

Homework 1 - Learning about CG

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。

计算机图形学的主要研究内容就是研究如何在计算机中表示图形、以及利用计算机进行图形的计算、处理和现实的相关原理与算法。计算机图形学主要包含四大部分内容：建模（Modeling）、渲染（Rendering）、动画

（Animation）和人机交互（Human-computer Interaction, HCI）就我自己的理解来说，计算机图形学就是在计算机上操作图形，对图形进行处理，以输出想要的效果。通过建模，渲染和动画来完成计算机对图形的一系列操作，至于人机交互，可以插入这三个步骤中的任意一个来实现更好的操作。

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?

老师在课堂上简单地将OpenGL解释成一个API，可能我们在使用过程中确实也能将它当作API来使用，使用其中一系列的操作图形、图像的函数。然而，OpenGL本身并不是一个API，它仅仅是一个由 [Khronos组织](#) 制定并维护的规范 (Specification)。

早期的OpenGL使用立即渲染模式 (Immediate mode，也就是固定渲染管线，这个模式下绘制图形很方便，容易使用和理解，但是效率太低。由于OpenGL的大多数功能都被库影藏起来，开发者不能控制计算自由，也不知道OpenGL具体是如何运作的。

从OpenGL3.2开始，规范文档开始废弃立即渲染模式，并鼓励开发者在OpenGL的核心模式(Core-profile)下进行开发，这个分支的规范完全移除了旧的特性。当使用OpenGL的核心模式时，OpenGL迫使我们使用现代的函数。现代函数要求使用者真正理解OpenGL和图形编程，它有一些难度，然而提供了更多的灵活性，更高的效率，更重要的是可以更深入的理解图形编程。

OpenGL自身是一个巨大的状态机(State Machine)：一系列的变量描述OpenGL此刻应当如何运行。OpenGL的状态通常被称为OpenGL上下文(Context)。我们通常使用如下途径去更改OpenGL状态：设置选项，操作缓冲。最后，我们使用当前OpenGL上下文来渲染。当使用OpenGL的时候，我们会遇到一些状态设置函数(State-changing Function)，这类函数将会改变上下文。以及状态使用函数(State-using Function)，这类函数会根据当前OpenGL的状态执行一些操作。

OpenGL ES (OpenGL for Embedded Systems) 是 OpenGL 的子集，针对手机、PDA和游戏主机等嵌入式设备而设计。去除了glBegin/glEnd，四边形 (GL_QUADS)、多边形 (GL_POLYGONS) 等复杂图元等许多非绝对必要的特性。

WebGL（全写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完美地解决了现有的Web交互式三维动画的两个问题：第一，它通过HTML脚本本身实现Web交互式三维动画的制作，无需任何浏览器插件支持；第二，它利用底层的图形硬件加速功能进行的图形渲染，是通过统一的、标准的、跨平台的OpenGL接口实现的。

Vulkan是一个跨平台的2D和3D绘图应用程序接口，跟OpenGL一样，Vulkan是Khronos组织制定的“下一代”开放的图形显示API。是与DirectX12能够匹敌的GPU API标准。Vulkan是基于AMD的Mantle API演化而来的。简单理解的是，Vulkan是OpenGL的进化版本，但并不代表OpenGL会被废弃。

DirectX, (Direct eXtension, 简称DX) 是由微软公司创建的多媒体编程接口。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。它们旨在使基于Windows 的计算机成为运行和显示具有丰富多媒体元素 (例如全色图形、视频、3D 动画和丰富音频) 的应用程序的理想平台。DirectX 包括安全和性能更新程序, 以及许多涵盖所有技术的新功能。应用程序可以通过使用DirectX API 来访问这些新功能。

3. `gl.h` `glu.h` `glew.h` 的作用分别是什么?

<GL/gl.h>: OpenGL所使用的函数和常量声明。

<GL/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。

glu是实用库, 包含有43个函数, 函数名的前缀为glu。Glu 为了减轻繁重的编程工作, 封装了OpenGL函数, Glu函数通过调用核心库的函数, 为开发者提供相对简单的用法, 实现一些较为复杂的操作。

GLEW是一个跨平台的C++扩展库, 基于OpenGL图形接口。有了GLEW扩展库, 你就再也不用为找不到函数的接口而烦恼, 因为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,glx,wgl,glx的全部函数。

4. 使用GLFW和freeglut的目的是什么?

GLFW是一个专门针对OpenGL的C语言库, 它提供了一些渲染物体所需的最低限度的接口。它允许用户创建OpenGL上下文, 定义窗口参数以及处理用户输入, 这正是我们需要的。

在我们进行OpenGL编程的时候, 我们需要先建立一个窗口来显示图形, 还需要从鼠标、键盘获取事件。但OpenGL仅仅是针对图形化的一个接口, 它是跨平台的, 至于如何建立窗口和监听IO由平台自带库实现, 像windows可以用windows.h里面的方法建立窗口, 但使用自带库建立窗口一般都特别麻烦。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。

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)

2018 专题 —— Interaction/VR 专题: 主要内容是有关虚拟现实与人机交互的内容。

该专题中, “重定向行走”。他是我们平时玩的VR采用的一种技术。当给人蒙上眼睛情况下, 我们就会对空间失去感知能力。大家一定有过这样的感觉, 就是蒙上眼睛走路, 本以为是走直线, 但其实曲线或者一个圆圈。而在虚拟现实世界, 这种缺陷恰恰变成了很大的利用价值。所以简单讲就是你在真实世界走的路线可能是一个圆圈或者曲线, 而在虚拟现实的世界里你所感知的路线是直线, 这就是重定向行走。

通过将计算机图形绘制系统与人眼视觉过程同步, 可以在虚拟现实中利用这一现象, 特别是实现位置和方向的重定向。

通过这种“动态扫视重定向”, 通过扫视方式隐藏“重定向行走”。这种方法的优势在于它能够向用户隐藏重定向行走; 适用于更广泛的VR场景; 而且是动态形式, 可以引导用户避开静态物体, 甚至是移动的物体。

这种方式采用的基本原理是利用人的一种视觉特性：扫视抑制。扫视抑制简单讲就是我们眼睛在快速扫视的时候视觉敏感性会大大降低的现象，因此对于实物辨识度的正确率也会下降。利用这种现象来隐藏小幅度的转动。然后再借助头显精确的眼球追踪技术，检测并利用这种临时“失明”来隐藏用户场景的轻微旋转。当用户向前行走环视场景时，场景将慢慢地旋转，每次扫视只旋转几度，这样用户为了响应新的视觉线索将条件反射式地改变行走方向，在不知不觉中选择不同的实际物理路径，从而有助于他们避免撞上其他玩家或墙壁等障碍物。