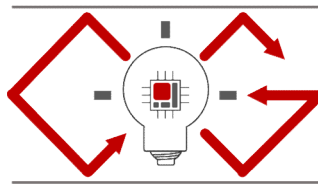


Programming Assignment 2

2021 Computer Graphics



Computer Graphics
Laboratory

Submission

Deadline : 23:59:59, Friday, April 16th, 2021 (KST, +0900)

- Github server clock

To submit your assignment, you **must** do two things, **Both of them must be done BEFORE deadline.**

1. You should push your commit to your assignment repo before deadline.
2. You should comment the last commit (before deadline) id (SHA-1 hash) in github issue board. (See next slide)

The last commit **BEFORE** deadline will be considered as submitted assignment.

- Github server will track this for me.
- Timestamp in your commit (local time) will be ignored. (I will use github server timestamp instead)

Policy

In the following cases, your grade for this PA will be 0

- Late submission (Late push before deadline or Late last commit id comment on issue board)
- Build/execution failure
- Making public of your assignment repository
- If you tried to push your commit with force option(Tried to change history of remote server)

Your final grade will be "F"

- Copy

Task Lists

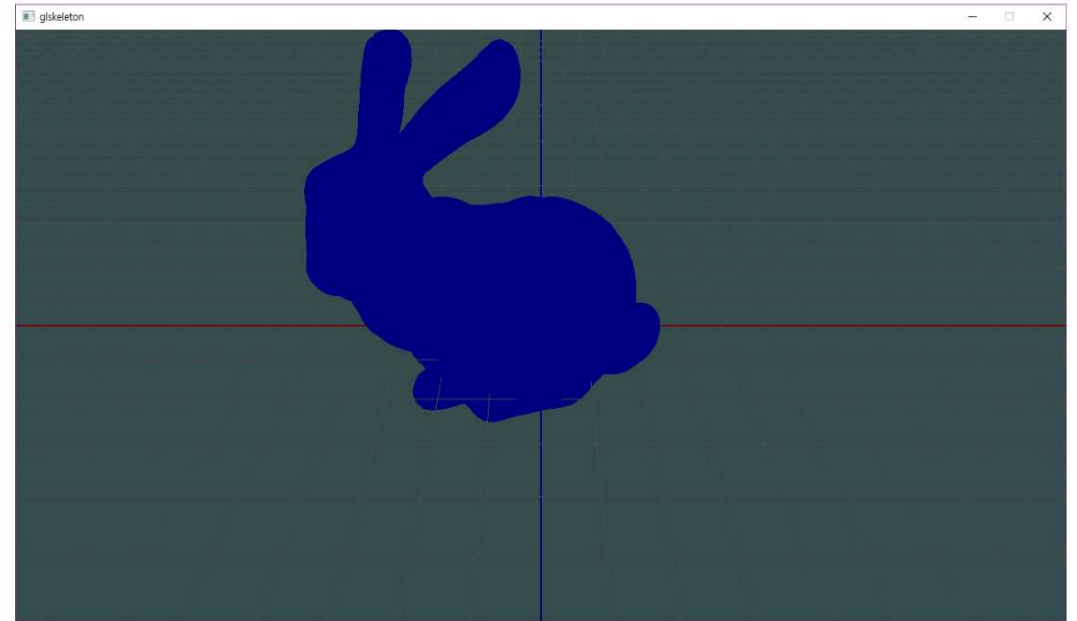
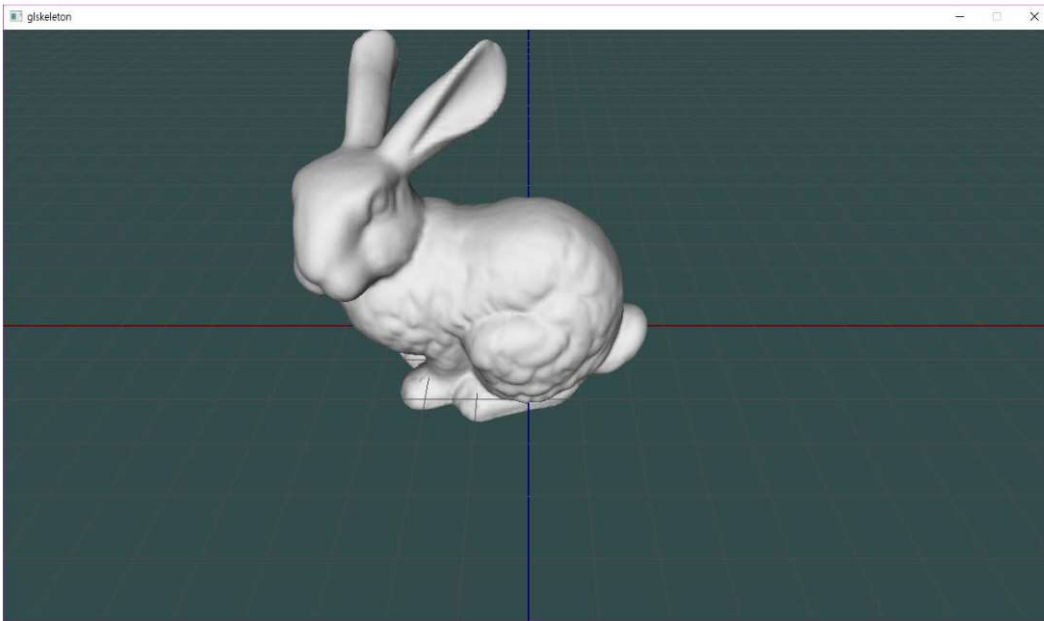
1. Implement object picker [18 Points]

