**深 圳 大 学 实 验 报 告**

**课程名称：­ 计算机图形学**

**实验项目名称： 实验一 OpenGL基本绘制**

**学院： 计算机与软件学院**

**专业： 计算机科学与技术**

**指导教师： 徐鹏飞**

**报告人：郑雨婷 学号：2021150122 班级： 高性能**

**实验时间：2023年 9月10日 -- 2023年 10月18日**

**实验报告提交时间： 2023年 10月18日**

**教务部制**

|  |
| --- |
| 实验目的与要求：   1. 掌握Visual Studio Community 2019集成开发环境的安装；掌握CMake跨平台构建工具的安装；掌握Git版本控制工具的安装；掌握vcpkg库管理工具的安装；掌握系统环境变量的设置；了解和掌握OpenGL的环境配置；掌握OpenGL工程项目的建立和基本设置。 2. 理解OpenGL的原理；了解和熟悉OpenGL着色语言；掌握基于OpenGL的C++程序结构；掌握OpenGL中若干基本二维图形的绘制；了解顶点着色器的使用；了解片元着色器的使用。 3. 使用现代OpenGL中的着色器，绘制多个简单的二维图形，形状内容不限，自己发挥。 |
| 实验过程及内容：   1. OpenGL的环境配置   因为之前已经安装过Visual Studio Community和Git，实验过程中只需要安装CMakehe和vcpkg。从实验1.1中提供的安装包下载安装即可。在安装完成之后，需要配置环境变量，    之后，进行OpenGL库（GLFW，GLAD，GLM）的安装，将实验提供的VS2019软件包中的downloads压缩包解压到vcpkg文件夹中。之后打开命令行，在命令行中输入命令 vcpkg install glfw3 glad glm，等待glfw、glad和glm完成编译即可。最后，解压实验1.1代码并进入到实验1.1\_参考代码文件夹，在当前文件夹下打开命令行，执行 cmake -B .，执行后会在当前项目文件夹内生成main.sln文件，点击打开。在出现的VS界面中，可以看到“解决方案管理器”里面右键点击 “main”项目，将其设置为启动项。点击菜单栏下方的“本地Windows调试器”编译运行程序，看到一个红色三角形，表示环境配置成功。     1. 绘制二维图形   实验一留空代码中比实验1.2新增了函数generateEllipsePoints，用来绘制圆和椭圆。分析代码里面的参数，知道该函数会根据verticalScale是否等于1去绘制不同的图形，当verticalScale=1的时候，就会绘制出圆，否则会绘制出椭圆来。再仔细看函数，就能理解这是两个半径的比值了。  经过实验1.2，我们知道，生成一个形状，首先需要在init()中定义点和颜色数组，然后调用函数计算点的位置信息和颜色信息并且存入数组。代码中已经有了其他形状的绘制，参照其绘制圆和椭圆即可。找到@TODO:定义一个圆和椭圆的点和颜色的数组，再往上去看一下常量，有CIRCLE\_NUM\_POINTS和ELLIPSE\_NUM\_POINTS都是100，所以一个圆和椭圆会绘制100个点。    之后就是调用生成圆和椭圆顶点位置的函数，依照参数名字的大概意思，vertices是传圆的点的位置的数组，colors是圆的颜色的数组，startVertexIndex开始绘制的下标，一般都是从0开始的，numPoints是绘制的点的数量，center是图形的中心点的位置，scale圆的大小，verticalScale是绘制的图形的半径的比值。那参入对应的参数之后，并调用函数，初始化绘制圆和椭圆的两个数组。    再根据之前的例子，我们知道还需要初始化圆和椭圆的数据。我们要对vao数组开更大的空间，三角形、圆形、椭圆、矩形共四个形状，因此需要将vao数组大小设为4，对顶点缓存对象进行管理。然后复制前面初始化三角形的代码，只需要改一下vao的下标，把点和颜色的数组改为对应的圆和椭圆的数组。    这些都做完之后，就可以在display函数中，调用对应的接口进行绘制图形了。参照三角形的写法，先写glBindVertexArray函数绑定圆和椭圆的顶点数组对象，再调用glDrawArrays对圆和椭圆进行绘制。    最终运行结果如下：     1. 不同的图形颜色效果   自己再根据已有的几何形状，或者自己设定的几何形状绘制一张和查看图片不同的图片。我们现在已有了正方形、圆形、椭圆形、三角形。用这四种形状可以绘制出一个只小狗。  首先把背景颜色改为蓝色，这个可以在init函数中的glClearColor函数修改，把参数全改为(0.50, 1.0, 1.0, 1.0)。  接着，是绘制一个黄色的矩形作为脸部。先是要定义一个vec2的位置信息的数组，还有定义vec3颜色信息的数组，并用这两个数组传入generateSquarePoints函数，在其中初始化，generateSquarePoints函数其中修改，全部统一为传入的颜色为黄色，并选择好合适的中心坐标还有大小尺寸。注意还需要在display函数中对图形进行绘制。    再创建四个数组，用于存储耳朵的颜色和位置。同理，设置好合适的大小还有位置，传入generateEllipsePoints对各个数组进行赋值。为了设置耳朵的颜色为黄色，需要改动generateEllipsePoints内部，若verticalScale=1，颜色设为黄色。还在init函数中，对以上的四个数组进行初始化、并和一个vao对象绑定。最后在display函数中进行绘制出来。结果如下图所示：    再创建四个数组，用于存储眼睛的颜色和位置。同理，设置眼睛的颜色为黑色，并设置好合适的大小还有位置，传入generateEllipsePoints对各个数组进行赋值。verticalScale不等于1，是椭圆，颜色设为黑色。init函数中，对以上的四个数组进行初始化、并和一个vao对象绑定。最后在display函数中进行绘制出来。结果如下图所示：    最后，利用之前的绘制三角形的几个函数还有遗留下来的数组，对generateTrianglePoints函数进行修改，把各个顶点的颜色都修改为黑色，调整大小和位置。 |

深圳大学学生实验报告用纸

|  |
| --- |
| 实验结论：  本次实验的主要目的是为了掌握OpenGL开发环境的配置和基本使用，理解OpenGL的原理，以及熟悉OpenGL着色语言。在本次实验中，我们成功安装了Visual Studio Community 2019、CMake、Git、以及vcpkg库管理工具。通过这些工具的安装和配置，我们建立了一个适合OpenGL开发的环境。之后，我们深入了解了OpenGL的原理，熟悉了OpenGL着色语言，掌握了基于OpenGL的C++程序结构。  具体地，在实验中，我们了解了顶点着色器和片元着色器的使用。这两种着色器是构建OpenGL应用程序的关键组成部分，它们用于控制顶点和像素的渲染过程。并且成功使用现代OpenGL的着色器，绘制了多个简单的二维图形，包括正方形、三角形、圆形和椭圆。这些图形的绘制不仅锻炼了我们对OpenGL的使用技能，还展示了图形学的基础知识。  总的来说，通过本次实验，我们不仅掌握了OpenGL的基本知识和开发环境配置，还锻炼了创造力和编程技能。这将为我们更深入地研究和应用图形学领域提供坚实的基础。在实验的最后，我们成功生成了两幅图像，一幅是参考图片的样式，另一幅是我们自己创作的独特图形，进一步展示了我们的学习成果。 |
| 指导教师批阅意见：  成绩评定：  指导教师签字：  年 月 日 |
| 备注： |

注：1、报告内的项目或内容设置，可根据实际情况加以调整和补充。

2、教师批改学生实验报告时间应在学生提交实验报告时间后10日内。