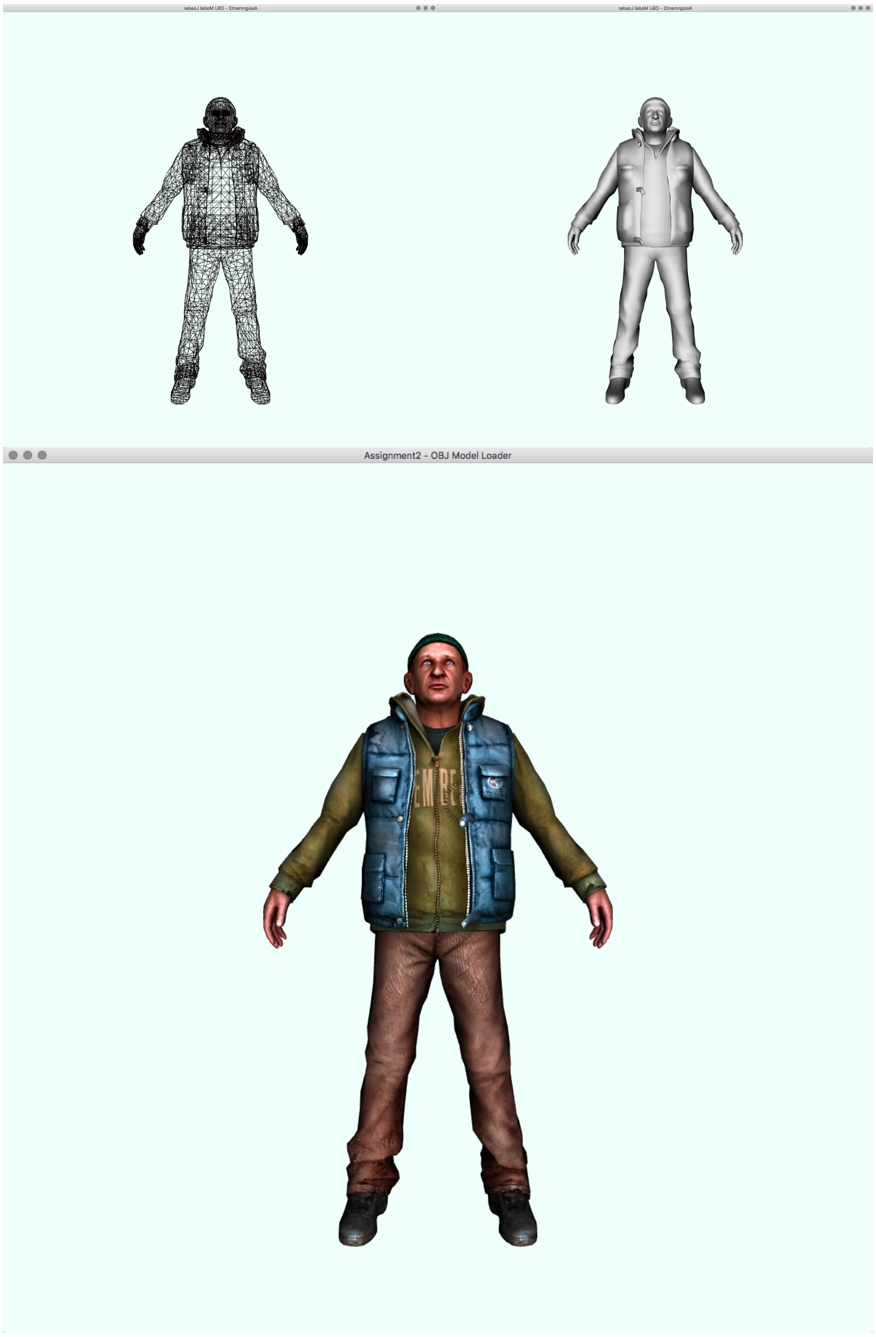
Assignment 3

Deadline:	Hand in by 5:00 p.m. on Friday 12th of June
Evaluation:	30 marks – which represents 30% of your final grade
Late Submission:	Marks will be deducted for late submission if no extension has been granted.
Work:	This assignment must be done **individually** – your submission may be checked for plagiarism against other assignments and internet sources. If you adapt any material from the internet you **must** clearly acknowledge your source.
Purpose:	Design and implement at 3D model viewer program.

Overview:
The goal of this assignment is to write an OpenGL program to load a model mesh from a simple ***wavefront .obj file*** (see `paulbourke.net/dataformats/obj` for information on the format) and render it on the screen using the ***Phong Reflection Model.***



(continued)

Requirements:

1. The 3D model viewer must load an `.obj` model from file and render it on the screen using the ***Phong reflection model*** to calculate the lighting. The user should be able to rotate around the model to view it at different angles.

2. The model loader must support (at the very least) the loading of the ***vertexes*** and ***normals*** from the model file. Your assignment does **NOT** need to support different smoothing groups, free-form curves or transparent meshes.

You should support:

Vertex Data

- Geometric vertices - v
- Texture vertices - vt
- Vertex normal - vn

Elements

- Face - f

Display/render attributes

- Material name - usemtl
- Material library - mtllib

3. For full marks, the model loader should also support reading and rendering of models with ***texture coordinates*,** associated ***materials*** (.mtl files) and ***textures***. Your model loader **ONLY** needs to support the following parameters from the materials file – `Ka`, `Kd`, `Ks`, `Ns`, `map_Kd`, `map_Ks`.

4. You must only use: C++, OpenGL Core Profile, GLM, `stb_image.h`, GLSL, GLFW (and GLEW if your OS requires). Do **not** use any operating system specific code (`windows.h` etc), any window management and/or input/output should be handled with GLFW. For this assignment you must **not** use any other libraries or APIs.

5. Your assignment should properly delete all memory, buffers when it closes.

6. **Extra credit:** some .obj files have normal (bump) maps, loading and correctly rendering models with normal maps will gain you extra marks.

Considerations

- A single model file may have multiple meshes associated with different materials, how are you going to store and represent these different meshes?

- Will you write a single, configurable shader program or use different shader programs for different options?

- Not all .OBJ files available online are well-formed, see the stream site for a link to the example model.

You **must** put the following comments at the top of your program code and provide the appropriate information.

```
% Assignment number, 159.709, 2020 S1
% Family Name, Given Name, Student ID,
% Explain what the program is doing . . .
```

Hand-in: Submit your script electronically through Stream

If you have any questions about this assignment, please ask the lecturer.