

# 浙江大学实验报告

专业：\_数字媒体技术\_

姓名：\_杨锐\_

学号：\_3180101941\_

日期：\_4.23\_

地点：\_

课程名称：\_计算机图形学\_ 指导老师：\_唐敏\_ 成绩：\_

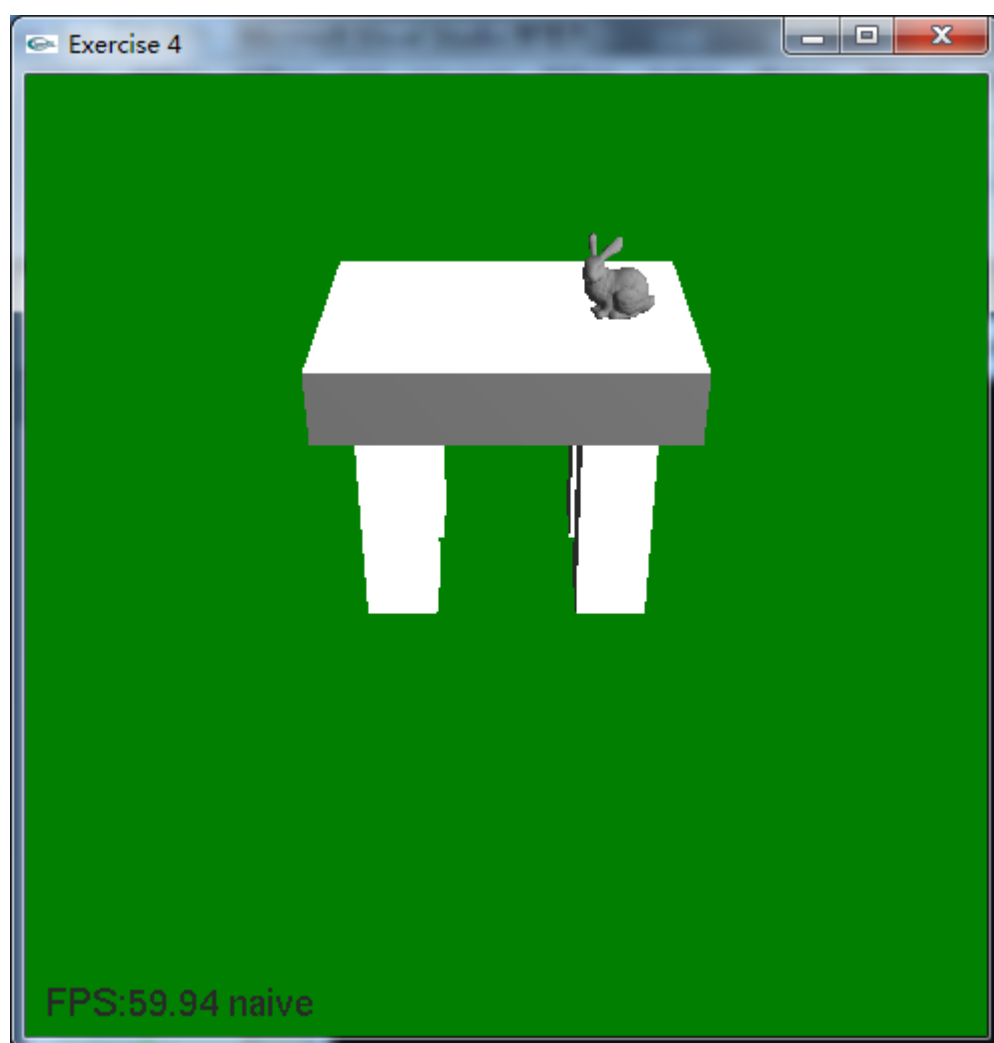
实验名称：\_OpenGL 显示列表\_ 实验类型：\_基础实验\_ 同组学生姓名：\_

