

# 计算机图形学

## 环境搭建与使用

### Assignment 1

姓名：陈家豪

班级：2018 级计科 1 班

学号：18308013

# 目录

1	作业任务与要求	2
2	实验过程与结果	2
2.1	虚拟机的搭建与使用 . . . . .	2
2.2	框架代码的运行 . . . . .	3
2.3	Eigen 库向量的使用 . . . . .	4
2.4	Eigen 库矩阵的使用 . . . . .	5
3	实验总结与感想	6

# 1 作业任务与要求

- (1) 搭建虚拟机，编译运行框架代码；
- (2) 了解 Eigen 库的向量的使用，在代码中实现  $v$  和  $w$  向量点乘并输出结果；
- (3) 了解 Eigen 库的矩阵的使用，在代码中实现  $i$  与  $j$  的矩阵相加、 $i$  与 2.0 的数乘、 $i$  与  $j$  的矩阵相乘和  $i$  与  $v$  的矩阵乘向量，并输出结果；

## 2 实验过程与结果

### 2.1 虚拟机的搭建与使用

本次实验需要我们下载虚拟机软件 Oracle VM VirtualBox，并使用课程提供的虚拟机进行编写、编译和运行代码。

软件和虚拟机文件的下载和安装不再赘述。我们将重心放在虚拟机的配置和环境的搭建上。首先我们打开 VirtualBox，点击“新建”按钮，在“新建虚拟电脑”界面中填入虚拟机的名称和保存位置后，类型选择“Linux”，版本选择“Ubuntu(64-bit)”，然后点击下一步。

