COMPUTER GRAPHICS

FINAL ASSIGNMENT

GOAL

Create a virtual world within OpenGL. You can use your homework from week 1 as a basis for this virtual world.

This assignment must be completed individually.

TERMINOLOGY

- Model
  - An object in your world. Examples: a house, a tree, etc.. A model sometimes consists of more meshes. For instance, a house is made of the ground floor (a cube mesh) and a roof (a pyramid mesh)
- Primitive mesh
  - A mesh that is defined in C++ (using triangles)
- Predefined object file
  - A downloaded .obj file
  - A mesh that was already present on ELO (box.obj, cylinder18.obj, cylinder32.obj, sphere.obj, torus.obj)
- Designed object file
  - A model or mesh that was created by yourself in 3DSMax, Blender, etc.
- Fragment shading
  - The way material looks
- Fragmentshader
  - fragmentshader.frag
- Modularity
  -

REQUIREMENTS

- Take your homework from week 1 as starting point
- Use at least 4 unique **models**. You can think of lightsources (lamp, lantern), buildings, cars, bikes, etc.
- At least one model consists out of 4 **primitive meshes** (with realistic textures or colors).
- Use at least one **designed object file**
- Use several textures (at least 2)
- For at least 2 models, make unique animations
- Use at least 2 different kinds of **fragment shading** (e.g. create a shiny item in your scene). Make the differences in shading clear.
  - Bonus: Use different **fragmentshaders**
- Possibility to move change camera view:
  - Walk mode : move camera through scene
    - Starting point is on the street (~1,75m from the ground)

        - Keys W/S/A/D : move forward, backward, left, right
        - Keys I/J/K/L : look around (also changing walking direction)
        - Make y-position fixed, so watch out you don't fall through the floor or start flying
        - Bonus:
            - Use the mouse instead of keys I/J/K/L.
            - Make it possible to jump in Walk mode
            - Make it possible to change y-position in Drone mode (Q/E)
    - Drone mode : look at scene from above
        - Choose a nice perspective view (like from a drone) to overview your entire scene
        - It must be possible to use W/S/A/D and I/J/K/L
        - Y-position is not fixed
    - Key "V"
        - If pressed from Walk mode, change to Drone mode. Camera position and angle must have the default starting values
        - If pressed from Drone mode, change to Walk mode. Camera position and angle must have the same values as before View mode

COOLNESS / MODULARITY

You can score points for coolness. See "Graphics design" on evaluation form. You can think about the following:

- Moving lightsources
    - For instance a rising and falling sun
- Very realistic environment (please state this in the evaluation form, maybe together with a picture or a link to streetview)
- Etc., etc.

You can score points for modularity. When adding objects, textures, etc. you can choose to either add a lot of them (in main.cpp) or to make a very modular solution so more objects can easily be added. For modularity, you can think about the following:

- Classes for models, meshes, material, etc.
    - So a model consists of meshes which each have some kind of material
- Render method inside the models / meshes
- Animation method inside the models / meshes
- Etc., etc.

SUBMITTING

- Make sure your solutions works "out of the box". That is, make sure all libraries are located at the locations that have been given during lectures.
- On the evaluation form, fill in the green sections.
    - If you want to earn credits for certain subjects, justify why.
        - For instance, if you want to earn credits for the number of textures, please indicate where you use which texture.
        - For instance, if your world only contains five models, you can earn credits by indicating that your solution easily can be expanded with more (modularity)
        - For instance, if you spent extra effort in comments, design patterns, cohesion, documentation, etc., please indicate on the evaluation form why you earns credits for this.

DEADLINE

Deadline for the final assignment : see ELO