## 一、实验目的和要求

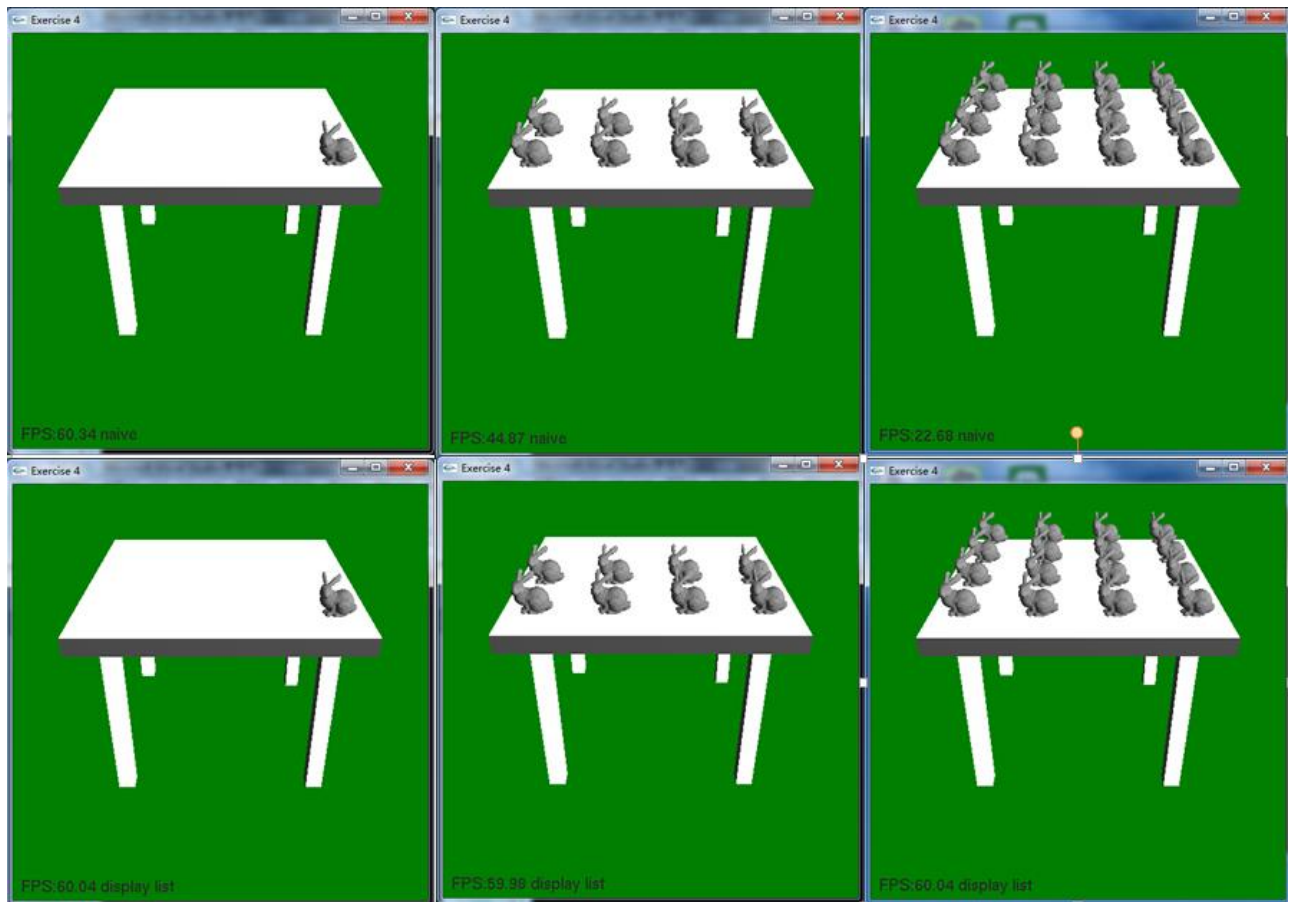
在三维观察实验的基础上，通过实现下述实验内容，掌握 OpenGL 中显示列表的作用和使用方法。

## 二、实验内容和原理

使用 Visual Studio C++编译已有项目工程，



修改代码，通过**键盘按键**，控制兔子的数量（1 至 16 个）以及整个场景的渲染模式，生成以下图形：



用按键 I、K 添加兔子数量增减（所有兔子均摆放着在桌面上，兔子间不要有交叉，桌面不够大可自行调整），按键 L 来切换显示列表和非显示列表绘制方式。WASDZC 控制上下左右前后移动, 空格键控制整体旋转。

通过动画以及对 FPS 的理解和分析显示列表对程序绘制性能的影响。

### 三、主要仪器设备

Visual Studio C++

glut.zip

Ex4-vs2010 工程

## 四、操作方法和实验步骤

### 1.控制兔子的增多减少

使用全局变量count控制当前兔子数量，count的范围限制在[0, 16]。使用for循环控制兔子的绘制，每绘制一只兔子后使用 `glTranslatef(0, 0, 0.5)` 平移到下一个位置。每绘制完四只兔子。使用 `glTranslatef(-0.6, 0, -1.5)` 平移到第二行。

