COSC 471: Computer Graphics

Spring 2023, Assignment 1

VM, VSC, Eigen3, and HelloWorld

(Total points: 100)

Due date: Feb 26th, 2023 (end of the day)

Submission Instructions

1. Compile your solutions and generate screenshots of your output, then write your understanding of output in a file. Compress source codes output and your explanation into one compressed file.
2. Rename the compressed file following this notation:
   FirstnameLastnamePA1.zip (or any format you prefer)
   Do not use a space in the filename.
3. Upload and submit the compressed file through Blackboard.
4. **NO** late submission is accepted.

---

1. Usage of VM

To complete the programming assignment in COSC 471 course, students are required to install and use virtual machine (VM).

1.1 Installing virtual machine

You can choose any VM you prefer or follow CIS department suggestions. Please search in department website and look for student resources about VMware.

If you would like to use Oracle VM VirtualBox, here is some information:

- For windows users, you can download VirtualBox here:
  `https://download.virtualbo x.org/virtualbox/6.1.4/VirtualBox-6.1.4-136177-Win.exe`
- For Mac OS users, you can download from here:
  `https://download.virtualbo x.org/virtualbox/6.1.4/VirtualBox-6.1.4-136177-OSX.dmg`
- For Linux kernel OS users, please check the link below and find the corresponding downloads for your OS:
  `https://www.virtualbox.org/wiki/Linux_Downloads`

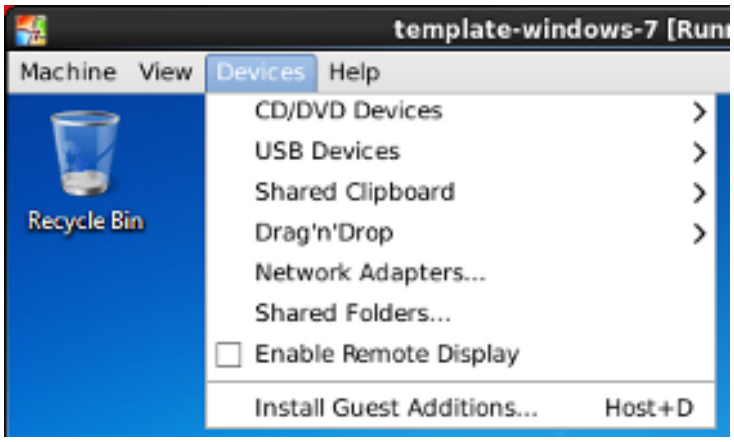
After downloading, please install it as directed.

1.2 Download virtual OS HD

After VM software is ready, please download a virtual Ubuntu. You should be able to find Ubuntu on CIS department resources for students. We need version at least Ubuntu 18.04.0 and 64 bit is preferred. It is suggested you set the VM's memory to be more than 2GB.

1.3 Install Guest Additions

After getting into the Ubuntu system, click "Devices" and then click "install Guest Additions"



You can also refer to link below for more details:

https://docs.oracle.com/cd/E36500_01/E36502/html/qs-guest-additions.html

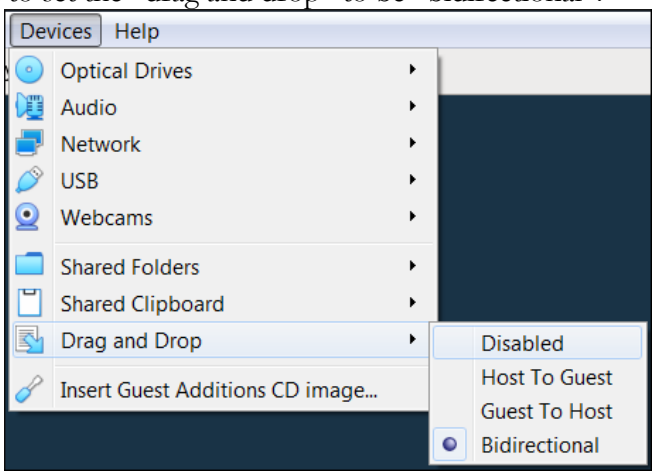
If the installing above is not successful, please open terminal using ctrl+alt+t, and then install using the following commands:

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

After installing is completed, please **restart** the virtual machine.

