Assignment 1: 环境搭建与使用

Computer Graphics Teaching Stuff, Sun Yat-Sen University

- Due Date: 9月27号晚上12点之前，提交到zhangzk3@mail2.sysu.edu.cn邮箱。

> 第一次作业是相关的环境搭建和使用，为了方便同学们专注于算法实现而不用拘泥于各种渲染API接口的调用，我们统一在一个linux虚拟机上搭建环境，该linux虚拟机镜像已经装好了一些库和IDE，后续的作业将给出相关的代码框架，同学们只需填补其中的函数即可。注意：你可以跟同学、助教或老师讨论，或者在网上查阅资料，但必须独立完成自己的作业，严禁直接抄袭！

1、虚拟机的使用

为了免去配置作业所需环境的麻烦，本次课程使用虚拟机，学生在虚拟机内编写，编译和运行代码。我们提供的文件为虚拟硬盘文件，使用虚拟机挂载该文件后，就可以保证所有人的环境是统一并且完善的，不需要再手动配置环境。在安装完虚拟机后，我们需要手动安装 Guest Additions 来增强虚拟机的功能。

1.1、安装虚拟机

这里我们使用 Oracle VM VirtualBox 虚拟机。

如果你使用 Windows 系统，你可以直接下载https://download.virtualbox.org/virtualbox/6.1.4/VirtualBox-6.1.4-136177-Win.exe，下载完成后按照指示完成安装。

如果你使用 Mac OS 系统，你可以直接下载https://download.virtualbox.org/virtualbox/6.1.4/VirtualBox-6.1.4-136177-OSX.dmg，下载完成后按照指示完成安装。

如果你使用 Linux 内核的系统，你可以查看https://www.virtualbox.org/wiki/Linux_Downloads，找到你使用的系统，按照对应的指示完成安装。

1.2、下载虚拟镜像文件

虚拟硬盘文件的下载地址为 https://cloud.tsinghua.edu.cn/f/103133da1bf8451b8ba6，密码为games101。下载完成后得到 GAMES101_Ubuntu 18.04.2 (64bit).rar，将其解压后得到虚拟硬盘文件GAMES101_Ubuntu 18.04.2 (64bit).vdi。

1.3、配置虚拟机

打开**Virtual Box**，点击新建，设置类型为**Linux**，版本为**Ubuntu-64 bit**，建议设置虚拟机的内存大小为2GB，然后选择使用已有的虚拟硬盘文件，设置为之前解压得到的GAMES101_Ubuntu 18.04.2 (64bit).vdi文件，最后点击创建就完成了虚拟机的配置工作。

之后就可以使用创建好的虚拟机了，选中刚刚创建好的虚拟机，点击右侧上方的启动按钮就可以启动虚拟机，**Unbuntu**系统的密码为**Ilovegraphics**（注意，首字符I是大写）。

1.4、安装Guest Additions

进入系统后，点击上方菜单的设备，点击**安装增强功能**，如下图所示。安装完成后，重启虚拟机系统就完成了Guest Additions的安装。若菜单栏未显示，可能是设置了缩放模式，可尝试Ctrl+C解决（必须是键盘右侧的Ctrl键）。

如果上面的方法安装失败了，可以使用按键组合`ctrl+alt+t`调出linux终端，使用如下命令安装Guest Additions功能。

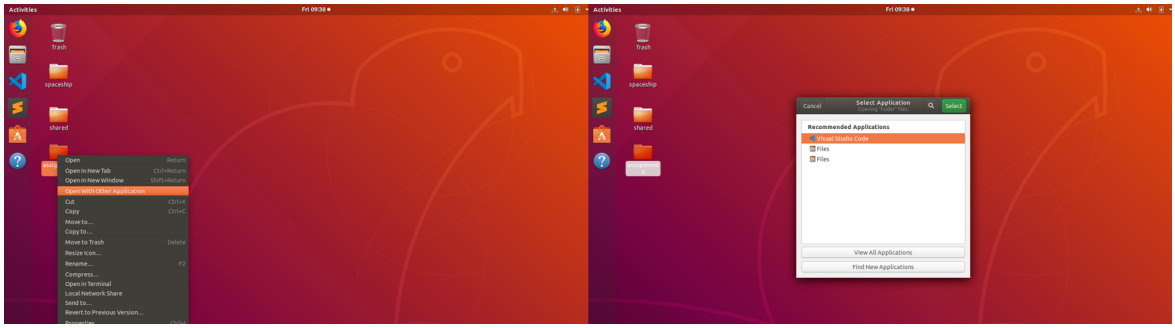
```
sudo mkdir -p /media/cdrom
sudo mount -t auto /dev/cdrom /media/cdrom/
cd /media/cdrom/
sudo sh VBoxLinuxAdditions.run
```

执行完毕之后，重启虚拟机系统就完成了Guest Additions的安装。

1.5、作业框架的传输及编辑

作业框架的导入和导出有很多方式，这里只提一种。当你在你的个人电脑上下载好作业框架后，直接将其拖进虚拟机系统里。这里需要开启Virtual Box的拖放功能：进入虚拟机系统后，点击上面菜单的**设备**，将拖放功能设置为**双向**即可。

导入作业框架后，可以使用Visual Studio Code来查看和编辑。右键作业框架的文件夹，选择使用其他应用（Open with other Application）来打开，选择Visual Studio Code，具体如下图所示



