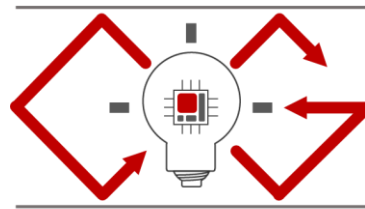


Programming Assignment 2

2023 Computer Graphics



Computer Graphics
Laboratory

TA Hours

Time: every Mon and Wed (13:00~14:00 pm, 20:00~21:00 pm), 104 Dasan Bldg.

Email: Please send me only for score or urgent problems.

- I received a lot of questions and had to schedule a meeting for each. (need a lot of time)
- If it's not important, please use TA hours.

Question rules

- I can't see your code. (Debugging is also your ability)
- We usually give you skeleton codes, and theoretical hints. (I can explain this in detail)
- Compilation and git error. (Please try many things and ask me if it's really hard to solve)
- Welcome to ask me theoretical questions and how to start assignment.

- Final-term exam: 50%

- Programming assignment: 40%

- Attendance: 10%

Submission

Deadline : 23:59:59, Sunday, April 9th, 2023 (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

Your final grade will be "F"

- Copy

Task Lists

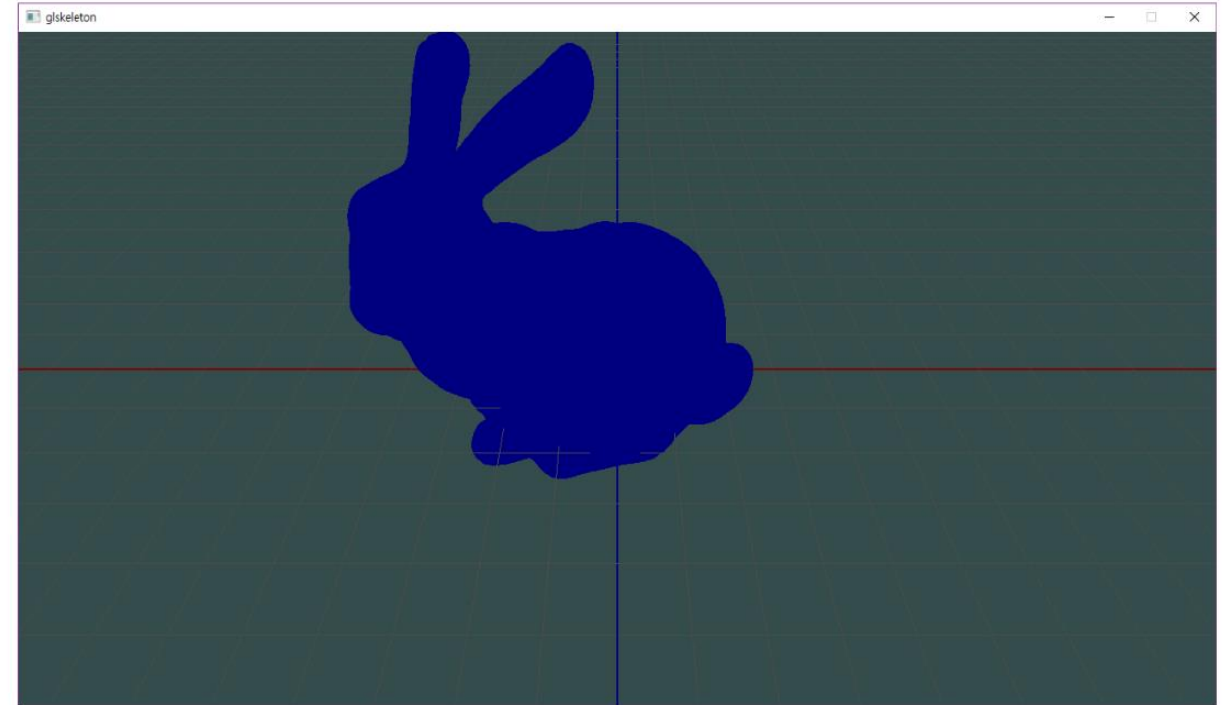
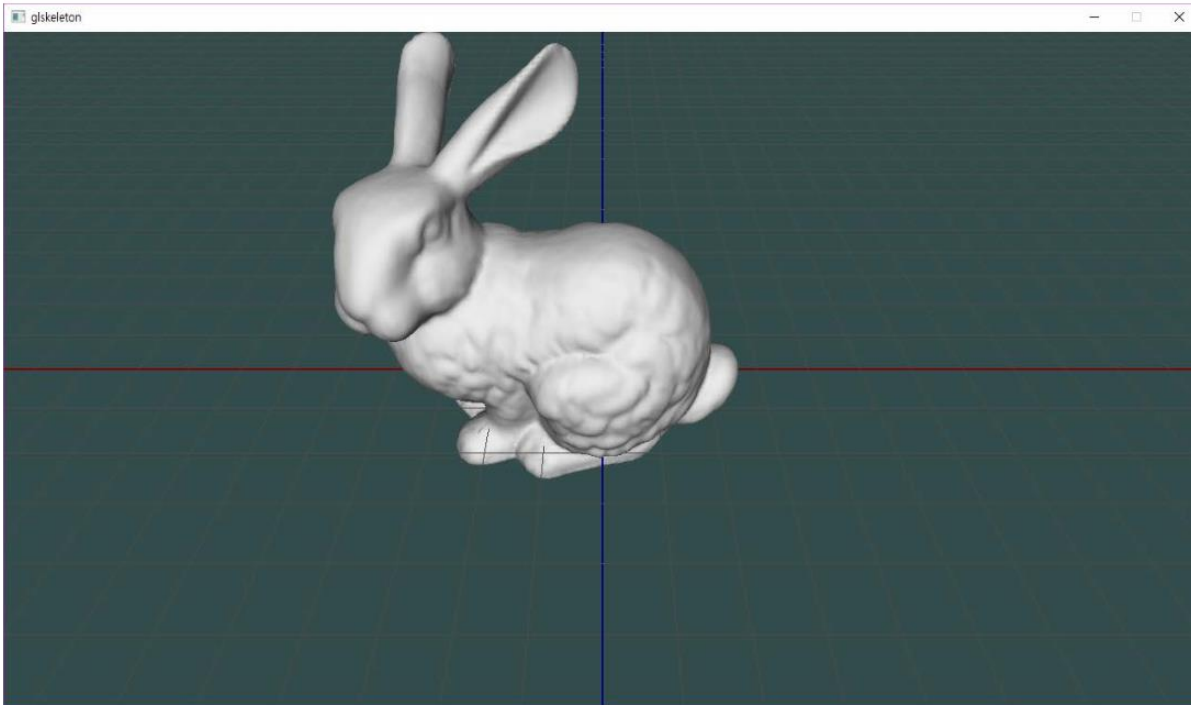
1. Implement object picker [18 Points]