对于非显示列表的模式，直接调用绘制函数；对于显示列表模式，程序开始时先使用 `GLint lid = glGenLists(2)` 创建两个空列表并返回第一个列表的索引，然后使用 `glNewList (GLuint list, GLenum mode)`，将兔子的节点和像素数据放入列表的内存中，最后在绘制的时候调用 `glCallList (GLuint list)` (参数为对应列表索引)将内存中的数据显示出来，而不再需要每一帧都重新执行、传输数据。

### 2.fps的计算

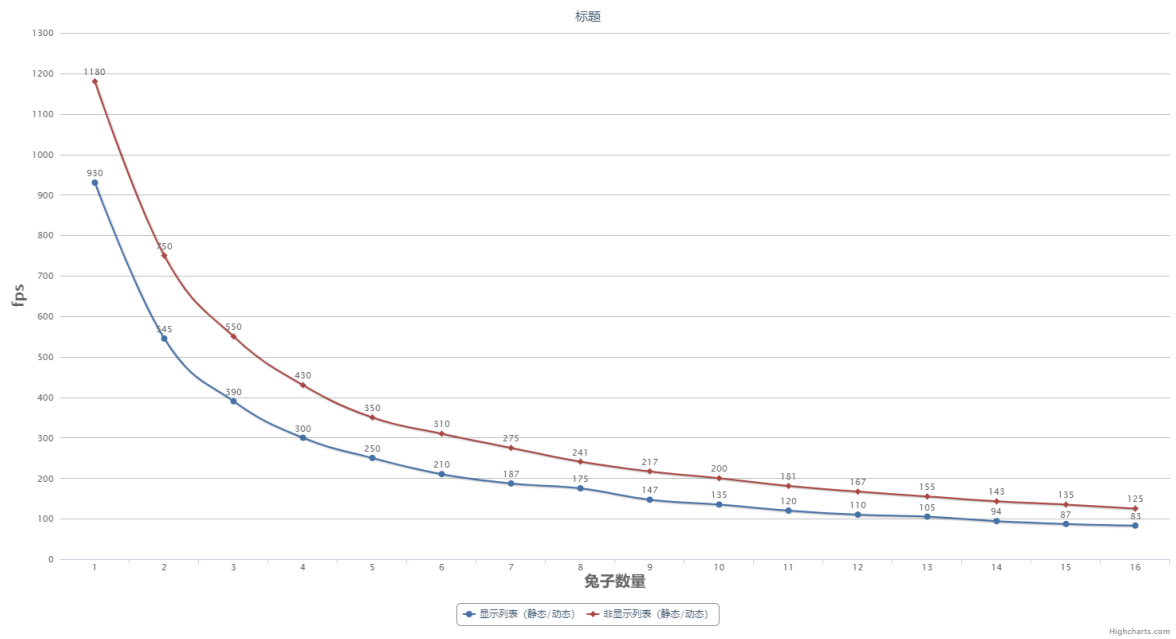
fps是frames per second的缩写，是一种衡量动画（泛指一切motion video）信息储存、显示量大小的指标。每一帧都是静态图像，将每一帧快速、连续的显示出来，就产生了运动的错觉。这也是OpenGL的执行流程。fps直观的反应了动画的流畅程度，对于人眼而言，30-60fps最为适宜，超过60，直观上流畅度没有太多影响，但动画的信息量增加，成本增加。低于30，会有明显jerky motion的感觉。

因此，要计算fps需要计算单位时间内调用 `redraw()` 函数的次数，使用一个frame变量记录帧数，使用 `glutGet(GLUT_ELAPSED_TIME)` 返回两次调用之间的时间间隔（ms），当时间达到单位时间（1s）时，更新fps的值，并通过 `glutBitmapCharacter(GLUT_BITMAP_HELVETICA_18, *c)` 显示在窗口中

## 五、实验数据和处理

### 1.数据

兔子数量/绘制模式/状态	显示列表（静态/动态）	非显示列表（静态/动态）
1	930/896	1180/1100
2	545/500	750/670
3	390/370	550/500
4	300/285	430/400
5	250/230	350/330
6	210/197	310/280
7	187/171	275/250
8	175/150	241/220
9	147/142	217/198
10	135/125	200/ 189
11	120/115	181/170
12	110/107	167/157
13	105/99	155/150
14	94/91	143/135
15	87/85	135/130
16	83/80	125/123



