

浙江大学计算机图形学

课程实验报告（2023-2024 年秋冬）

姓名：展翼飞 学号：3190102196 专业：计算机科学与技术

1. 实验内容

1.1 实验内容

了解 OpenGL 中的坐标系与投影变换，熟悉模型坐标系，世界坐标系以及正投影，透视投影的概念以及它们在 OpenGL 中的基础运用，并使用透视变换绘制三维的会动的后三位学号。

1.2 预期结果

绘制窗口上有 196 三个立体数字，左边数字在其模型坐标系 y 轴上移动，中间数字绕 y 轴旋转，右边数字从小到达缩放，效果类似给出的 demo.mp4。

2. 实验过程

2.1 熟悉相关函数

```
glutReshapeFunc(); //用于注册窗口变换回调函数
glutPostRedisplay(); //标记当前窗口需要重新绘制
glutIdleFunc(); //注册没有窗口事件时的回调函数，配合实现不断重绘
glMatrixMode(); //选择当前矩阵模式为投影，模型视图或纹理矩阵
glPushMatrix(); //保存当前的坐标系矩阵至栈
glPopMatrix(); //将之前保存的坐标系矩阵出栈，与 push 配合可以用作
                //保存现场，创建一个临时的坐标系矩阵进行绘制，然后
                //再恢复现场
glutInitDisplayMode(GLUT_RGBA | GLUT_DOUBLE) //参数 GLUT_DOUBLE 与绘制
                函数中的 glutSwapBuffers() 设置双缓冲绘图，使动画
                更流畅
```

2.2 尝试绘制

使用 `glBegin()` 与 `glEnd()` 中的 `GL_LINES` 与 `GL_LINE_LOOP` 绘制三维数字的边框与相应连线。

```
//画6前方外边框, 宽度0.1, 高度为1
glBegin(GL_LINE_LOOP);

glVertex3f(-0.5, -1, -0.1);
glVertex3f(-0.5, 1, -0.1);
glVertex3f(0.5, 1, -0.1);
glVertex3f(0.5, 0.8, -0.1);
glVertex3f(-0.3, 0.8, -0.1);
glVertex3f(-0.3, 0, -0.1);
glVertex3f(0.5, 0, -0.1);
glVertex3f(0.5, -1, -0.1);

glEnd();

//画6后方外边框
glBegin(GL_LINE_LOOP);

glVertex3f(-0.5, -1, 0.1);
glVertex3f(-0.5, 1, 0.1);
glVertex3f(0.5, 1, 0.1);
glVertex3f(0.5, 0.8, 0.1);
```

使用`<time.h>`给定的计时函数原型给绘制函数提供基于时间变换的值, 以实现动态效果。

```
//通过经过时间计算图形的变换信息
//以一秒为1个周期 最左边中心从-1移动至1 中间绕y轴旋转90° 左右边图形从0.5
clock_t duration = clock() - tStartTime;
float sec = (float)(duration % (CLOCKS_PER_SEC)) / (CLOCKS_PER_SEC);
fTranslate = -1 + sec * 2;
fRotate = (duration / 6) % 360;
fScale = 0.5 + 0.5 * sec;

// 如果想使用线框模式
glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);

glClear(GL_COLOR_BUFFER_BIT);
glLoadIdentity(); // 重置变换矩阵

//左边
glPushMatrix(); // 设置一个新的坐标系
//where you need to write 沿着y轴上下运动
glTranslatef(0, fTranslate, 0);
glTranslatef(-2.0f, 0.0f, -6.0f); // 向左平移
Draw1();
glPopMatrix(); // 回到前一个坐标系
```

3. 实验结果

绘制窗口上有 196 三个立体数字，左边数字在其模型坐标系 y 轴上移动，中间数字绕 y 轴旋转，右边数字从小到达缩放，效果类似给出的 demo.mp4，其静态效果如下图所示：