- Load and draw mesh [9 Points]
 - Do not read your mesh with absolute path. If you read your mesh with absolute path, it would not run in my system. (Execution fail, Your score will be 0)
- Implement picking with front and back buffer method [9 Points]
 - Change color when you only pick the surface of bunny
 - Draw the rendering result in the front buffer
 - Draw the image of object id in the back buffer
 - Read the pixel value of back buffer when you click the image and identify what object is under the cursor.

Report [2 Points]

- Write your name, student id, github id in report.md [1 Points]
- Attach at least two result images in report.md [1 Points]

Expected Results



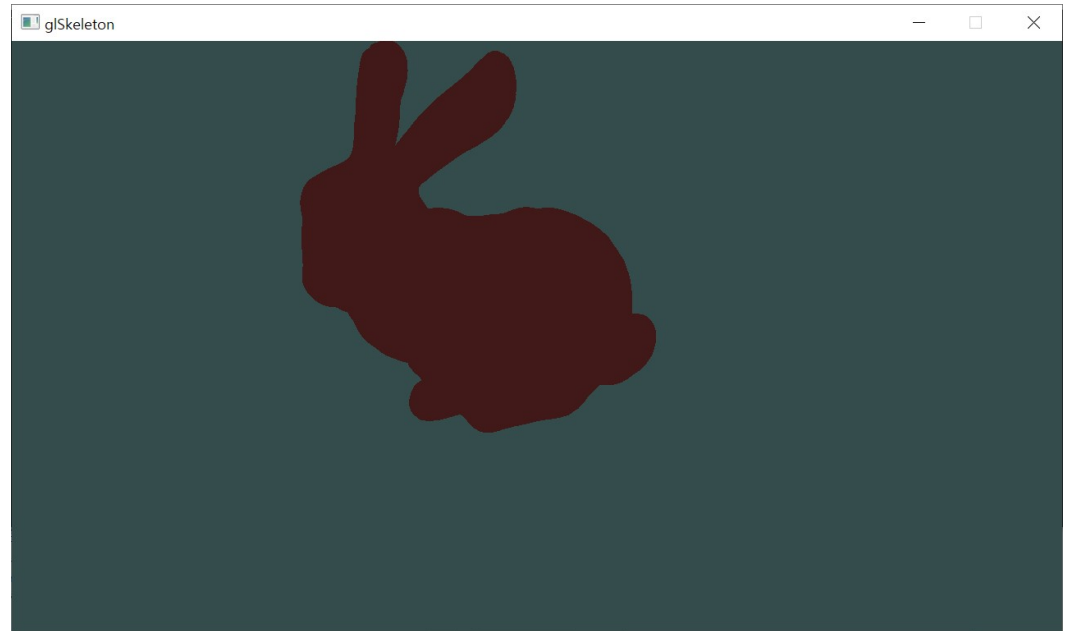
Change the color only when you click on the bunny.

Currently I applied lighting and disabled it when it is picked.
But you don't have to.

2021-03-29

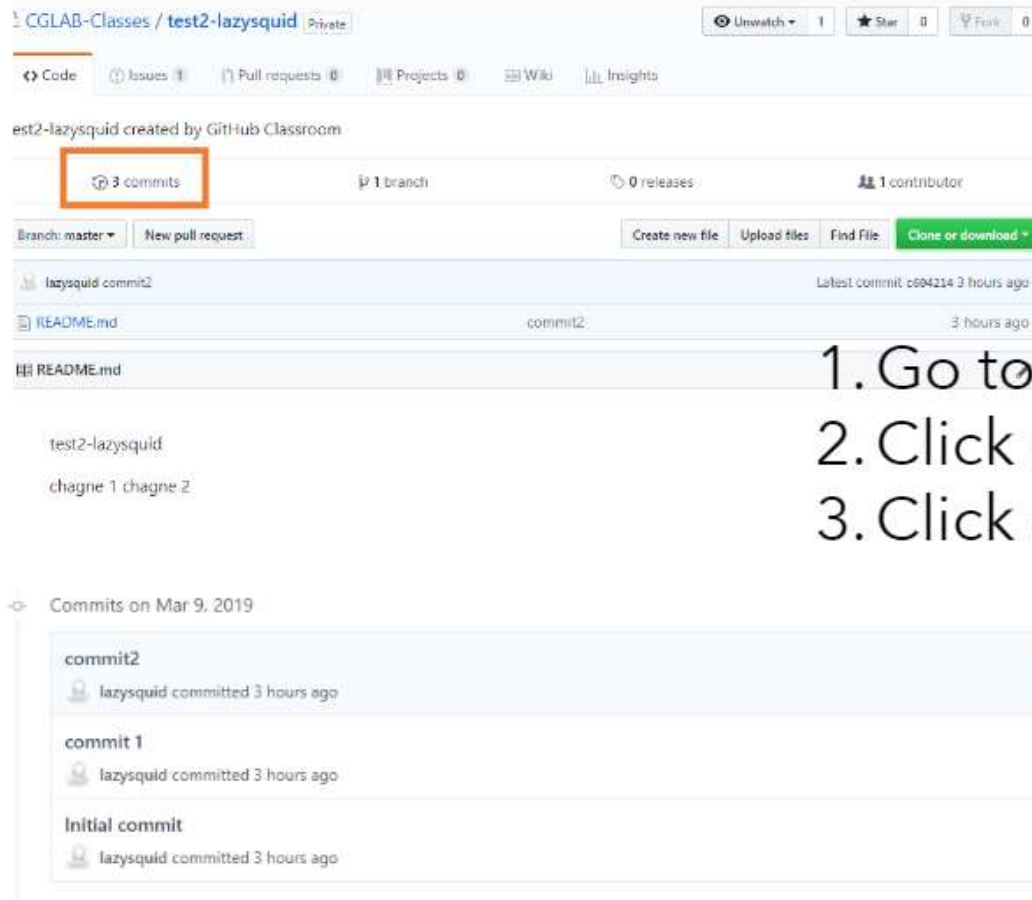
Also, you don't have to draw axes. It is just for debugging

Expected Results



Results without lighting and axes.

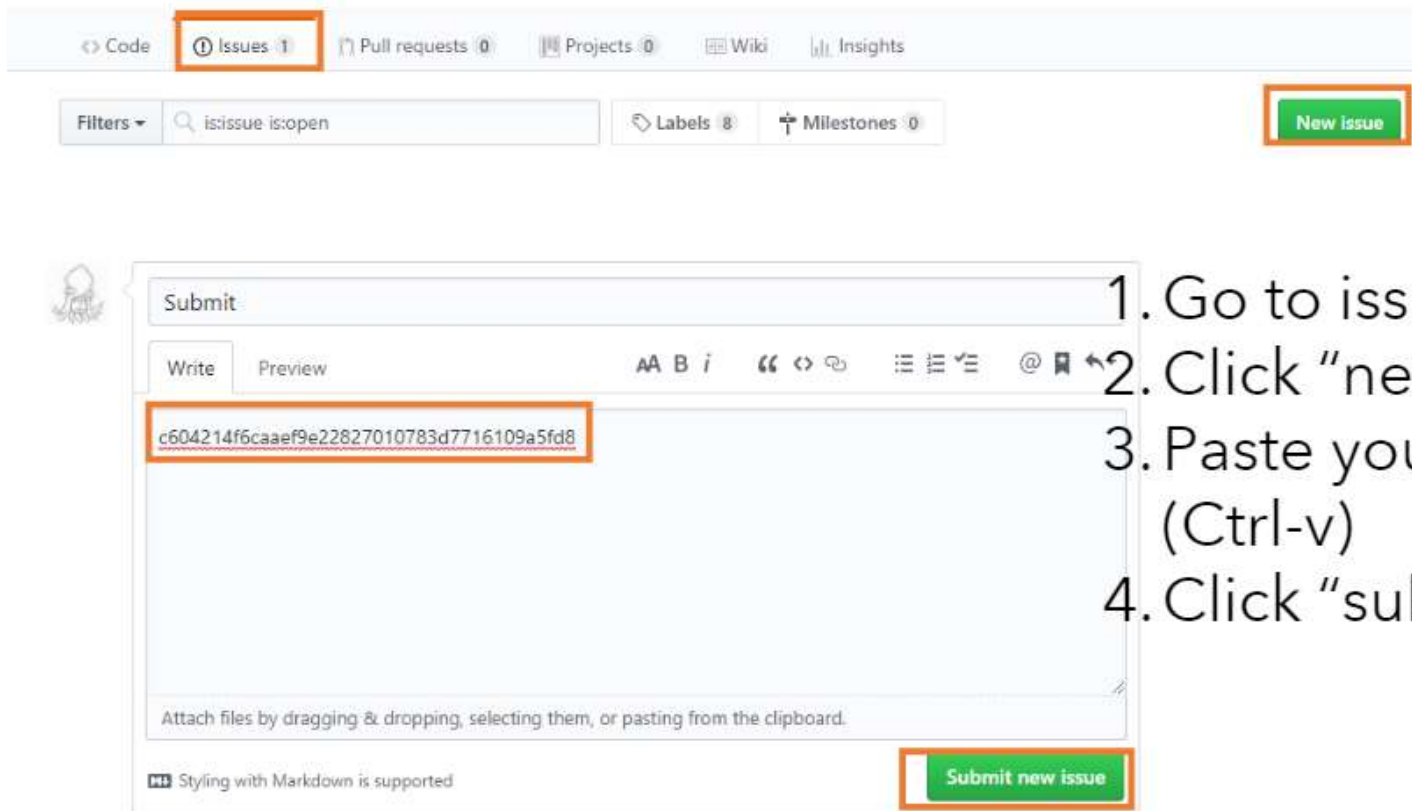
Commenting Commit ID 1/2



The screenshot shows a GitHub repository page for 'CGLAB-Classes / test2-lazysquid'. The 'commits' tab is selected and highlighted with an orange box. Below the repository header, there are buttons for 'Create new file', 'Upload files', 'Find File', and 'Clone or download'. The file list shows 'lazysquid commit2' as the latest commit, 3 hours ago. Below the file list, the commit details for 'commit2' are shown, with the commit ID 'c604214' highlighted by an orange box. The commit details include the commit message 'commit2' and the author 'lazysquid committed 3 hours ago'. Below the commit details, there is a section for 'Commits on Mar 9, 2019' which lists three commits: 'commit2', 'commit 1', and 'Initial commit', each with its respective commit ID and author information.

