

Computer Graphics exam

2019. 12. 17.

Structure of the exam

Any subsite of cg.elte.hu can be used during the exam, including all [lecture slides](#) and practice programs:

- http://cg.elte.hu/~agostons/cg_en/
- <http://cg.elte.hu/~galgergo/>

Older sites:

- http://people.inf.elte.hu/puzsaai/eng_cg/
- http://vision.sztaki.hu/~danini/Computer_Graphics/
- <http://iffan.web.elte.hu/eng/>

Other usable sites: the [OpenGL reference](#), [GLM's documentation](#), [SDL documentation](#) and [Wikipedia](#)

Nothing else can be used (online forums, tutorials, your neighbour, etc.)

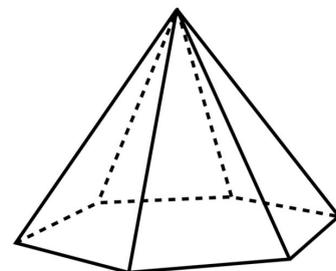
Download and set up the [OGL Pack](#) for the exam, open a project of your individual choice in Visual Studio! Should you have any questions, please don't hesitate to ask!

You need to create certain geometric entities, build a virtual scene and respond to user interactions using C++ and OpenGL. You have 4 hours to finish your program adequately! Once you finished the exam, an examiner will discuss and grade your exam with you. Do NOT leave the lab, without getting your grade.

Tasks

Create a hexagonal pyramid where the base hexagon is regular. The length of the base sides should be 1 unit each, and the top vertex should be 3 units above the hexagon ([hints](#)). **(5 points)**

(Alternatively: create a square based pyramid with the same parameters for 2 points)



Specify normal vectors for each face, such that normals are perpendicular to the triangle. **(1 point)**

Create a planequad (rectangular geometry using VAOs and VBOs or glVertexBuffer) in the $y=0$ plane, of dimensions 30×30 and place it down as the 'ground', the models 'standing' on top of it. **(1 point)** Specify correct texture coordinates **(1 point)** and normals **(1 point)** for the planequad.

Load the [tree.obj](#) from file **(1 point)** and put [this texture](#) on it **(1 point)** Don't change the texture of the ground.

Set the color of the pyramid inside the fragment shader to be constant $RGB(0.5, 0.7, 0.9)$ instead of texturing (use uniforms) **(1 point)**. Put the pyramid 20 units above the ground, pointing down **(2 points)**. Have it fall into the ground (with constant speed), it should take 5 seconds to reach it **(1 point)**, while rotating around its Y axis **(1 point)**. When it has reached the ground, it should start again from its starting position **(1 point)**.

Put the tree (loaded from OBJ) on top of the ground so that it is touched (not above or below) **(1 point)**. Let the user control it with the arrow keys, so that it always stays on the ground and can move in four directions. (E.g. when pressing the right key, it moves a little in the +X direction, when pressing the up key, a little in the -Z direction, etc.) **(2 points)**

Create multiple (at least 10) pyramids all 20 units above the ground, with random positions in the X and Z dimensions (so that they are still above the 30×30 ground) (*you can just use rand() to generate random integers*) **(2 points)**. They should all fall into the ground **(1 point)**. Let them have different speeds, reaching the ground should be a random duration between 3 and 10 seconds **(1 point)**. When they reach the ground, they should start again from 20 units above **(1 point)**. They should be given a new random generated starting X and Z coordinates **(1 point)**.

Let the light source, illuminating the scene, be a point light with green diffuse color. The point should be above the ground by 10 units **(2 points)**.

Total: 28 points

Grading:

0 - 11 : 1 :(
12 - 15 : 2
16 - 19 : 3
20 - 23 : 4
24 - : 5 :)

Hints for unit side hexagon:

The radius of the circumscribed circle is also 1, so all points are on a 1-radius circle around the origin. This means you can compute the coordinates using the *sin* and *cos* functions.

You can divide a hexagon to triangles multiple ways, some examples:

