

Programming Assignment 3

COMPUTER GRAPHICS



Submission

Deadline: 23:59:59, Sunday, May 10th , 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'. The repository statistics show '3 commits', '1 branch', '0 releases', and '1 contributor'. A toolbar contains 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, with 'README.md' files listed below it.

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 the user 'lazysquid', each committed '3 hours ago'. The commits are: 'commit2' (ID: c604214), 'commit 1' (ID: ea587c0), and 'Initial commit' (ID: f8b1e5d). Each commit entry includes a copy icon, the commit ID, 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 it, the search bar contains 'is:issue is:open'. To the right, there are buttons for 'Labels 8' and 'Milestones 0'. A green 'New issue' button is highlighted with a red box. Below this, the 'Submit' form is visible. The 'Write' tab is active, and the text input field contains the commit ID 'c604214f6caaef9e22827010783d7716109a5fd8', which is highlighted with a red box. At the bottom right of the form, a green 'Submit new issue' button is highlighted with a red box.

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

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

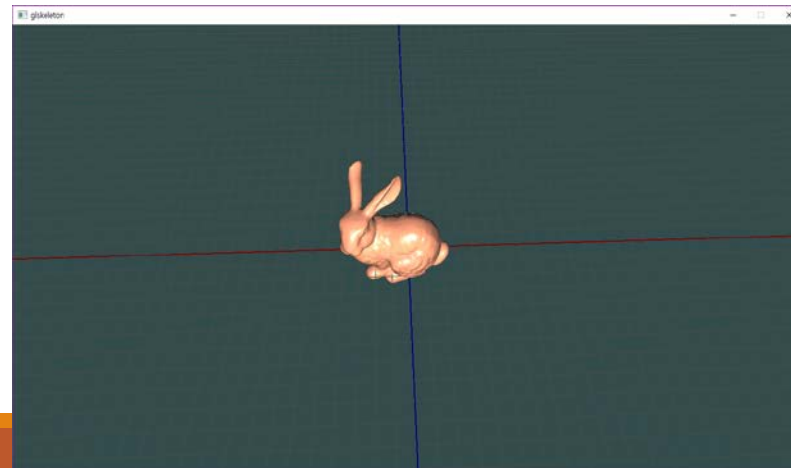
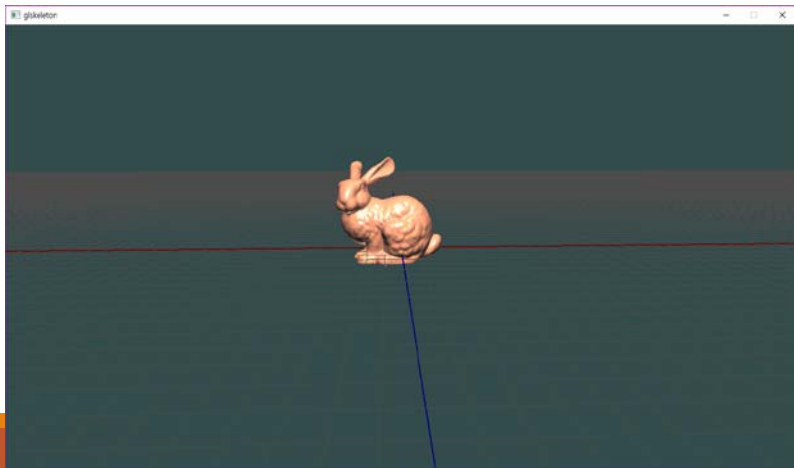
<https://github.com/CGLAB-Classes/glskeleton#for-windows-user>

Task Lists

1. Implement trackball camera [18 Points]
2. Lighting [10 Points]
3. 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]

