

作业0

环境搭建

本来笔者打算按照实验指导书Assignment0.pdf搭建实验环境，不过因为笔者是在课程已经结束后才开始学习Games101课程，有一些资源已经过期，所以笔者参考了其他教程，搭建的环境如下：

- VMware Workstation 16 pro
- Ubuntu 20.04.3 LTS

安装eigen和cmake

```
sudo apt-get install libeigen3-dev
sudo apt-get install cmake
```

若引用头文件时使用了 `#include <Eigen/Dense>`，则需要将eigen3下的所有文件夹及文件复制到 `/usr/include`

```
sudo cp -r /usr/include/eigen3/Eigen /usr/include
```

参考链接如下：

Eigen3安装及注意

`sudo cp -r /usr/include/eigen3/Eigen /usr/include` 注意：参考cp指令 `/usr/men /usr/zh` 将目录/usr/men下的所有文件及其子目录复制到目录/usr/zh中 因为eigen3被默认安装到了usr/include里了（系统默认的路径），在很多程序中include时经常使用`#include` 而不是使用`#include` 所以要做一下处理，否则有些程序在编译时找不

🔗 <https://www.cnblogs.com/lihanwen/p/9928063.html>

安装make

```
sudo apt-get install ubuntu-make
sudo apt-get install make
```

安装gcc及相关

```
sudo apt-get install gcc
sudo apt-get install g++
sudo apt-get install gcc-multilib
```

代码编写

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

void Train()
{
// Basic Example of cpp
    std::cout << "Example of cpp \n";
    float a = 1.0, b = 2.0;
    std::cout << a << std::endl;
    std::cout << a/b << std::endl;
    std::cout << std::sqrt(b) << std::endl;
    std::cout << std::acos(-1) << std::endl;
    std::cout << std::sin(30.0/180.0*acos(-1)) << std::endl;

    // 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 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
    std::cout << i + j << std::endl;
    // matrix scalar multiply i * 2.0
    std::cout << i * 2.0 << std::endl;
    // matrix multiply i * j
    std::cout << i * j << std::endl;
```

```

// matrix multiply vector i * v
std::cout << i * v <<std::endl;

/*
 * PA 0
 */
// TO DO: Define point P
// TO DO: Define rotation matrix M
// TO DO: M * P
}

void Assignment1()
{
    Eigen::Vector3f p (2.0f,1.0f,1.0f); //定义齐次坐标形式的p点
    Eigen::Matrix3f transformedMatrix(3,3); //定义旋转+平移的总变换矩阵
    double angle = 45.0/180.0*M_PI; //定义旋转角度，单位为弧度
    transformedMatrix << cos(angle), -sin(angle), 1,
                        sin(angle), cos(angle), 2,
                        0, 0, 1;
    Eigen::Vector3f p_Transformed = transformedMatrix * p; //计算转换
    std::cout<< p_Transformed << std::endl; //输出结果
}

int main(){

    //Train();
    Assignment1();
    return 0;
}

```

运行结果如下：

```

wlh@ubuntu:~/桌面/图形学/Assignment 0/代码框架$ g++ -o main main.cpp
wlh@ubuntu:~/桌面/图形学/Assignment 0/代码框架$ ./main
1.70711
4.12132
1
wlh@ubuntu:~/桌面/图形学/Assignment 0/代码框架$ cd build
wlh@ubuntu:~/桌面/图形学/Assignment 0/代码框架/build$ cmake ..
-- Configuring done
-- Generating done
-- Build files have been written to: /home/wlh/桌面/图形学/Assignment 0/代码框架/build
wlh@ubuntu:~/桌面/图形学/Assignment 0/代码框架/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
wlh@ubuntu:~/桌面/图形学/Assignment 0/代码框架/build$ ./Transformation
1.70711
4.12132
1

```

使用cmake进行打包

参考实验指导书的命令即可，如下

```
mkdir build
cd build
cmake ..
make
./Transformation
```

至此，作业0完成！