- 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

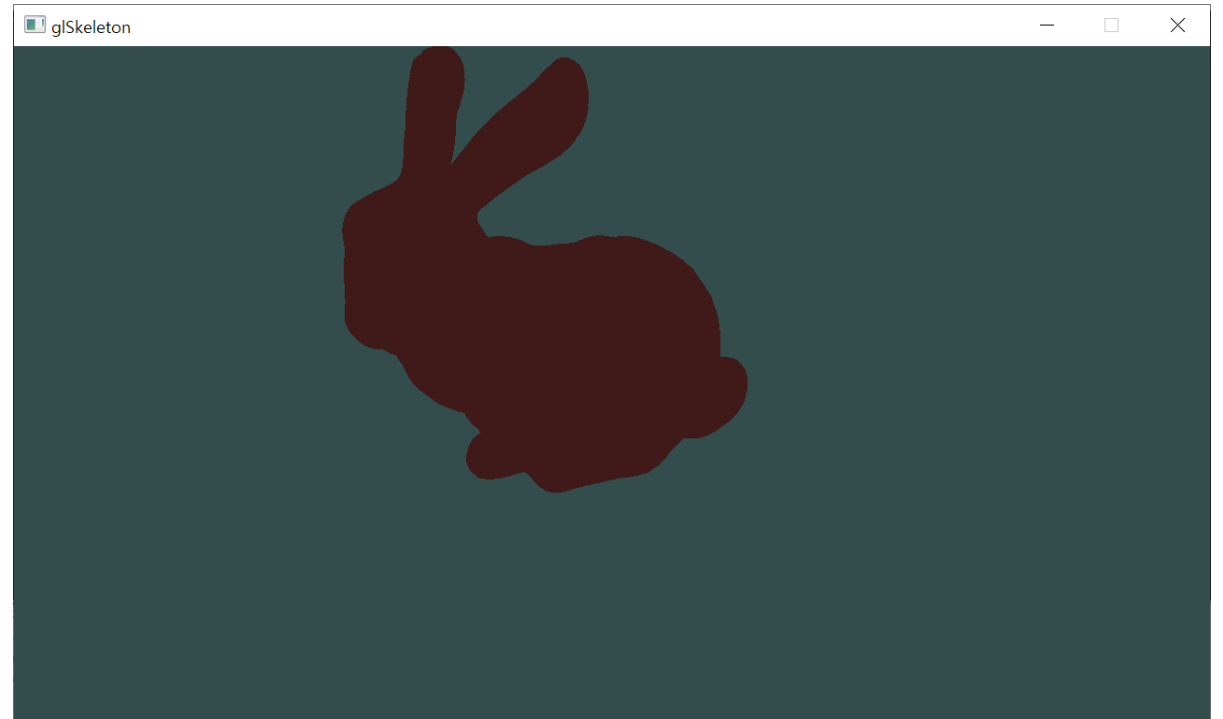
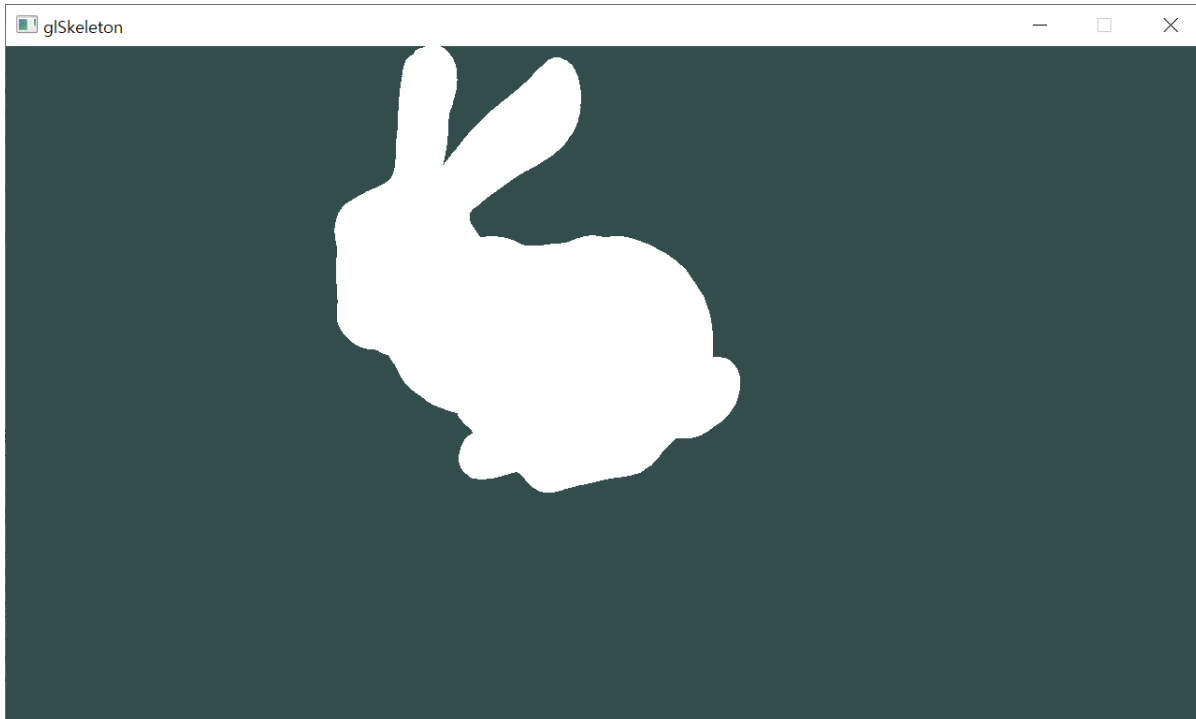


Change the color only when you click on the bunny.

