL 实现，虽然之前用 OpenGL 做过“行走的小人”，但俄罗斯方块在实现上要难得多，期间也遇到了很多障碍：

边界问题：方块在做变换的时候，有可能会超出边界，应对的方法是每次变换都做检测，要是超出边界则回滚不做变换。

视角转换：在视角做转换时，坐标系也会跟着变化，这时方块的移动操作方向会改变，所以在视角转换时，控制方块移动的函数要及时把移动的方向作相应的修改。

文字显示：界面上显示的文字的位图会在视角变换时跟着变化，因而在视角变换后得及时改变文字的位图，让文字正常地显示到界面上。

项目虽小，涵盖的内容却很多，整个项目做下来，对 OpenGL 有了更深入的了解，获益匪浅。