

521140S Computer Graphics (2019) Programming Assignment II

General information

In this assignment, the goal is to create a simple animation that animates the shape of the object. Also, texture mapping, transformations, and hidden surface removal techniques should be used where possible. In addition, a user-controlled camera positioning will be implemented.

Deadline for returning the work is 2019-04-23 24:00. If you need an extension for any reason, please apply for it before the deadline.

The maximum number of points for this assignment is 8, the minimum requirement is to get 5 points to pass the assignment. Missing or incorrect functionality will reduce the points.

Introductions in the task guidelines are based on the Windows platform. Programming on Linux or OS system is possible. Please find the Platform-Specific Information in Common additional material page at Noppa. A submission working on one platform is enough.

The assignment is to be finished alone. Feel free to discuss the problems but sharing code is not allowed.

Do not use any existing libraries to solve the rendering part of the assignment outside what is used in the framework itself. The purpose of this course is to learn how to implement graphics engines and not using existing ones.

Workstations. The framework has all the required libraries and a project file for MS Visual Studio 2013+ included for both 32- and 64-bit platforms. PC classrooms (TS135 and TS137) have Visual Studio 2015 installed so it is possible to work there.

Tasks

1. Render a scene that contains a flag pole that is attached to a flag.
 - a. The code framework for the assignment has been created in *assignment2.cpp* and *assignment2.h*.
 - b. Modify the code in `void Assignment2::createPole()` to create a pole that is attached to the flag. Use colors instead of a texture (`VertexColor` struct instead of `Vertex`). Utilize an index buffer. (1.5 p).
 - c. Render the pole using a separate vertex array object and separate shaders. (2p)
Hint: Create separate Vertex array object, Index buffer object, and Vertex buffer object for the pole in `bool Assignment2::init()`. Remember that you are now sending in color data not texture data! Bind your Vertex array object in `void Assignment2::render()` and load the shaders (*pole.vs* and *pole.fs*)
2. Create an animated flag.
 - a. Modify the vertex shader to animate the flag so that it gives an illusion of wind effect to the flag (*flag.vs* and *flag.fs*) (1.5 p)
Hint: See the shader shown in slides. Pass a time variable with the help of `glUniform1f()`.
3. The camera is controllable using keyboard (and mouse) that enables looking the scene from different positions and directions.
 - a. key control (2 p)
 - a, d: left translation, right translation (0.75 p)
 - w, s: front move, back move (0.75 p)
 - e, r: rotate the camera left/right (along y-axis). (0.5 p)
 - b. Flag and pole are rendered using back face culling and depth testing enabled. (1 p)
 - Do it with at least the flag pole (in the front the pole shows, in the back the pole doesn't show).

Hint: For SDL handle events, here is an example of getting `SDL_PollEvent`, <https://docs.huihoo.com/sdl/sdl-1.0-intro-en/usingevents.html>. The location of the camera can be updated by modifying the `viewMat` with `glm::lookAt` function. <https://www.khronos.org/registry/OpenGL-Refpages/gl2.1/xhtml/gluLookAt.xml>. The back face culling and depth test should be enabled in the `bool Assignment2::init()`.

The scene can also contain other objects like bending trees etc. that fit the scene or additional functionality that you wish to implement.

You can use any of the example code included in the framework directly in your assignment if you find them useful. You can also freely reuse any of the previous code from earlier assignments that you created yourself.

What to return

Please send the following to any one of the TAs using **title format “CG2 _<student id>_ <given name> _<family name>”** during the given deadline date at the latest (Deadline date is at first page, assume Finnish time).

The submission should contain the following documents:

1. A PDF document describing your programming work and how you solved your assignment. Make sure your document contains at least the following information:
 - a. Your name and student id number
 - b. What operating system(s) was/were used to develop and test the software.
 - c. Number of hours needed to finish the assignment
 - d. Description of how you solved the programming tasks (General description of the method, any used external information sources used etc.) Please include information of any problems you encountered and how you managed to solve or work around them.
 - e. Any feedback you wish to provide.
2. A zip file containing **assignment2.cpp, assignment2.h, flag.vs, flag.fs, pole.vs, pole.fs.**

Contacts

Please contact the TAs for submission or for help if you face with problems. The TAs for the programming assignments are as follow:

Tuomas Varanka: tuomas.varanka@oulu.fi (Finnish & English)

Yingyue Xu: yingyue.xu@oulu.fi (English)

Nhat Vo: nhat.vo@oulu.fi (English)