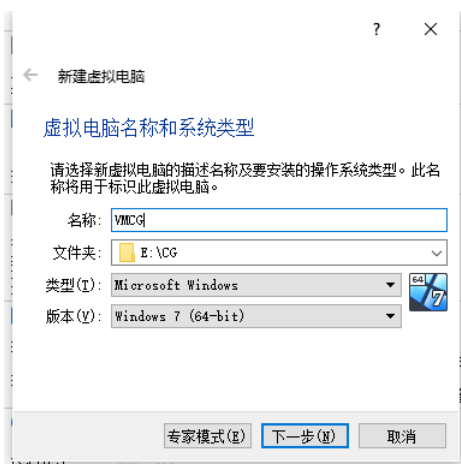


图 1: 搭建虚拟机 1

内存分配中，为了虚拟机运行更流畅，我选择设置内存大小为 4G 而不是 2G。

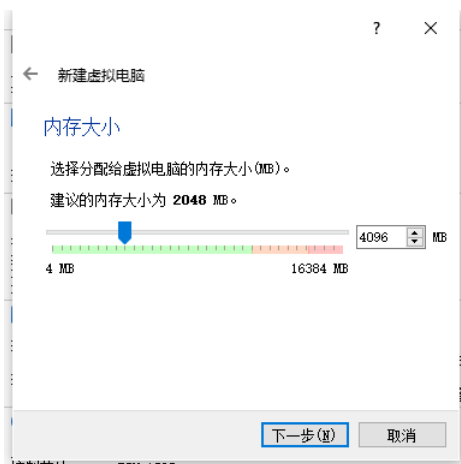


图 2: 搭建虚拟机 2

最后一步，在虚拟硬盘选项中，我们选择“使用已有的虚拟硬盘文件”选项，挂载课程提供的虚拟机文件。点击创建，即可生成新的虚拟机。

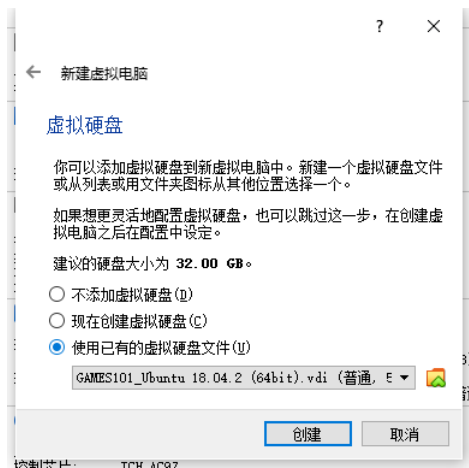


图 3: 搭建虚拟机 3

进入虚拟机，点击运行界面上方菜单的“设备”选项，然后选择“安装增强功能”，安装 Guest Additions。安装完成后，重启系统即可完成安装。然后点击上方菜单的“设备”选项，选择“拖放”中的“双向”，即可完成文件从虚拟机的导入和导出。