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

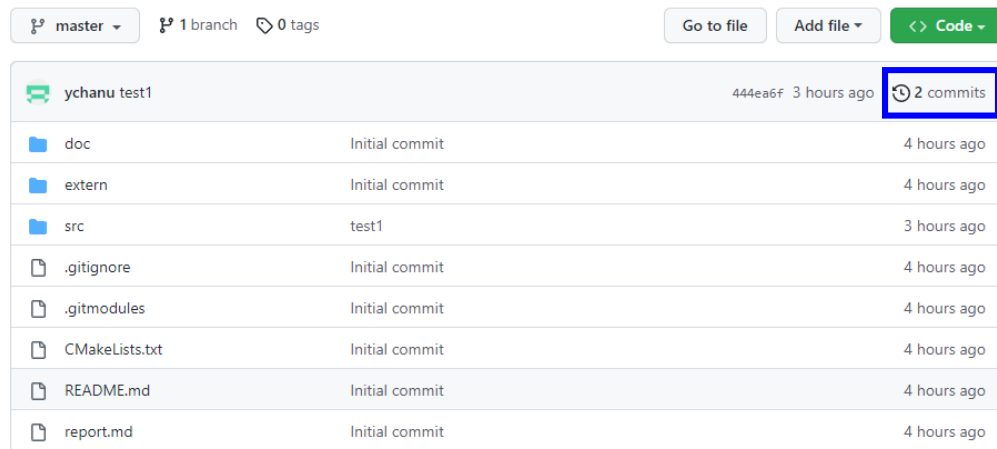
Also, you don't have to draw axes. It is just for debugging

Expected Results



Results without lighting and axes.

Commenting Commit ID 1/2



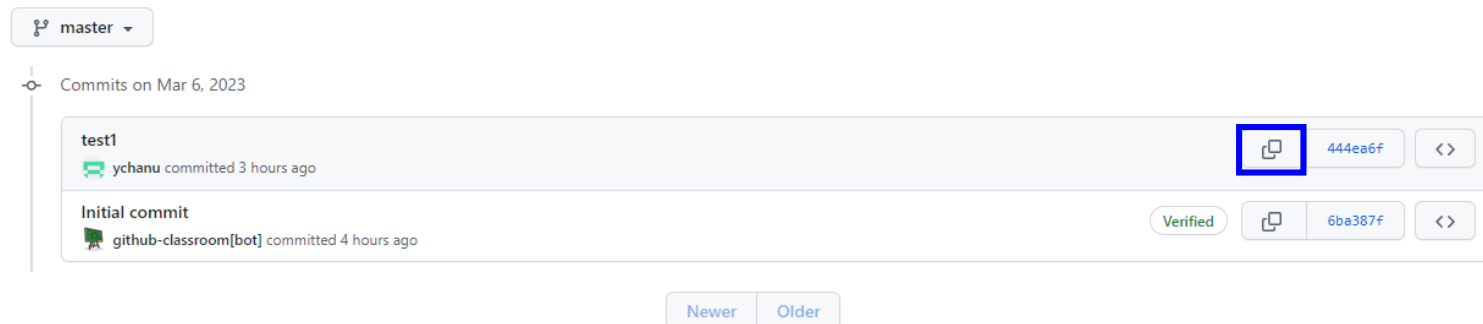
master 1 branch 0 tags

Go to file Add file <> Code

ychanu test1	444ea6f 3 hours ago	2 commits
doc	Initial commit	4 hours ago
extern	Initial commit	4 hours ago
src	test1	3 hours ago
.gitignore	Initial commit	4 hours ago
.gitmodules	Initial commit	4 hours ago
CMakeLists.txt	Initial commit	4 hours ago
README.md	Initial commit	4 hours ago
report.md	Initial commit	4 hours ago

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

<> Code Issues 1 Pull requests Actions Projects Security Insights



master

Commits on Mar 6, 2023

test1	ychanu committed 3 hours ago	444ea6f	<>
Initial commit	github-classroom[bot] committed 4 hours ago	Verified 6ba387f	<>