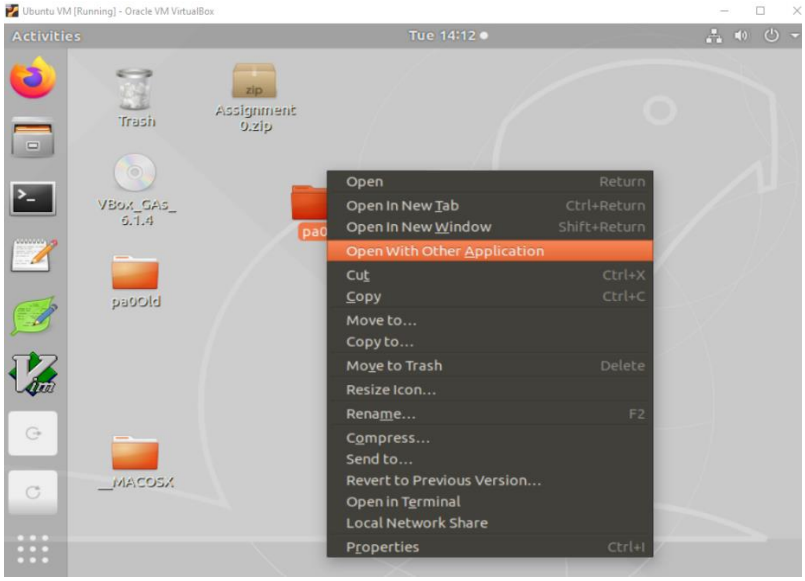
1.4 Assignment editing and importing/exporting

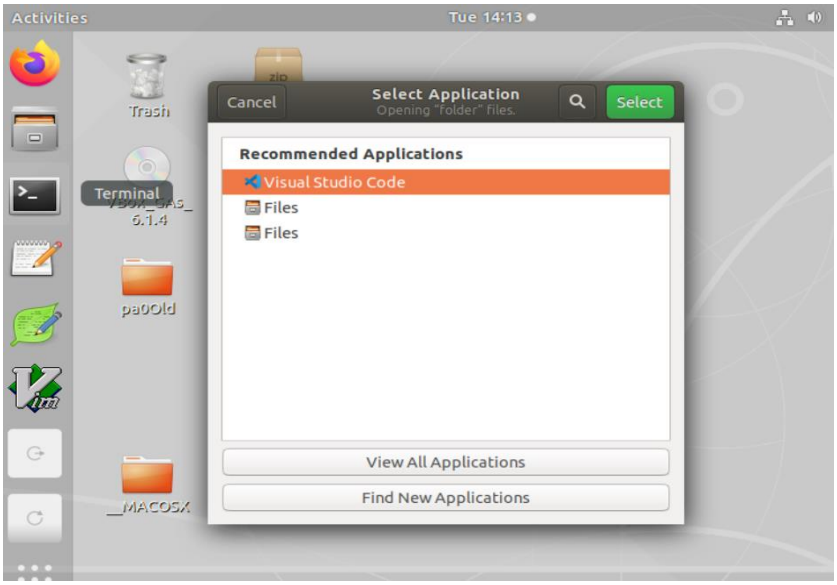
Assignment 1 skeleton is named as pa1, a compressed file. Download your assignment skeleton through Blackboard to your hosting machine. Then drag it to your virtual machine. Here, you may need to set the "drag and drop" to be "bidirectional".



You can also refer to the following link for more details:
https://docs.oracle.com/en/virtualization/virtualbox/6.1/user/guestadd-dnd.html

After you import the assignment to virtual machine, you can use Visual Studio Code to view and edit the assignment. Right click the assignment folder and then "open with other Application". Then choose "Visual Studio Code".





If you don't have VSC installed, please download and install it. You may need to check CIS department resource for students.

2. Assignment skeleton and programming environment

2.1 Vcpkg and Eigen3

You may need to download and install Vcpkg to help you manage C and C++ library on VSC. Please refer to the links below:
https://github.com/microsoft/vcpkg
https://docs.microsoft.com/en-us/cpp/build/vcpkg?view=vs-2019

Eigen is a library which help you manage Linear Algebra related works. You can find Eigen library information here:
http://eigen.tuxfamily.org/index.php?title=Main_Page
Specifically, you can find Vector and Matrix related topics here:
https://eigen.tuxfamily.org/dox/groupTutorialMatrixArithmetic.html