以上就是虚拟机的创建和配置过程，最终效果如图4所示。

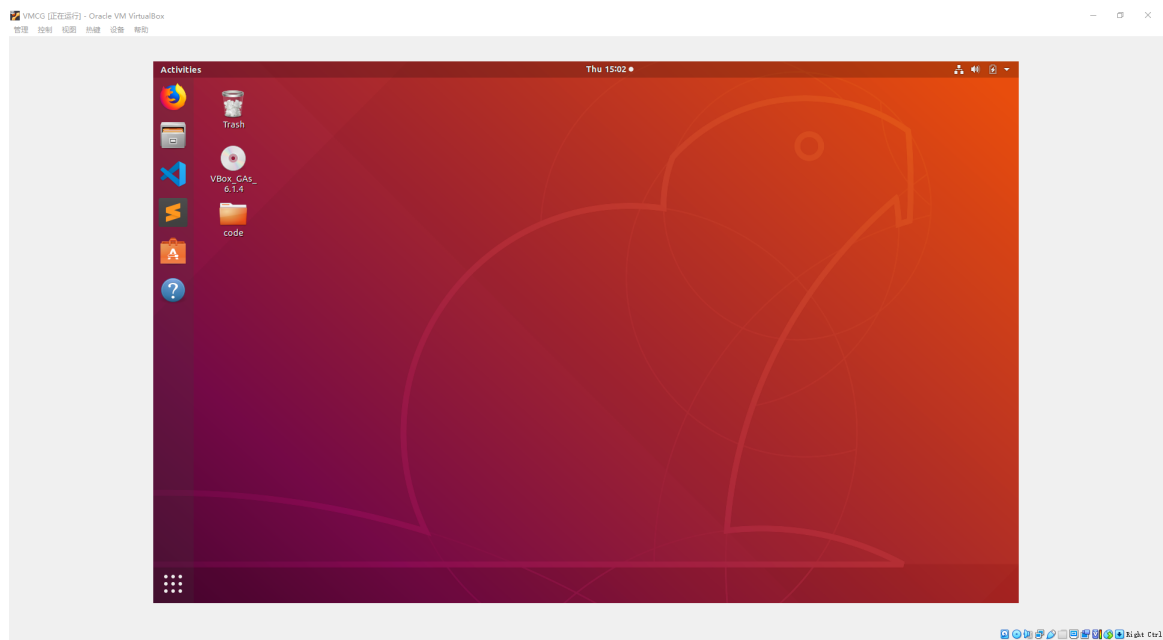


图 4: 虚拟机界面

## 2.2 框架代码的运行

从 Q 群下载本次实验的作业框架后，直接将文件拖放到虚拟机界面，即可完成文件的导入。我们首先尝试框架代码是否可以运行。

本次实验要求我们使用 `cmake` 进行编译。打开代码所在的目录终端，输入以下命令：

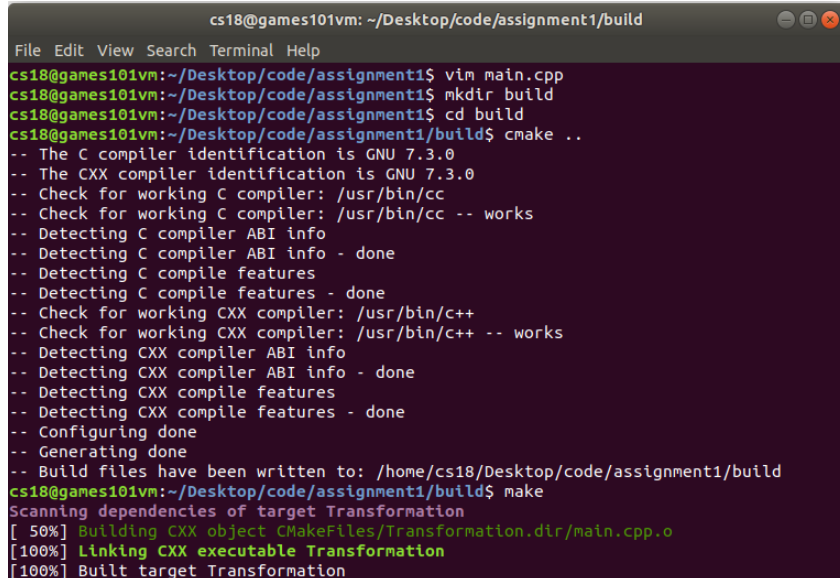
- (1) 新建名为 `build` 的文件夹：`mkdir build`

(2) 当前终端路径移动到 build 文件夹: `cd build`

(3) 使用 cmake 编译程序: `cmake ..`

(4) 使用 make 编译程序: `make`

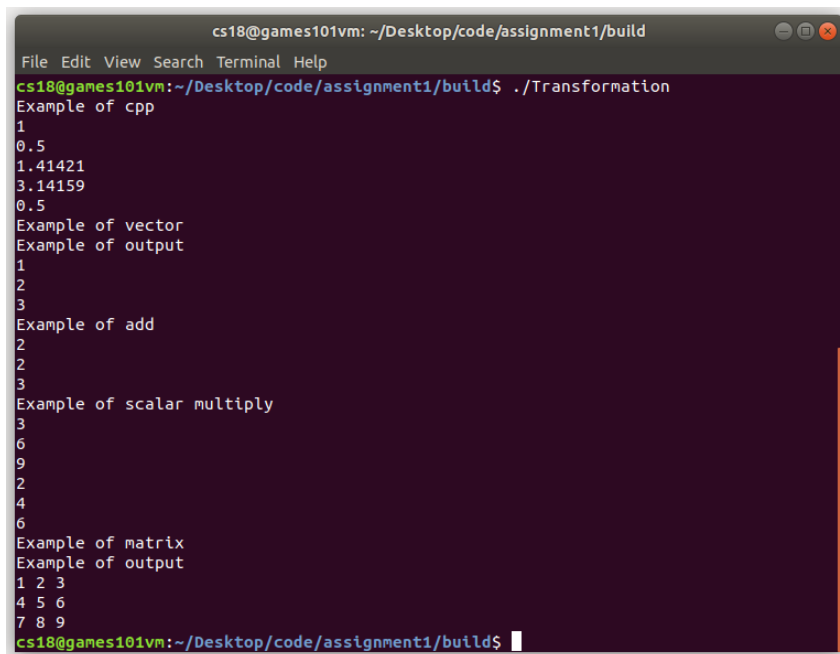
即可完成程序的编译。编译过程如图5所示。



```
cs18@games101vm: ~/Desktop/code/assignment1/build
File Edit View Search Terminal Help
cs18@games101vm:~/Desktop/code/assignment1$ vim main.cpp
cs18@games101vm:~/Desktop/code/assignment1$ mkdir build
cs18@games101vm:~/Desktop/code/assignment1$ cd build
cs18@games101vm:~/Desktop/code/assignment1/build$ cmake ..
-- The C compiler identification is GNU 7.3.0
-- The CXX compiler identification is GNU 7.3.0
-- Check for working C compiler: /usr/bin/cc
-- Check for working C compiler: /usr/bin/cc -- works
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Detecting C compile features
-- Detecting C compile features - done
-- Check for working CXX compiler: /usr/bin/c++
-- Check for working CXX compiler: /usr/bin/c++ -- works
-- Detecting CXX compiler ABI info
-- Detecting CXX compiler ABI info - done
-- Detecting CXX compile features
-- Detecting CXX compile features - done
-- Configuring done
-- Generating done
-- Build files have been written to: /home/cs18/Desktop/code/assignment1/build
cs18@games101vm:~/Desktop/code/assignment1/build$ make
Scanning dependencies of target Transformation
[ 50%] Building CXX object CMakeFiles/Transformation.dir/main.cpp.o
[100%] Linking CXX executable Transformation
[100%] Built target Transformation
```