注：

- 1.fps数值取稳定后平均值
- 2.动态包括：缩放、旋转、键盘、鼠标控制移动（相当于模拟一个FPS相机）

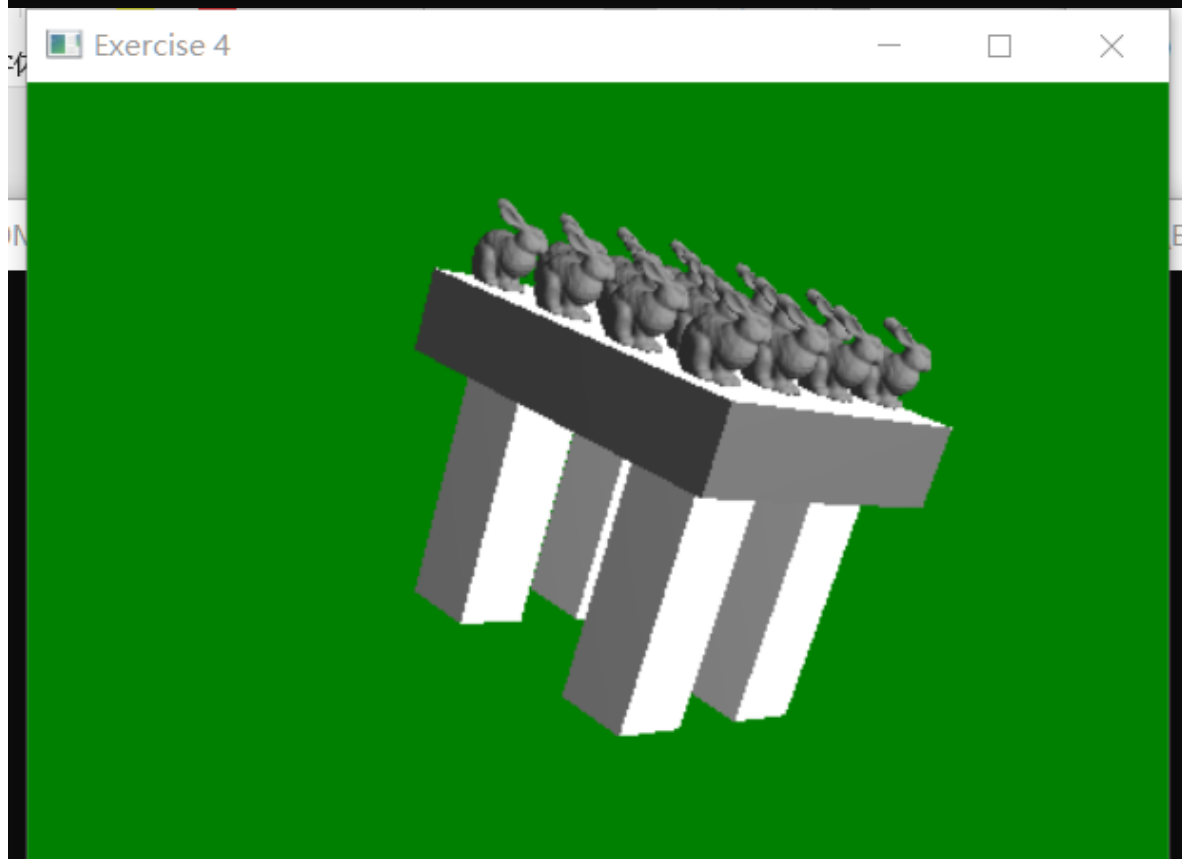
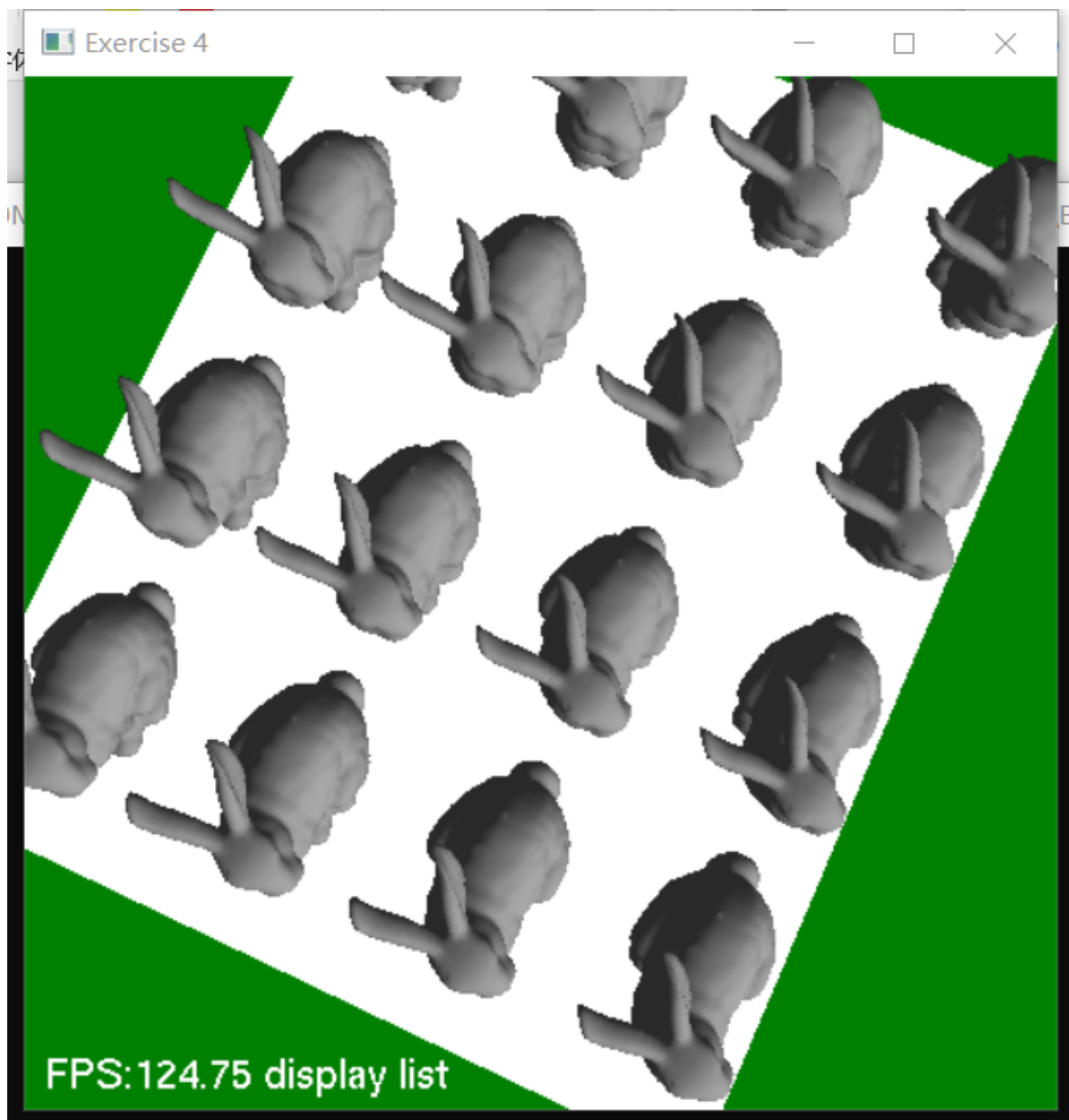
### 3.曲线图绘制fps取静态