2、作业框架说明

本部分将体现在样例程序main.cpp中。**本次作业的要求比较简单，主要是让同学们搭建好环境，并稍微熟悉一下Eigen库的使用。**

2.1、开发工具说明

虚拟机系统中已经自带Visual Studio Code与Sublime作为文本编辑器，本课程推荐使用Visual Studio Code，并在linux系统的终端命令行中编译、运行程序。

将框架代码拷贝到虚拟机中，打开VSCode，选择File->Open Folder找到代码所在的文件夹，选择打开即可。

2.2、库文件的使用

由于图形学涉及到矩阵和向量的运算，因此这里使用Eigen库作为线性代数运算库。除此之外，还会用到一些其他简单的数学运算，这里使用标准库的cmath。后续的框架代码还涉及到一些其他库，但不必担心，这些库都已经安装好在系统中了。

```
#include<cmath>
#include<eigen3/Eigen/Core>
#include<eigen3/Eigen/Dense>
#include<iostream>
```

Eigen库是本课程使用的线性代数运算库，官方文档为http://eigen.tuxfamily.org。

2.3、向量、矩阵

关于本部分内容，请详细阅读官方文档的矩阵部分https://eigen.tuxfamily.org/dox/group__TutorialMatrixArithmetic.html以获得更全面、清晰的理解。

```
    // Example of vector
    std::cout << "Example of vector \n";
    // vector definition
    Eigen::Vector3f v(1.0f,2.0f,3.0f);
    Eigen::Vector3f w(1.0f,0.0f,0.0f);
    // vector output
    std::cout << "Example of output \n";
    std::cout << v << std::endl;
    // vector add
    std::cout << "Example of add \n";
    std::cout << v + w << std::endl;
    // vector scalar multiply
    std::cout << "Example of scalar multiply \n";
    std::cout << v * 3.0f << std::endl;
    std::cout << 2.0f * v << std::endl;
```

上面关于Vector的使用样例展示了如何定义一个三维浮点向量并且执行输出、加减、数乘运算。**请自行根据数乘的形式与向量点积的形式探索点积的用法。**

```
    // Example of matrix
    std::cout << "Example of matrix \n";
    // matrix definition
    Eigen::Matrix3f i,j;
    i << 1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0, 9.0;
    j << 2.0, 3.0, 1.0, 4.0, 6.0, 5.0, 9.0, 7.0, 8.0;
    // matrix output
    std::cout << "Example of output \n";
    std::cout << i << std::endl;
    // matrix add i + j
    // matrix scalar multiply i * 2.0
    // matrix multiply i * j
    // matrix multiply vector i * v
```

上述关于Matrix的使用样例展示了如何定义一个三维浮点矩阵进行输出，**请自行根据注释与Vector部分的经验探索矩阵的加减、数乘、矩阵乘法、矩阵乘向量的用法。**

3、作业描述与提交

3.1、程序编译与运行

为了方便之后的作业编写，本次作业要求使用cmake进行编译。

在main.cpp所在的目录下，打开终端（命令行），一次输入下面的命令：

- 创建名为build的文件夹：`mkdir build`
- 将当前终端所在的路径移动到build文件夹下：`cd build`
- 使用cmake编译程序，其中`..`表示当前路径的上一级目录：`cmake ..`
- 然后使用make编译程序，编译过程的信息将会显示在终端：`make`
- 若上一步无错误，则可以运行程序（这里的Transformation是可执行文件名，可参照CMakeList.txt中修改）：`./Transformation`

3.2、作业提交

将PDF报告文件和代码打包提交到zhangzk3@mail2.sysu.edu.cn邮箱，邮件名和pdf报告文件格式为hw1_姓名_学号。

3.3、作业描述

本次作业要求同学们完成的工作如下所示：

1. 按照前面的教程搭建好虚拟机，并编译运行框架代码，在报告中贴出程序运行的结果。（60分）

2. 在框架代码的基础上，了解Eigen库的向量的使用，并在代码中实现`v`和`w`向量点乘并输出结果，在报告中贴上结果截图。（20分）

```
    // Example of vector
    std::cout << "Example of vector \n";
    // vector definition
    Eigen::Vector3f v(1.0f,2.0f,3.0f);
    Eigen::Vector3f w(1.0f,0.0f,0.0f);
    // vector output
    std::cout << "Example of output \n";
    std::cout << v << std::endl;
    // vector add
    std::cout << "Example of add \n";
    std::cout << v + w << std::endl;
    // vector scalar multiply
    std::cout << "Example of scalar multiply \n";
    std::cout << v * 3.0f << std::endl;
    std::cout << 2.0f * v << std::endl;

    // job1: 实现v和w的向量点积并输出结果
    {

    }
```

3. 在框架代码的基础上，了解Eigen库的矩阵的使用，并在代码中实现`i`与`j`的矩阵相加、`i`与2.0的数乘、`i`与`j`的矩阵相乘、`i`与`v`的矩阵乘向量，并输出相应的结果，在报告中贴上结果截图。（20分）

```
    // Example of matrix
    std::cout << "Example of matrix \n";
    // matrix definition
    Eigen::Matrix3f i,j;
    i << 1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0, 9.0;
    j << 2.0, 3.0, 1.0, 4.0, 6.0, 5.0, 9.0, 7.0, 8.0;
    // matrix output
    std::cout << "Example of output \n";
    std::cout << i << std::endl;
    // matrix add i + j
    // matrix scalar multiply i * 2.0
    // matrix multiply i * j
    // matrix multiply vector i * v
```

```
    // job2：实现i与j的矩阵相加、i与2.0的数乘、i与j的矩阵相乘、i与v的矩阵乘向量，并输出相应的结果
    {

    }
```