

# Programming Assignment 2

---

COMPUTER GRAPHICS

A solid orange horizontal bar at the bottom of the slide.

# Submission

---

Deadline: 23:59:59, Sunday, April 19<sup>th</sup>, 2020 (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.  
-Obviously, e-mail submission is not accepted
2. You should comment the last commit (before deadline) id (SHA-1 hash) in github issue board. (See next slide)

The last commit **BEFORE** dead line 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)

# Commenting Commit ID 1/2

The screenshot shows the GitHub repository interface for 'CGLAB-Classes / test2-lazysquid'. At the top, there are navigation links for 'Code', 'Issues 1', 'Pull requests 0', 'Projects 0', 'Wiki', and 'Insights'. Below this, it indicates 'est2-lazysquid created by GitHub Classroom'. A summary bar shows '3 commits', '1 branch', '0 releases', and '1 contributor'. There are buttons for 'Branch: master', 'New pull request', 'Create new file', 'Upload files', 'Find File', and 'Clone or download'. The file list shows 'lazysquid commit2' as the latest commit, and 'README.md' as a file.

1. Go to your assignment repository

2. Click commits

3. Click copy button of your last commit

This screenshot shows the 'Commits on Mar 9, 2019' section. It lists three commits from user 'lazysquid':

- commit2**: lazysquid committed 3 hours ago. Commit ID: c604214. Includes a copy icon and a code icon.
- commit 1**: lazysquid committed 3 hours ago. Commit ID: ea587c0. Includes a copy icon and a code icon.
- Initial commit**: lazysquid committed 3 hours ago. Commit ID: f8b1e5d. Includes a copy icon and a code icon.

# Commenting Commit ID 2/2

The screenshot shows the GitHub interface. At the top, the 'Issues' tab is selected and highlighted with a red box. Below the navigation bar, there are filters for 'is:issue is:open', 'Labels 8', and 'Milestones 0'. A green 'New issue' button is highlighted with a red box. The main content area shows a 'Submit' form with a 'Write' tab selected. The text area contains the commit ID 'c604214f6caaef9e22827010783d7716109a5fd8', which is highlighted with a red box. A green 'Submit new issue' button is highlighted with a red box at the bottom right of the form.

1. Go to issue tab
2. Click “new issue”
3. Paste your lastest commit id (Ctrl-v)
4. Click “submit new isse”

# Submission

---

- \* Git GUI Client program (personal preference)
  - GitKraken (require Git student pack (pro version will be given for students))
  - SourceTree
  - etc
- ❖ You can easily find how to **commit**, **push** your assignment in your repository

# Git/Markdown

---

- GitKraken tutorials and tips
  - <https://www.youtube.com/playlist?list=PLe6EXFvnTV78WqGmGSq8JPnafR3IAa55n>
  - [https://www.youtube.com/playlist?list=PLe6EXFvnTV7\\_8z5gjobbe9sMjEHNw8\\_GE](https://www.youtube.com/playlist?list=PLe6EXFvnTV7_8z5gjobbe9sMjEHNw8_GE)
  - [https://www.youtube.com/playlist?list=PLe6EXFvnTV7-\\_41SpakZoTIYCgX4aMTdU](https://www.youtube.com/playlist?list=PLe6EXFvnTV7-_41SpakZoTIYCgX4aMTdU)
- Git/github cheat sheet
  - <https://education.github.com/git-cheat-sheet-education.pdf>
  - <https://services.github.com/on-demand/downloads/github-git-cheat-sheet.pdf>
- Github flavored Markdown cheat sheet
  - <https://enterprise.github.com/downloads/en/markdown-cheatsheet.pdf>

