

# 14331047-陈主润-HW1

---

## 2.1 Know OpenGL

---

- **OpenGL**，全称为Open Graphics Library，顾名思义就是一个开放的C++图形库。它定义了一个跨编程语言、跨平台的编程接口规格的专业的图形程序接口，是一个功能强大，调用方便的底层图形库，通俗地讲就是它提供了一套庞大的函数给我们来操作图形和图像。OpenGL是个与硬件无关的软件接口，可以在不同平台像Windows、MacOS、Linux等间轻松移植。它主要有七大功能：建模、变换、颜色模式设置、光照和材质设置、纹理映射、位图显示及图像处理、双缓存动画。
  - **OpenGL ES**，全称是OpenGL for Embedded System，它实际上是OpenGL的一个子集，是针对手机、游戏主机等嵌入式设备而设计的。它去除了OpenGL中四边形、多边形等复杂图元等非绝对必要的特性，以更好地支持嵌入式设备。
  - **WebGL**，全称为Web Graphics Library，显然它是为Web服务的，通过标准的、统一的OpenGL接口实现的一种绘图标准。WebGL通过增加OpenGL ES 2.0的一个Javascript绑定，使得HTML5 Canvas在渲染时，可以借助GPU来渲染3d场景和模型，这样相比于原来单纯地借助浏览器来渲染图形，大大提高了web端的渲染效率。
  - **Vulkan**，它同OpenGL一样，都是由Khronos开发的跨平台的2D和3D绘图API。但是相比OpenGL，它更加接触底层，可以使用通过OpenGL无法访问的GPU硬件特性，且有超高的OS兼容性、渲染特性和硬件效率。目前只有Radeon显卡可以用到这些特性，当然这些显卡也是相对比较昂贵的。
  - **DirectX**，是由微软创建的、由C++实现的多媒体编程接口，它加强3D图形和声音效果，并提供一套硬件驱动标准，使得开发者不用为不同品牌的硬件来写不同的驱动程序（跨硬件）。它让以windows为平台的游戏或多媒体程序获得更高的执行效率。它不仅仅包括图形处理方面的API，还有声音部分、输入部分和网络部分的API。
- 
- **gl.h**，包含了OpenGL所使用的常量和函数的声明。
  - **glu.h**，是OpenGL实用库，是OpenGL标准的一部分，也意味着它就在OpenGL标准库中。它封装了gl.h中的一些函数，来提供更简单的用法，从而实现复杂的操作。
  - **glew.h**，是跨平台的OpenGL扩展库，它是为了解决一些OpenGL 2.0中的高级特性

（因为这些新特性不会添加到只支持OpenGL1.1的gl.h或glu.h中）在某些平台上无法使用的问题而产生的。它包含了gl.h、glu.h、glext.h等中的全部函数。

**GLFW Project:** GLFW是一个开源、多平台的图形库，主要用于创建窗口，渲染OpenGL，管理输入。它本身是用C语言实现的，支持Windows、MacOS和类Unix系统。相比于其它框架，GLFW是很轻量级的，可以花很少的代码就进行窗口管理，同时可以和原生OpenGL语言共用，是目前OpenGL开源框架的首选之一。在之后的作业中，我也打算使用GLFW来辅助实现。

```
1 #include <iostream>
2 #include <GLFW/glfw3.h>
3
4 #include <OpenGL/gl.h>
5
6
7 int main(int argc, const char * argv[]) {
8     // insert code here...
9     if (!glfwInit()) {
10         return -1;
11     }
12
13     glfwWindowHint(GLFW_OPENGL_FORWARD_COMPAT, GL_TRUE);
14     glfwWindowHint(GLFW_OPENGL_PROFILE, GLFW_OPENGL_CORE_PROFILE);
15     glfwWindowHint(GLFW_CONTEXT_VERSION_MAJOR, 3);
16     glfwWindowHint(GLFW_CONTEXT_VERSION_MINOR, 2);
17     GLFWwindow* window = glfwCreateWindow(640, 480, "Hello OpenGL", NULL, NULL);
18     if (!window) {
19         glfwTerminate();
20         return -1;
21     }
22
23     glfwMakeContextCurrent(window);
24     glfwSetKeyCallback(window, key_callback);
25 }
```

## 2.2 Know the State of Art

所选course: *Sketch-based modeling*(Frederic Cordier、Karan Singh、Yotam Gingold、Yotam Gingold)

该course主要介绍了一种3D建模方法，这种方法根据用户所画的素描（sketch）来自动生成对应的3D模型，使得3D建模变得和画画一样简单，这就是**基于素描的建模方法 (sketch-based modeling)**。

该course主要讲了以下内容：

- 素描画的获取和处理
- 对3D曲线和曲面的多视图的基于素描的建模
- 使用先导知识进行基于素描的建模
- 对3D曲线和曲面的单视图的基于素描的建模
- 基于素描的编辑

里面涉及到许多有趣的方法。在素描画的处理时，涉及到曲线的平滑化，文中就介绍到了一个叫sketch neatening的方法。通过输入一条不平滑的曲线和French曲线，将它们分段，以French曲线为模板来使得原曲线平滑。里面还讲到了轮廓识别（即根据输入判断是圆还是正方形还是其他）、轮廓抽象、通过膨胀的方法来进行基于素描的建模，还有利用先导知识和常识来对生活中常见的物体建模，包括动物、树、衣服等（常识和知识可以使得建模结果更加准确、更加高质量）。此外，文中还设计到一些螺旋线重构、素描工具以及对物体的分割识别等技术。

---

这个课程里介绍的就是CG的四个主要内容建模、渲染、动画、人机交互中的建模的一种建模方法--基于素描的建模方法。该方法中涉及到许多CG方面的技术，包括图形曲线的平滑、轮廓识别、角的识别、物体表面优化、NPR、图形分割与识别、曲线配对、CrossShade描影法，还有通过对2D图形采用膨胀法来得到3D物体。其实最主要的，还是对曲线的各种处理，包括获取、平滑、识别、重构等。