图 5: 作业框架编译过程

输入 “./Transformation” 即可运行编译好的程序 Transformation，如图6所示。



```
cs18@games101vm: ~/Desktop/code/assignment1/build
File Edit View Search Terminal Help
cs18@games101vm:~/Desktop/code/assignment1/build$ ./Transformation
Example of cpp
1
0.5
1.41421
3.14159
0.5
Example of vector
Example of output
1
2
3
Example of add
2
2
3
Example of scalar multiply
3
6
9
2
4
6
Example of matrix
Example of output
1 2 3
4 5 6
7 8 9
cs18@games101vm:~/Desktop/code/assignment1/build$
```

图 6: 作业框架的运行

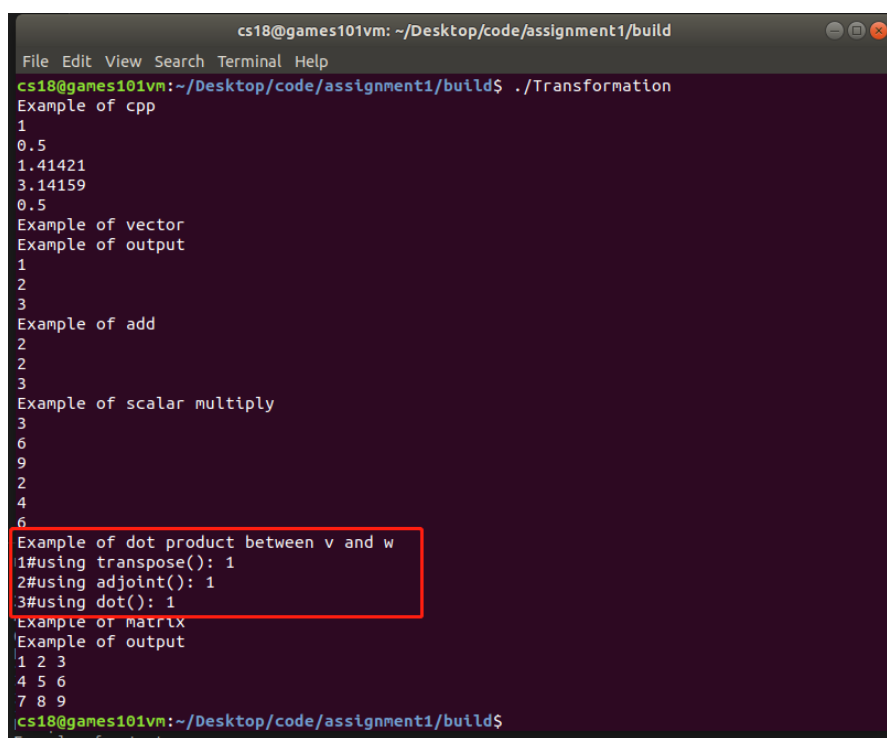
## 2.3 Eigen 库向量的使用

该部分作业要求我们熟悉 Eigen 库向量的使用，并实现向量  $v$  和向量  $w$  的点乘。

根据官方文档，向量的点乘可以使用两种方法：一是用 `dot()` 实现；二是先使用 `adjoint()` 转换向量 `v`，再与 `w` 相乘。除此以外，我们还有第三种方法，就是使用 `transpose()` 将向量 `v` 转置，再与 `w` 相乘。代码如下所示：

```
1 // job1: 实现v和w的向量点积并输出结果
2 {
3     std::cout << "Example of dot product between v and w\n";
4     std::cout << "1#using transpose(): ";
5     std::cout << v.transpose()*w << std::endl; // 方式1
6     std::cout << "2#using adjoint(): ";
7     std::cout << v.adjoint()*w << std::endl; // 方式2
8     std::cout << "3#using dot(): ";
9     std::cout << v.dot(w) << std::endl; // 方式3
10 }
```

最终编译结果如图7所示。可见，三种实现方式的结果均为 1，与实际情况相符。



```
cs18@games101vm: ~/Desktop/code/assignment1/build
File Edit View Search Terminal Help
cs18@games101vm:~/Desktop/code/assignment1/build$ ./Transformation
Example of cpp
1
0.5
1.41421
3.14159
0.5
Example of vector
Example of output
1
2
3
Example of add
2
2
3
Example of scalar multiply
3
6
9
2
4
6
Example of dot product between v and w
1#using transpose(): 1
2#using adjoint(): 1
3#using dot(): 1
Example of matrix
Example of output
1 2 3
4 5 6
7 8 9
cs18@games101vm:~/Desktop/code/assignment1/build$
```

