

# 虚拟现实技术 2021 秋 - 小作业 1

满分: 100

截止日期: 2021 年 10 月 15 日 23:59:59

## 1 书面部分 (50 分)

假设有三维空间点  $\mathbf{p} = (5, 2, 2)^T$ .

(i) (模型变换) 依次做如下模型变换

- x 坐标扩大为 2 倍, y 坐标和 z 坐标扩大为 3 倍
- 绕 y 轴旋转 180 度
- 沿 x、y、z 轴分别平移 2、1、3 个单位

分别写出缩放、旋转和平移矩阵(10 分); 计算  $\mathbf{p}$  经过变换后的坐标(5 分).

(ii) (视角变换) 接下来将世界坐标转换为相机空间坐标, 相机的摆放方式为:

- 相机位置  $\mathbf{e} = (4, 0, -3)$
- 相机看向点  $\mathbf{c} = (0, 0, 0)$
- 相机上方向  $\mathbf{u} = (\frac{3}{5}, 0, \frac{4}{5})$

写出视角变换矩阵(10 分); 对于 (i) 中模型变换后的点  $\mathbf{p}$ , 继续计算其经过视角变换后的坐标(5 分).

(iii) (投影变换) 透视投影参数如下:

- aspect: 1
- fovy:  $90^\circ$
- zNear(n): 3, zFar(f): 12

写出透视投影矩阵(5 分); 接着 (ii) 中的结果计算  $\mathbf{p}$  在截断空间中的坐标(5 分)和经过透视除法后的 NDC 坐标(5 分).

(iv) (视口变换) 假设有分辨率为  $300 \times 300$  像素的屏幕, 屏幕最左下方像素坐标为 (0, 0), 计算经过视口变换后  $\mathbf{p}$  点在屏幕空间中的坐标, 计算结果不需要取整(5 分).

## 2 编程部分 (50 分)

本次作业中, 你将基于给定的代码框架, 补全顶点坐标变换过程中模型变换、视角变换、投影变换的变换矩阵计算函数. 代码框架使用 Python( $\geq 3.6$ ) 语言编写, 借助 OpenGL 进行图形绘制. 请依照下述流程进行环境的配置, 并补全 transform.py 中留空的函数. 注意, 不需要修改除了 transform.py 以外的其他文件, 我们将调用该文件中的函数进行正确性检查.

### 2.1 环境配置

- 安装 Python  $\geq 3.6$
- 安装 Python 库依赖: `python3 -m pip install -r requirements.txt` (推荐使用虚拟环境)

- 运行主程序: `python3 main.py`

## 2.2 代码说明

`transform.py` 中, 有 7 个与变换相关的函数, 你需要补全除 `translate` 之外的 6 个函数, 它们均需要返回齐次坐标下大小为  $4 \times 4$  的变换矩阵, 类型为 NumPy 数组. 每个函数的功能和分值如下:

- `translate` 输入平移量, 计算平移变换矩阵. 该函数作为样例, 无需实现.
- `rotate` 输入旋转轴和旋转角度, 计算旋转变换矩阵. (15 分)
- `scale` 输入缩放系数, 计算尺度变换矩阵. (5 分)
- `modelTransform` 输入平移、旋转、缩放参数, 调用 `translate`、`rotate`、`scale` 计算模型变换矩阵. (5 分)
- `viewTransform` 输入相机位置、朝向点、上方向, 计算视角变换矩阵. (15 分)
- `orthogonalProjection` 输入正交投影变换相关参数, 计算正交投影变换矩阵. (5 分)
- `perspectiveProjection` 输入透视投影变换相关参数, 计算透视投影变换矩阵. (5 分)

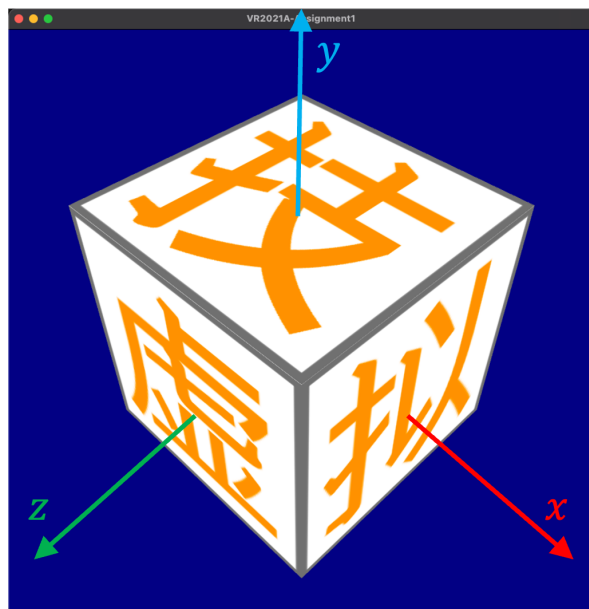
对于函数更加详细的输入输出描述, 请见代码注释. `transform.py` 中还提供了一些辅助函数以供使用:

- `identity` 输入大小  $N$ , 输出  $N \times N$  的单位矩阵.
- `zeros` 输入大小  $N$ , 输出  $N \times N$  的零矩阵.
- `normalize` 输入一维向量, 输出其归一化后的单位向量.

为方便调试, 框架提供了一个主程序 `main.py`, 渲染了一个立方体并可以在运行时动态改变模型、视角、投影变换的参数. 未补全代码直接运行 `main.py`, 你将看到如下画面:



在正确实现了 `transform.py` 中的函数后, 运行 `main.py`, 你将看到如下写着“虚拟现实技术”的立方体 (图中坐标轴为初始情况下世界坐标系的示意, 不出现在所渲染的图像中):



并且, 可以通过按键改变模型、视角、投影变换的参数:

- Q/W: 沿 x 轴平移
- A/S: 沿 y 轴平移
- Z/X: 沿 z 轴平移
- E/R: 绕 x 轴旋转
- D/F: 绕 y 轴旋转
- C/V: 绕 z 轴旋转
- T/Y: x 坐标缩放
- G/H: y 坐标缩放
- B/N: z 坐标缩放
- 1: 切换透视投影/正交投影
- 方向上/下: 上/下移动相机
- 方向左/右: 左/右移动相机
- O/P: 透视投影时向后/前移动相机; 正交投影时减小/增大渲染得到图形的尺度

如果以上操作可以得到正确的结果, 那么可以说明函数的实现基本正确.

## 2.3 参考资料

- NumPy 基础教程 [https://numpy.org/doc/stable/user/absolute\\_beginners.html](https://numpy.org/doc/stable/user/absolute_beginners.html)
- OpenGL 基础教程 <http://www.opengl-tutorial.org/> (本次作业不要求 OpenGL 相关知识, 仅供感兴趣的同学参考学习)

## 3 提交要求

将提交书面部分的作答 (电子版/手写拍照均可) 与代码打包上传至网络学堂. 样例文件结构如下所示. 如果使用 Python 虚拟环境, 请注意**不要**将包含虚拟环境的文件夹上传上来.

2019011262

```
├── 书面部分.pdf
├── code/
│   ├── transform.py
│   └── ...
```

如对本次作业有任何疑问, 请邮件联系助教或在微信群中提问. 助教联系方式:

- 郭元晨 [guoyc19@mails.tsinghua.edu.cn](mailto:guoyc19@mails.tsinghua.edu.cn)
- 王 琛 [wchen20@mails.tsinghua.edu.cn](mailto:wchen20@mails.tsinghua.edu.cn)