Task – Trackball Camera

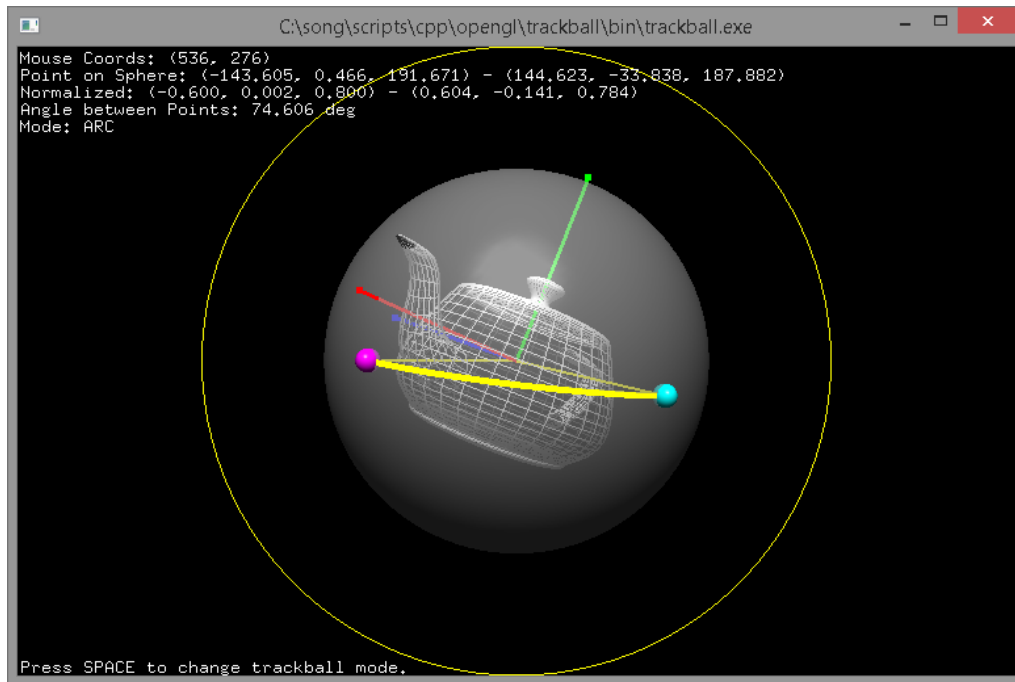
- Fix the lookat position to world **origin**
- Rotate your camera by **dragging** [12 Points]
- Dolly in and dolly out by **scrolling** [3 Points]
 - Move your camera toward/backward to lookat direction.
- Zoom in and zoom out [3 Points]
 - Use key callback to do this. “q” for zoom in, “w” for zoom out



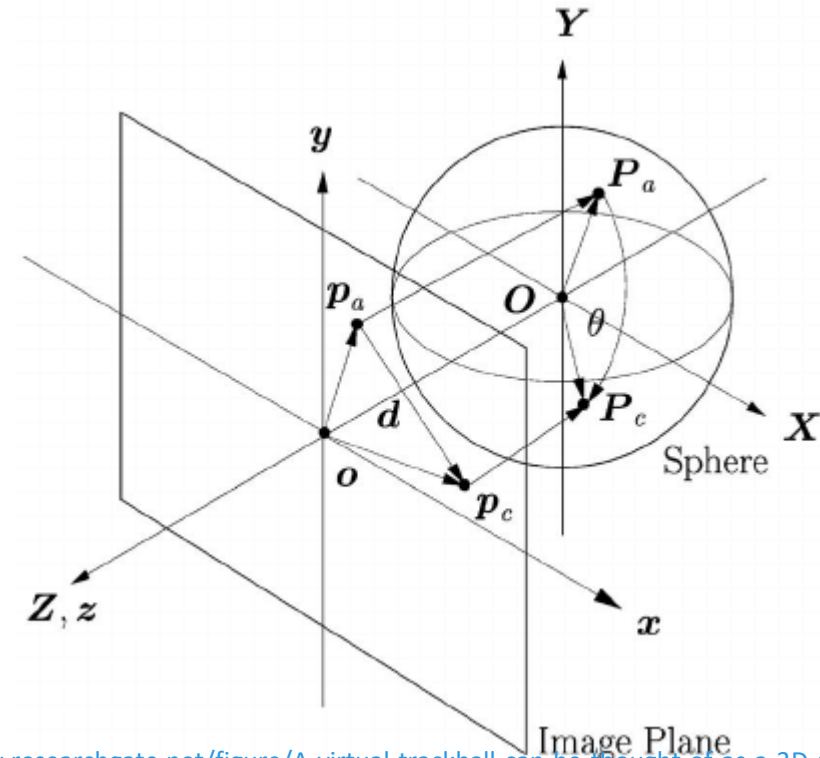
Task - Lighting

- You should apply Phong shading model(ambient/diffuse/specular) and Gouraud shading option.
- Set one point light [5 Point] with on/off functionality.
 - ON, key “1”
 - OFF key “2”
- Set one directional light [5 Point] with on/off functionality.
 - ON, key “3”
 - OFF key “4”

Trackball Cameras



http://www.songho.ca/opengl/gl_camera.html



https://www.researchgate.net/figure/A-virtual-trackball-can-be-thought-of-as-a-3D-sphere-located-behind-the-screen-The_fig2_8329656

The camera which is orbiting around virtual sphere.