## 2.分析

从实验数据结果可以看到，相同模式下，兔子数量越多，动态操作越多，fps越低，不同模式下，显示列表明显高于非现实列表。

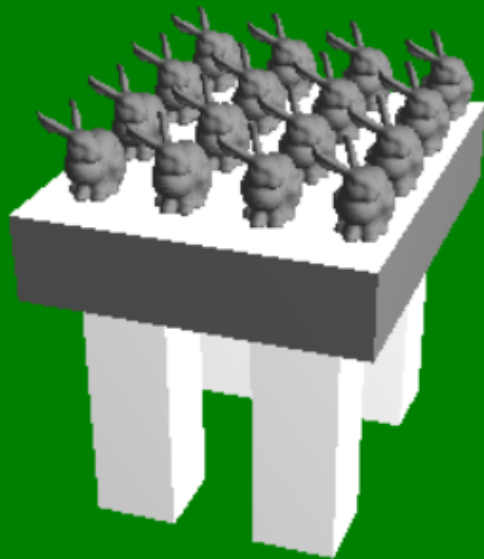
.而显示列表的原理是提前将顶点和像素数据计算、编译并储存在内存中，之后可以直接调用而不需要再次计算、传输，并减少cpu的执行时间，并且可以将数据上传到服务器端实现多客户端共享。因此显示列表非常适合绘制静态数据的图像，特别是时间开销的大操作如：光照、纹理等。但缺点也是非常明显的，就是一旦编译后，数据就不能再修改，如果需要频繁改变数据，可以使用**Vertex Buffer Object**.

## 六 实验结果和心得



FPS:82.50 naive

Exercise 4



FPS:125.37 display list

这次实验练习了显示列表的使用，并对fps的计算有了一定理解，同时改进了上次实验的鼠标操作。