

## 实验 1 实验报告

实验人	刘沅昊	学号	15331220
学院	数据科学与计算机学院	年级专业	16 级软件工程(数字媒体技术)

### 实验内容：

## Homework

### Basic:

回答下面的问题：

1. 结合上述参考链接（或其他参考资料），谈谈自己对计算机图形学的理解。
2. 结合上述参考链接（或其他参考资料），回答什么是OpenGL? OpenGL ES? Web GL? Vulkan? DirectX?
3. `gl.h` `glu.h` `glew.h` 的作用分别是什么?
4. 使用GLFW和freeglut的目的是什么?
5. 结合上述参考链接（或其他参考资料），选择一个SIGGRAPH 2017/2018上 你最喜欢的专题，介绍该专题是做什么的，使用了什么CG技术？（不少于100字）

### 实验结果：

1. 计算机图形学涵盖面积特别的广，不好对其进行严格的定义，但主要的目的和内容都是大同小异的。不过多数地方都提到这门学科开创之初是为了解决如何在计算机中表示三维几何图形，以及如何利用计算机进行图形的生成、处理和显示的相关原理与算法，产生令人赏心悦目的真实感图像。不过也包括了二维图形的处理，只是因为现实生活中三维图形比较常见而被人们忽略了。

从文章《什么是计算机图形学？》中我初步了解到了计算机图形学是什么，以及计算机图形学主要内容：建模、渲染、动画和人机交互，每个过程都有自己的特点。这里面每个过程的具体内容还有待深入学习。不过几何计算贯穿在各个过程之中，所以可以说数学是计算机图形学的基础。除此之外，编程也是计算机图形学的基础，数学是基础，编程是手段，生成图像是目的。

计算机图形学的应用邻域非常广，医疗、影视、游戏、科学、建筑等领域都有它的身影，所以我认为这学期学好，至少是入门这门学科是非常有意义的。

2. **OpenGL:** OpenGL (Open Graphics Library) 是一个由 Khronos 组织制定并维护的规范

(Specification)。不过我在百度百科和维基百科查到都说它是一个 API，但因为 OpenGL 库基本上都是开发者实现的，所以可能会出现一些 bug。

OpenGL 的核心模式和立即渲染模式，我感觉就类比操作系统原理学到的大核小核一样。核心模式和小核操作系统一样，可扩展性更强，对使用者的要求更高。

**OpenGL ES:** OpenGL ES (OpenGL for Embedded Systems) 是 OpenGL 三维图形 API 的子集。顾名思义，它是针对手机、PDA 和游戏主机等嵌入式设备而设计，比如，iPad、iPhone、Android 等等。

**WebGL:** WebGL (Web Graphics Library) 是一种 3D 绘图协议，这种绘图技术标准允许把 JavaScript 和 OpenGL ES 2.0 结合在一起，通过增加 OpenGL ES 2.0 的一个 JavaScript 绑定，WebGL 可以为 HTML5 Canvas 提供硬件 3D 加速渲染，这样 Web 开发人员就可以借助系统显卡来在浏览器里更流畅地展示 3D 场景和模型了，还能创建复杂的导航和数据视觉化。WebGL 技术标准免去了开发网页专用渲染插件的麻烦，可被用于创建具有复杂 3D 结构的网站页面，甚至可以用来设计 3D 网页游戏等等。

**Vulkan:** Vulkan 是一个低开销、跨平台的 3D 图形和计算 API。Vulkan 面向所有平台的高性能实时 3D 图形应用，如视频游戏和交互式媒体。与 OpenGL 和 Direct3D 11 以及 Direct3D 12 和 Metal 相比，Vulkan 旨在提供更高的性能和更均衡的 CPU/GPU 使用。

**DirectX:** DirectX 是应用程序编程接口 (API) 的集合，用于在 Microsoft 平台上处理与多媒体（尤其是游戏编程和视频）相关的任务。由 C++ 编程语言实现，遵循 COM。被广泛使用于 Microsoft Windows、Microsoft XBOX、Microsoft XBOX 360 和 Microsoft XBOX ONE 电子游戏开发，并且只能支持这些平台。

3. **gl.h:** OpenGL 所使用的函数和常量声明。

**glu.h:** GLU (OpenGL 实用库) 所使用的函数和常量声明。GLU 库属于 OpenGL 标准的一部分。

**glew.h:** GLEW 是一个跨平台的 C++ 扩展库，基于 OpenGL 图形接口。GLEW 能自动识别你的平台所支持的全部 OpenGL 高级扩展函数。也就是说，只要包含一个 glew.h 头文件，你就能使用 gl, glu, glext, wgl, glx 的全部函数。

4. **glut** (OpenGL 实用工具库) 为我们提供了一个统一的接口，方便我们进行窗口管理，IO 监听，还包括一些三维物体绘制函数，用来绘制特定的更为复杂的物体，像球体、圆面体、茶壶等。freeglut 是 glut 的一个开源版本，一直有专门的人员维护和更新。freeglut 完全兼容 glut，算是 glut 的代替品，功能齐全，但是 bug 太多。稳定性也不好，GLFW 应运而

生。

GLFW 提供了一个轻量级的, 开源的, 跨平台的库。支持 OpenGL、OpenGL ES 及 Vulkan, 它提供了简单的 API 来管理窗口, 读取输入, 处理事件等。

5. 近年来, 要说热门的内容, 当然还是 VR 了, 不管是 2017 年还是 2018 年的 SIGGRAPH 上, VR 都是最吸引人的。因为 VR 的发展, 推动了硬件设备的更新换代。我个人比较喜欢玩游戏, 对显卡的了解就比较多, 英伟达去年发布的新显卡架构使用了图灵架构, 能够支持光线追踪, 本来我是不太清楚什么是光线追踪的, 这学期我才知道这是计算机图形学里面的内容。

因为比较好奇什么是光线追踪, 自己也在网上查了一些不是很准确的解释。在真实世界中, 我们看到的 3D 物体被光源照亮, 且光子可以在到达查看者的眼睛以前从一个物体反弹到另一个物体。光线可能会被某些物体阻挡, 形成阴影, 或可能会从一个物体反射到另一个物体。比如我们看到一个物体的图像反射在另一个物体表面的情景。然后会发生折射 - 光线穿过透明或半透明物体 (如玻璃或水) 时发生变化的情况。光线追踪通过从我们的眼睛 (观景式照相机) 反向追踪光线捕捉这些效果。

我的理解是光线在传播的时候可能会被反射、折射等经由不同路径到达物体, 所以光源的光线, 有一部分是直接照射到物体 P1 上, 有一部分是经过其他物体 P2 反射到物体 P1 上, 而物体 P2 的材质、光线角度可能都会影响反射光线的强度, 而光线追踪算法就是将这些所有光线叠加起来得到物体 P1 表面的光线强度。这个光是想一想就觉得计算量非常大, 所以目前的一些做法是对收益明显的光源使用光线追踪算法, 而其他的使用简单的投影等算法。