# 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

# Policy

---

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

- upload only \*.cpp files
- omit the files in CmakeLists

<https://github.com/CGLAB-Classes/glskeleton#for-windows-user>



# Task Lists

---

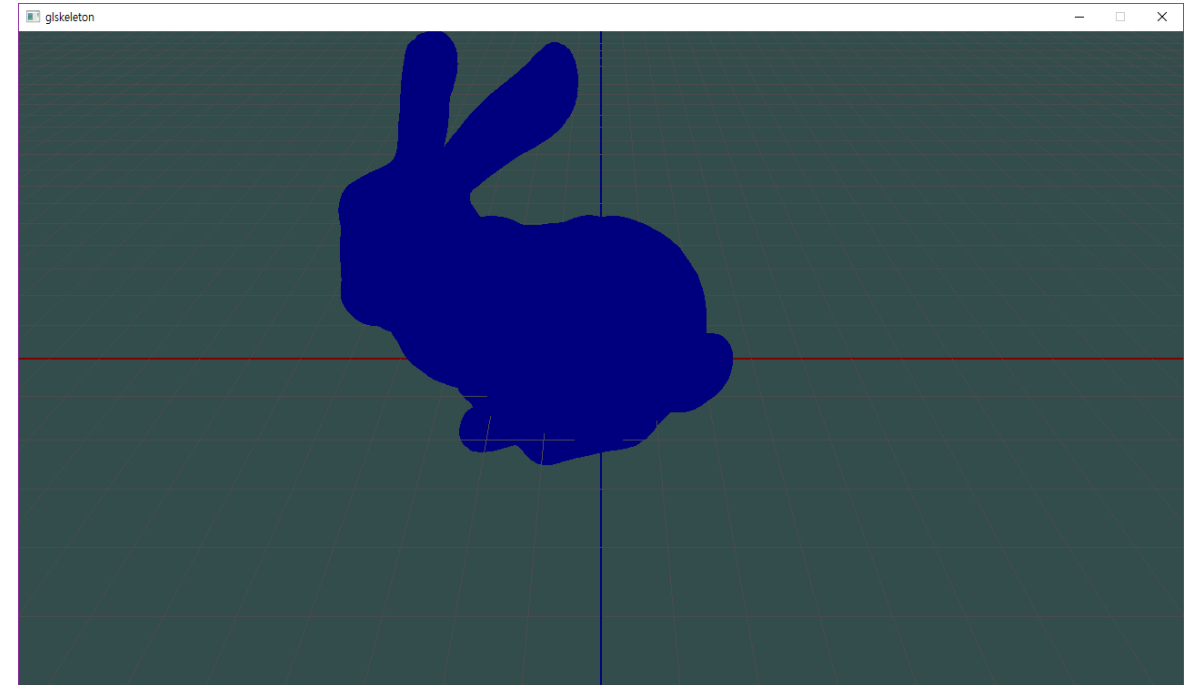
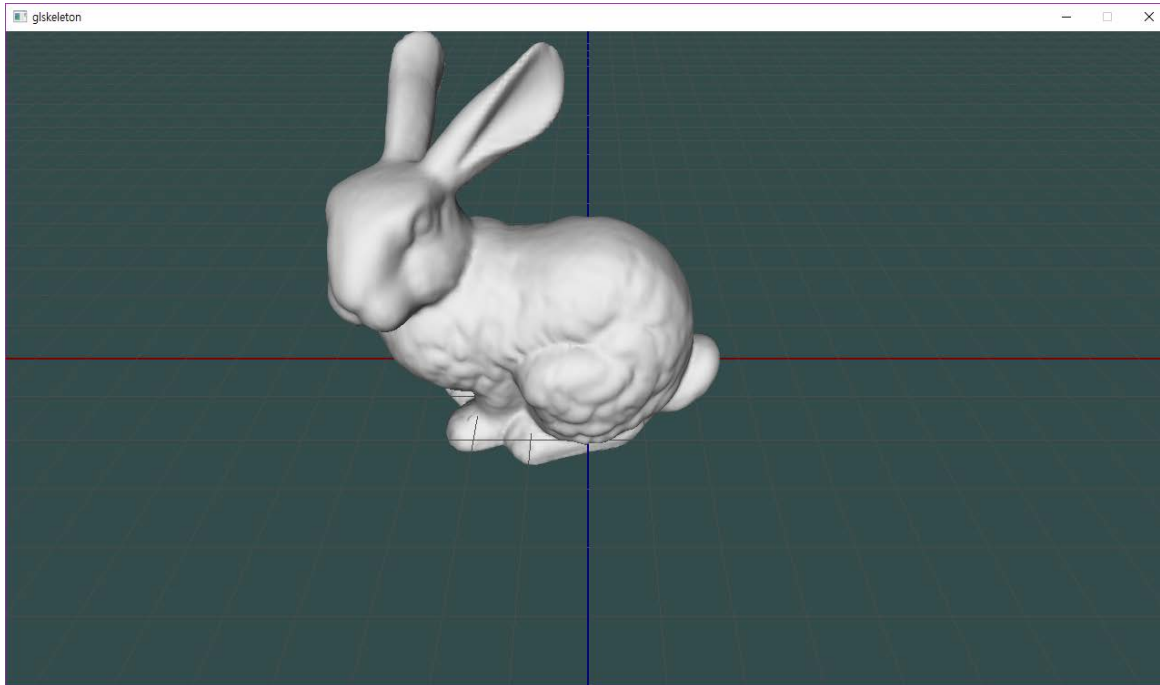
## 1. Implement object picker [18 Points]

- 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.

## 2. Report [2 Points]

- Write your name, student id, github id in report.md [1 Points]
- Attach at least two result(Picked/Unpicked) 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.

# PA2 Link

---

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

# Use `glm::*` instead of `GLU*`

---

- OpenGL 3.0 specification has deprecated some features that have been removed from OpenGL 3.2 core profile specification. GLM provides some advantageous replacement functions.
  - Simply you cannot use GLU functions in your assignment
- `glm::` library provide replacement functions for deprecated `glu*` functions. (Examples)
  - `gluPerspective(float fovy, float aspect, float zNear, float zFar)`  
`glm::perspective(float fovy, float aspect, float zNear, float zFar)`
  - `gluLookAt(GLdouble eyeX, GLdouble eyeY, GLdouble eyeZ, GLdouble centerX, GLdouble centerY, GLdouble centerZ, GLdouble upX, GLdouble upY, GLdouble upZ)`  
`glm::lookAt(glm::vec3 eye, glm::vec3 center, glm::vec3 up);`
- Take a look section 5 to see full deprecated function replacements list.
  - <https://github.com/g-truc/glm/blob/master/doc/manual.pdf>

# Rendering Loop Example

---

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

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

// Render loop
while (!glfwWindowShouldClose(window))
{
    glfwWaitEvents();
    ...

    // 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.
```

# 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));
```

# Hints

---

## 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 `#include <tinyobjloader/tiny_obj_loader.h>` to use tinyobjloader

# Manual Helpers

---

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