Rotation Between Two Vector

```
#include <glm/gtx/quaternion.hpp>
// reference
// http://www.opengl-tutorial.org/kr/intermediate-tutorials/tutorial-17-
quaternions/

glm::quat RotationBetweenVectors(glm::vec3 start, glm::vec3 dest) {
    start = glm::normalize(start);
    dest = glm::normalize(dest);

    float cosTheta = dot(start, dest);
    glm::vec3 rotationAxis;

    if (cosTheta < -1 + 0.001f) {
        // special case when vectors in opposite directions:
        // there is no "ideal" rotation axis
        // So guess one; any will do as long as it's perpendicular to start
        rotationAxis = cross(glm::vec3(0.0f, 0.0f, 1.0f), start);
        if (glm::length2(rotationAxis) <
            0.01) // bad luck, they were parallel, try again!
            rotationAxis = cross(glm::vec3(1.0f, 0.0f, 0.0f), start);

        rotationAxis = normalize(rotationAxis);
        return glm::angleAxis(glm::radians(180.0f), rotationAxis);
    }
}
```

```
rotationAxis = cross(start, dest);

float s = sqrt((1 + cosTheta) * 2);
float invs = 1 / s;

return glm::quat(s * 0.5f,
                 rotationAxis.x * invs,
                 rotationAxis.y * invs,
                 rotationAxis.z * invs);
}
```

Rotation is usually represented with quaternion.
Understanding quaternion is out of scope.

But we can convert them into rotation matrix using glm.

So use this like this

```
const glm::mat4 R =
    glm::toMat4(RotationBetweenVectors(v1,v2)
);
```

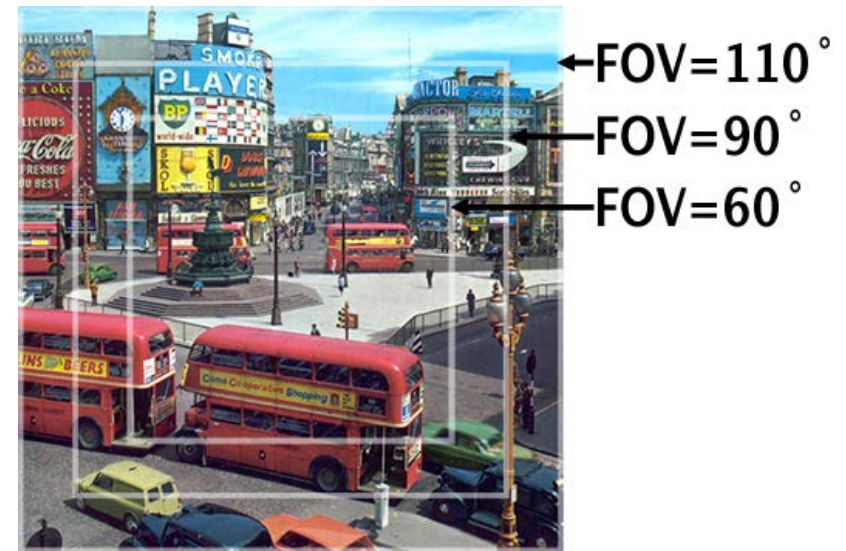
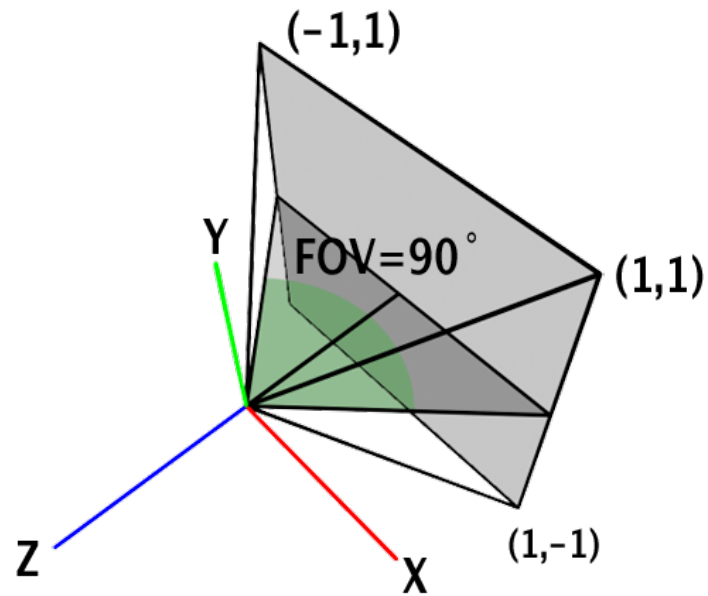
Dolly In/Out

- Translation of camera origin
 - Move your camera toward/backward to look at direction.
- It affects to camera(view) matrix



Zoom In/Out

- Changing of camera
- It affects to projection matrix



PA3 Link

1. Login to github
2. Go to following link https://classroom.github.com/a/_8N6EnE9
3. Accept the assignment