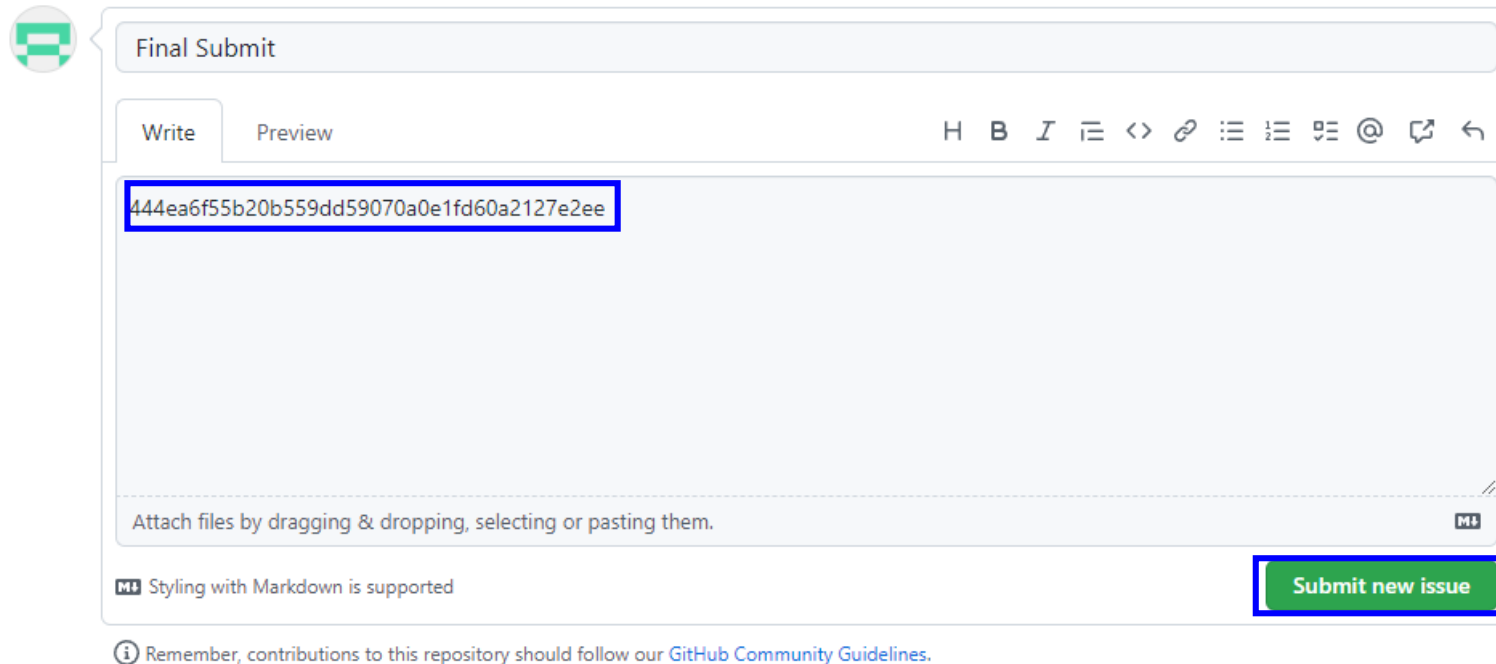
Newer Older

Commenting Commit ID 1/2

The screenshot displays the GitHub interface. At the top, the navigation bar includes 'Code', 'Issues 1', 'Pull requests', 'Actions', 'Projects', 'Security', and 'Insights'. The 'Issues 1' tab is highlighted with a blue box. Below the navigation bar, the repository is identified as 'master'. The commit history for 'Mar 6, 2023' is shown, with two commits: 'test1' by 'ychanu' (committed 3 hours ago) and 'Initial commit' by 'github-classroom[bot]' (committed 4 hours ago). The commit IDs '444ea6f' and '6ba387f' are visible, along with a 'Verified' badge for the initial commit. Below the commit history, there are 'Newer' and 'Older' buttons. At the bottom, the 'Filters' section shows a search bar with 'is:issue is:open', 'Labels 9', and 'Milestones 0'. A 'New issue' button is highlighted with a blue box. Below the filters, the issue list shows '0 Open' and '1 Closed' issues. The main content area displays a message: 'There aren't any open issues. You could search all of GitHub or try an advanced search.'









1. Go to issue tab
2. Click "New issue"

Commenting Commit ID 1/2





Final Submit

Write Preview


H B I       @  

444ea6f55b20b559dd59070a0e1fd60a2127e2ee

Attach files by dragging & dropping, selecting or pasting them. 

 Styling with Markdown is supported

Submit new issue

 Remember, contributions to this repository should follow our [GitHub Community Guidelines](#).

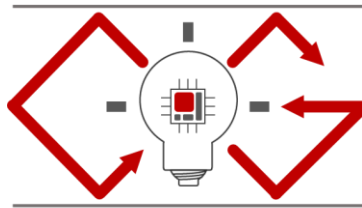
1. Paste your lastest commit id (Ctrl-v)
2. Click "Submit new issue"

PA2 Link

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

OpenGL : Loop

2022 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

```
#include <iostream>
#include <glad/glad.h>
#include <GLFW/glfw3.h>

#include <tinyobjloader/tiny_obj_loader.h>
#include <glm/gtc/type_ptr.hpp>
#include <glm/gtx/string_cast.hpp>
```

Manual Helpers

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

Q&A

- Email: yangchanu@gm.gist.ac.kr
- Office: 104 Dasan Bldg.