1. Go to your assignment repository
2. Click commits
3. Click copy button of your last commit

Commenting Commit ID 2/2



The screenshot shows the GitHub interface for creating a new issue. The 'Issues' tab is selected in the top navigation bar. The 'New issue' button is highlighted in green. The 'Write' tab is active, and the commit ID 'c604214f6caae9e22827010783d7716109a5fd8' is pasted into the text area. The 'Submit new issue' button is highlighted in green.

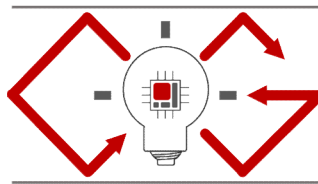
1. Go to issue tab
2. Click "new issue"
3. Paste your latest commit id (Ctrl-v)
4. Click "submit new issue"

PA2 Link

1. Login to github
2. Go to following link <https://classroom.github.com/a/pz8OjXKD>
3. Accept the assignment

OpenGL : Loop

2021 Computer Graphics



Computer Graphics
Laboratory

OpenGL Rendering Loop Example

```
glm::mat4 matModel1 = ...
glm::mat4 matModel2 = ...

glm::mat4 matView = ...
glm::mat4 matProj = ...

// render loop
While (!glfwWindowShouldClose(window))
{
    glfwWaitEvents(); //waits for input
    ...
    // set projection matrix for this frame
    glMatrixMode(GL_PROJECTION); // set projection matrix
    // use either of following lines to set the value of projection matrix
    glLoadMatrixf(glm::value_ptr(matProj)); // you should include glm/gtc/type_ptr.hpp for glm::value_ptr
    glLoadMatrixf(&matProj[0][0]); // you can use this also.

    // set modelview matrix for the model1
    glm::mat4 modelView1 = matView * matModel1;
    glMatrixMode(GL_MODELVIEW);
    glLoadMatrixf(glm::value_ptr(modelView1));
    // draw your model 1
    for(some-condition)
        glVertex3f(...);
    // set modelview matrix for the model2
    glm::mat4 modelView2 = matView * matModel2;
    glMatrixMode(GL_MODELVIEW);
    glLoadMatrixf(glm::value_ptr(modelView1));
    // draw your model 2
    for(some-condition)
        glVertex3f(...);
    ...
    glFinish(); // Do not swap buffer since we use both back and front buffer.
}
```

Hints

1. Init values of model, view, projection matrix that is used in pa2-ref-demo

```
glm::mat4 matModel= glm::identity<glm::mat4>(); //4x4 identity matrix
glm::mat4 matView= glm::lookAt(glm::vec3(0, 4, 4),
                               glm::vec3(0, 0, 0),
                               glm::vec3(0, 1, 0));
glm::mat4 matProj= glm::perspective(glm::radians(60.0f), (float)WIDTH/HIEGHT, 0.1f, 100.0f);
```

2. Set projection and model view matrix

```
// set projection matrix for this frame
glMatrixMode(GL_PROJECTION); // set projection matrix
glLoadMatrixf(glm::value_ptr(matProj)); // you should include glm/gtc/type_ptr.hpp for glm::value_ptr
// set modelview matrix for this frame
glm::mat4 modelView= matView* matModel
glMatrixMode(GL_MODELVIEW);
glLoadMatrixf(glm::value_ptr(modelView));
```

3. Use following functions to set up your object id

Use glColor3ub if you use unsigned integer.
Use glColor3b if you use signed integer.

4. Use string cast to debug your matrix/vector

```
include <glm/gtx/string_cast.hpp>
Then you can do this (left the code, right is the consol output)
glm::vec4 test{1,2,3,4};
std::cout<<glm::to_string(test)<<std::endl; vec4(1.000000, 2.000000, 3.000000, 4.000000)
```

5. Include <tinyobjloader/tiny_obj_loader.h> to use tinyobjloader

Manual Helpers

- <https://github.com/gtruc/glm/blob/master/manual.md>
- <https://github.com/syoyo/tinyobjloader#usage>
- <http://www.opengl-tutorial.org/>