图 7: 向量点乘的结果

## 2.4 Eigen 库矩阵的使用

该部分作业要求我们熟悉 Eigen 库矩阵的使用，并在代码中实现矩阵相加、数乘、矩阵相乘和矩阵与向量的相乘。根据官方文档，最终实现的代码如下所示：

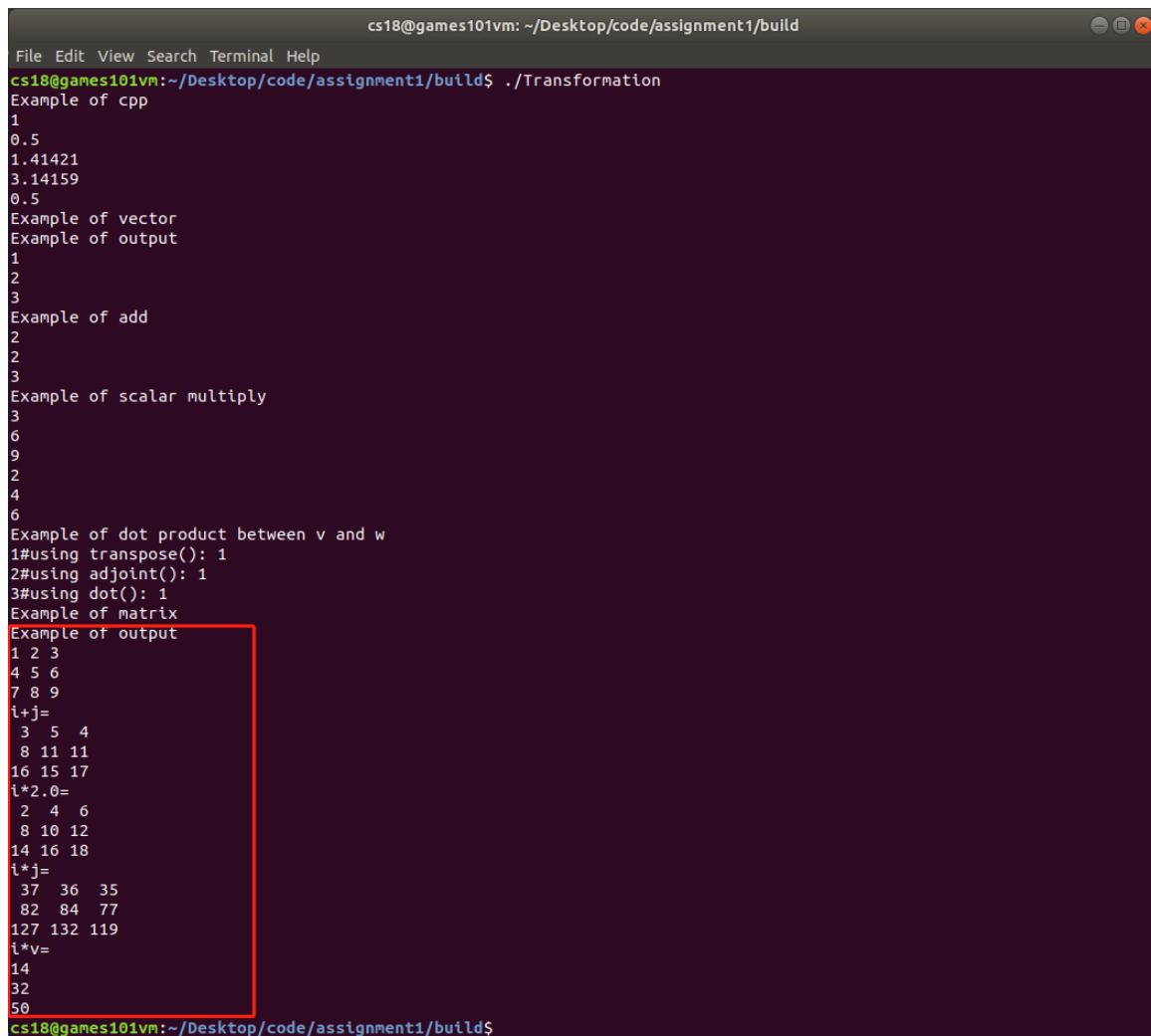
```
1 // job2: 实现i与j的矩阵相加、i与2.0的数乘、i与j的矩阵相乘、i与v的矩阵乘向量，并输出相应的
   结果
2 {
3     cout << "i+j=\n" << i+j << endl; // 矩阵相加
4     cout << "i*2.0=\n" << i*2.0 << endl; // 数乘
5     cout << "i*j=\n" << i*j << endl; // 矩阵相乘
```

```

6   cout << "i*v=\n" << i*v << endl; // 矩阵乘向量
7   }

```

最终编译结果如图8所示。可见，矩阵的计算结果均正确，没有异常。



```

cs18@games101vm: ~/Desktop/code/assignment1/build
File Edit View Search Terminal Help
cs18@games101vm:~/Desktop/code/assignment1/build$ ./Transformation
Example of cpp
1
0.5
1.41421
3.14159
0.5
Example of vector
Example of output
1
2
3
Example of add
2
2
3
Example of scalar multiply
3
6
9
2
4
6
Example of dot product between v and w
1#using transpose(): 1
2#using adjoint(): 1
3#using dot(): 1
Example of matrix
Example of output
1 2 3
4 5 6
7 8 9
i+j=
3 5 4
8 11 11
16 15 17
i*2.0=
2 4 6
8 10 12
14 16 18
i*j=
37 36 35
82 84 77
127 132 119
i*v=
14
32
50
cs18@games101vm:~/Desktop/code/assignment1/build$

```

图 8: 矩阵计算的结果

### 3 实验总结与感想

本次作业是计算机图形学课程的第一次作业，要求我们搭建好本次课程需要的编译环境，以及掌握 Eigen 库的使用。与其他实验课程不同的是，本课程提供了配置好的虚拟机环境以及作业框架，我们可以集中精力完成算法的实现，让我十分感动。本次作业较为简单，实现难度不高，在浏览官方文档以及完成作业任务后，我对 Eigen 库有了较为基础的了解，也学会了向量和矩阵的有关计算。

在此感谢助教和老师在实验过程中提供的帮助，希望我能顺利完成下一次作业。