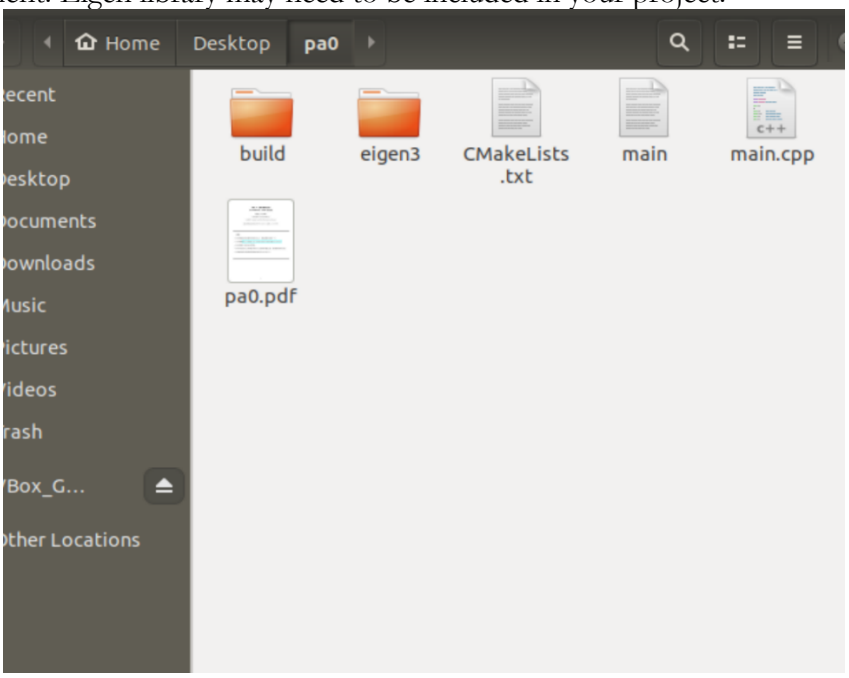
```
// Example of  vector
std::cout << "Example of vector \n";
//    vector    definition
Eigen::Vector3f v(1.0f,2.0f,3.0f);
Eigen::Vector3f w(1.0f,0.0f,0.0f);
// vector output
std::cout << "Example of output \n"; std::cout << v << std::endl;
// vector add
std::cout << "Example of add \n"; std::cout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

Example 1 above shows how to define a floating point vector with 3 dimensions output it and then apply addition and multiplication.

```
// 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
```

Example 2 above shows how to define and output a matrix.

For our assignment, you need to download the Eigen library and leave it the same location as your assignment. Eigen library may need to be included in your project.



2.2 Assignment 1 skeleton

Assignment 1's skeleton is included in the pa1 file. You need to focus on main.cpp file and edit it to get your solutions.

Before modifying the provided skeleton file, please note that the files provided are already compliable and runnable. So please first compile and run the code following the compiling suggestions in section 3 later.

One of the most common errors is "undefined reference to xxx". When you get this error, please check the environment setting and linkage of your VSC.

3. Assignment programming and submission

3.1 Programming assignment description

In this assignment, you are required to fine a point $P=(2,1)$, and then rotate it according to original point counter-clock wise 45°

Output the original coordinates of P and coordinates after rotation.

3.2 Compile

This assignment and all future assignments are required to be compiled using **cmake**.

First, editing the main.cpp file as the requirement of assignment description. Then at the same directory of main.cpp, open Terminal and then type in following commands:

- **mkdir build** (this is to build a folder called "build")
- **cd build** (this is to move to the build folder)
- **cmake ..** (note that the .. implies the one level up, if its current level, use .)
- **make** (compile the code and check if there are any errors)
- **.Transformation** (if there is no errors from last step, Transformation is a execute file and you can run it now. file name "Transformation" may be changed, please refer to the CMakeList.txt file).

You can also refer to the following links about how to use **cmake** to compile codes:

https://terryoy.github.io/2018/01/learn-to-use-cmake.html

https://tuannguyen68.gitbooks.io/learning-cmake-a-beginner-s-guide/content/index.html

3.3 Submission

Compile your solutions and generate screenshots of your output, then write your understanding of output in a file. Compress source codes output and your explanation into one compressed file named **FirstnameLastnamePA1.zip (or any format you prefer).** Submit the compressed file through Blackboard before deadline.

3.4 Evaluation

Assignment 1 is for you to get familiar with programming environment and make sure settings are ready. This assignment will be graded based on the following aspects:

- Get both team members environment setting ready and all libraries ready (25%)
- Define point and rotated point and output them correctly (50%)

- Compile and run codes correctly and can see the output points (25%)