

# Computer Graphics Assignment – Final Project

Grade: 60%

**DEADLINE:** Friday, December 15<sup>th</sup>, 2023

---

*The purpose of this assignment is to allow you to extend the theory you learned in lectures by creating a post-apocalyptic visualization of Dublin City, using C/C++ and shader-based OpenGL.*



*Figure 1: example ideas for the visualization*

## SUBMISSION

You are required to submit by the deadline above:

- A final detailed report, in the template provided, describing all required and advanced features.
- A detailed YouTube video demonstrating all **required and advanced features**. Please provide a voiceover and/or overlaid text/arrows etc.

### 1. Outline

- This assignment is strictly **individual** (no groupwork).
- You **must** use modern shader-based OpenGL (V3 or higher), in order to pass the project.
- You must use C/C++.
- You are required to create a 3D visualization of a post-apocalyptic Dublin City, with an animated crowd of 3D citizens/rebels/zombies/monsters. Specifications at the end.
- **NOTE:** Your project will be demonstrated in your report and recorded videos, **but you may additionally be required to demonstrate your working program to the lecturer.**

For the deadline, you will be required to submit via Blackboard:

1. a PDF report that explains your design and technical choices, with images. Note: We will use this report as an extra input to determine the effort you put in, and to clarify anything that isn't clear from the video demo. It should be between 6-10 pages, depending on how many images/figures/pseudocode you add;
2. all code source files and assets for the apocalyptic visualization demo in a ZIP file (or split it across 2 files if it's slow to upload);
3. a YouTube link to video-recordings of your required and advanced features demos (videos can be recorded using a tool such as FRAPS or Nvidia Shadowplay);
  - NB: The videos should be maximum 7 minutes long and should clearly demonstrate all features. Please consider providing a voiceover and/or overlaid text/arrows etc.
  - **If it can't be seen on the demos, it can't be graded.**
4. A mandatory declaration that the work on both the programming and written assignment is entirely your own and you have not collaborated with anyone.

*Try to demonstrate the technical capabilities of your project with visually pleasing images/videos.*

---

#### Important notices

- If you fail to submit your demo or report on time without a certified reason, you will be reported as absent and will receive a grade of 0%.
- You can use GLUT or any other library you like for systems calls, windowing, menus, interaction devices and other such tasks.
- You can use Windows, Linux or whatever platform you wish. (*Note: you may not use WebGL, as the functionality is a subset of OpenGL, and it uses JavaScript and not C/C++ as required.*)
- It is allowed to use a library to load models, as long as this is acknowledged in the report.
- It is also allowed to use a library for some special effect, extra to the core functionality, such as physics, as long as this is acknowledged in the report. If in doubt, ask the lecturer or the demonstrators.
- It is **not** allowed to use a graphics engine (e.g., UE4, Unity, etc.). This is a test of your ability to program the basic 3D graphics functionality covered in class, so no higher-level libraries or engines are allowed for rendering, camera transformations, etc.

**Be aware that demonstrating code that was not created by you, without crediting the source, is considered plagiarism and will be reported as such. Each student should produce a unique individual project.**

---

## 2. Specifications:

Your post-apocalyptic visualization of Dublin City can take any form you like, but the project must have the following basic functionality:

1. 3-dimensional objects and views (note: fixed top-down orthographic view not allowed).
2. User interaction and camera-control
  - user should be able to move around the scene using the keyboard and/or the mouse. At a minimum, implement moving forwards and backwards, turning left and turning right;
3. A Hierarchical animated Zombie/Robot etc. character or object relevant to the theme;
4. A crowd of moving Zombies/Robots etc.. characters or objects relevant to the theme;
  - The crowd can be implemented in any way that you like. At a minimum, the crowd members should be translating around the scene, e.g., forwards and backwards. (A crowd simulation algorithm such as boids can be implemented as an Advanced Feature and having a hierarchical character for each crowd member is also an Advanced Feature).
5. Texture-mapping your scene and creatures using image files (e.g., jpg);
6. Implementation of the Phong Illumination model. Specular highlights must be visible;
  - Multiple light sources (at least 2, can be point, directional, or spotlight)
  - Multiple different material properties (at least 5 on 5 different objects – e.g., you could have several crowd characters with different material properties)
  - Normal must be transformed correctly
  - Shading must use a combination of ambient, diffuse, and specular lighting

This required functionality is worth up to 70% of the project mark

The final 30% will be given for advanced features, with approximately 15% per perfectly-executed feature – simple features will receive less, whereas elaborate or well-designed features will receive more.

Advanced Features can include the following, or indeed others that you think of:

- More complex boids/crowd animation
- Hierarchical creatures that move realistically in your crowd
- Simulated dynamics or physics of any sort
- Intelligent Characters
- Great models that you made yourself
- Height-mapped terrain
- Procedurally generated terrain or meshes
- More advanced texturing effects; multi-texturing, environment-mapping, bump-mapping, specular-mapping, etc.
- ??? – your own feature ideas

Note: The marking scheme provided shows the **maximum** marks that can be obtained for each section if completed perfectly. Merely attempting a section does not imply the full score indicated.