CSC4140 Assignment II

Computer Graphics

February 1, 2022

Transformation

This assignment is 10% of the total mark.

Strict Due Date: 11:59PM, Feb 15^{th}, 2022

Student ID:

Student Name:

This assignment represents my own work in accordance with University regulations.

Signature:

1 From Model to Screen (80 points)

If we want to get a model to screen, we bacially need the following steps:

1. Model Translation **T**
2. Model Scaling **S**
3. Model Rotation **R**
4. View Translation
5. View Rotation
6. Project

Written in Matrix production:

$$ModelViewProject = Project \cdot View_R \cdot View_T \cdot Model_S \cdot Model_R \cdot Model_T \tag{1}$$

For this assignment, your tasks are

1.1 Implement Model Matrix (40 Points)

Complete the given function.

```
get_model_matrix(float rotation_angle,
                 Eigen::Vector3f T,
                 Eigen::Vector3f S,
                 Eigen::Vector3f P0,
                 Eigen::Vector3f P1):
```

Note: express the tranforms in each steps use homogeneous coordinates. For example use 4× Matrix to express the translation Vector3f T. (You can also validate your results by using the Eigen's lib function.)

Follow the instructions in the give code structure.

1.2 Implement perspective projection Matrix (40 Points)

```
get_projection_matrix(float eye_fov,
                      float aspect_ratio,
                      float zNear,
                      float Zfar)
```

1.3 Implement main() function according to your needs (20 Points)

```
    int main(int argc, const char** argv)
    {
        \\your code
    }
```

1.4 Useful information you need this time

```
\\Member Variables
    \\three transformation matrix
    Matrix4f model, view, projection;

    \\frame buffer: cache what you want to draw on screen
    vector<Vector3f> frame_buf;

\\Member Functions
        \\transfer model matrix as
        \\parameters to rasterizer(given)
        set_model(const Eigen::Matrix4f& m)

        \\set view transformation to view matrix
        set_view(const Eigen::Matrix4f& v)

        \\ transfer projection matrix to rasterizer
        set_projection(const Eigen::Matrix4f& p)

        \\ set pixel(i,j) on screen with color (r,g,b)
```

```
20          \\ and write to frame buffer
21      set_pixel(Vector2f point, Vector3f color)
```

2 Your report (20 Points)

For this assignment, 10% is for the template using. Do submit your report using the given template in the first assignment. 80% is listed in Section 1. 10% is for your completing you main() function and make it easier to use (e.g., define a .xml file to record eye_position, rotation angle or other parameters. This part is not limited)

In your report,

1) You need to define a proper eye position freely

2) define a proper triangle

3) define your eye_fov, aspect_ratio, zNear, zFar.

4) all other missing information

You need to think where you should define the object (triangle), your eye position and other parameters. As if you cannot define a proper parameters, you might get nothing on your screen as your "eye" cannot see it. So in your report, write down this parameters by either inside your code or an extra parameter file (e.g. .xml which you do not always need compile the code), and show the corresponding results on the screen.

In your report, I would like to see at least two corresponding results (at least see a triangle on your screen, you can put more). Make it easier to run for TAs, just save the results images as "results_1.png, results_2.png, ...). Your report and your code result must match each other.

submission list all naming roles is given in last assignment, zip them together submit it.

1) PDF report

2) codes

3) CMake file

4) Readme: describe how to run your code and some extra work you think I can give you more grades.

5) a bash file that can compile the code, run the code(save the results). Direct run and see result.