

CSU4405 Computer Graphics

Final Project Guideline

Binh-Son Hua

Trinity College Dublin

In the final project, you will develop a computer graphics application to showcase the techniques you have learned in the module in a single framework.

The project title is **Toward a Futuristic Emerald Isle**. This project is strictly **individual** (no groupwork). Your project will be demonstrated in your project deliverables, but you may additionally be required to demonstrate your working program to the lecturer upon request.

Your application should include the following features:

- The application is implemented in C/C++, using shader-based **OpenGL 3.3**.
- A minimum frame rate of **15 FPS** must be achieved, when running on the latest generation of GPU (i.e., a desktop 4090). Refer to Lab 4 on how frame rate is calculated and displayed in the window title.
- The application should demonstrate an **infinite** scene. The camera should be controllable (using up, down, left, right keys or mouse buttons). When the camera moves, the application should simulate an endless effect, demonstrating that the camera can move without going out of the scene.
- The application should include the **four basic features** covered in Lab 1, 2, 3, 4: geometry rendering, texture mapping, lighting and shadow, and animation.
- The application should allow 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.
- The application should include an implementation of one of the following advanced features that are not discussed in the class. For other features you are welcome to discuss with the lecturer before implementing them.
 - Deferred shading
 - Screen-space ambient occlusion
 - Screen-space depth of field
 - Environment lighting
 - Level of details
 - Instancing
 - Real-time global illumination, e.g., voxel cone tracing
 - Physics-based animation, e.g., particle systems, smoothed particle hydrodynamics
 - Support multi-platform graphics: Android/iOS, WebGL, AR/VR.

Deliverables

The deliverables of the final includes:

- A final report (max 4 pages) that includes:
 - (1) **an introduction** of your application (what is it about, what features you implement, what is your achievement);
 - (2) **progress report** that demonstrates the development of the application over time by showing at least 5 screenshots that capture the application rendering at each stage of the development;
 - (3) **a discussion** on the quality and robustness of the application;
 - (4) **a discussion** on current limitation and potential future work.
 - (5) **an acknowledgement** paragraph for any peers helping or discussing with you in the project, and for any open data and source code used.
- An illustrative mp4 video that captures the final state of your application. 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. The video should illustrate the main results of your application. By default, we might consider uploading selected videos to a YouTube playlist for reference for future classes. You can opt-out if you do not want your work to be published by sending an email to notify the lecturer.
- All source code and data (C++ code, shaders, model files) packaged in a zip file including a Git repository that stores all the history of your code development.

For the project report, it is recommended that you use Overleaf and LaTeX and follow the ACM SIGGRAPH template to write your report.

- A template on Overleaf is provided here:
<https://www.overleaf.com/read/vtbyjvngrzgze28726>
- Trinity College Dublin provides professional Overleaf subscriptions for staff and students. <https://www.overleaf.com/edu/tcd>

Submit all your deliverables to **Blackboard**.

Timeline

- Project final submission: **Sunday, December 29, 2024 at 23:59 (midnight)**.
- Late submissions are accepted until **Sunday, January 05, 2025 at 23:59 (midnight)**. After this date, the submission system will be closed. **No further submissions** are allowed after this date.

Evaluation criteria

The project is 60% of the total module marks with the following breakdown:

- Originality, creativity: **10%**
- Technical quality and complexity: **30%**
- Robustness: **10%**
- Report: **10%**
- A bonus of max 10% will be given to projects that demonstrate advanced feature implementations.
- A penalty of 20% is applied to late submissions, meaning you will get a maximum of 40% for your project.
- Note that not showing the progress report will result in significant deduction in the technical marks. If you forgot to capture screenshots during the development, you can disable some of your code and take screenshots. Keep a Git history will help in this case.
- Academic dishonesty and misconduct (e.g., plagiarism, fabrication) are strictly not tolerated. Doing so will result in a penalty for your project evaluation.

On the use of open-source code and GPT/AI models:

- You are free to explore open source and GPT-generated code to assist your development.
- You can explore GPT/AI models to create assets for your application such as 3D geometry, textures. It is good to give credits in such cases (which model is used for such content generation).
- You cannot use GPT to generate all the project deliverables. Doing so will result in a zero mark for your project.
- 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.

Further Notes

- You can use OpenGL version ≥ 3.3 as well as other window framework such as SDL if you need some additional features there, but you will need to provide a justification in the report.