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 theme is **Rendering an Imaginative Wonderland**. 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. The application must be developed based on code base from your labs. Using third-party code (e.g., from LearnOpenGL open sources) without strong justification will result in mark deductions.

- 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 endingless effect, demonstrating that the camera can move without going out of the scene. Infinite perception should not only include a skybox or a planar surface, but should handle foreground objects as well.

- The application should include the **four basic features** covered in Lab 1, 2, 3, 4: geometry rendering, texture mapping, lighting and shadow, and animation. Animation should include object movement and skeletal animation with skinning applied.

- 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 labs. Note that the marks given for each feature could be different depending on its technical and implementation complexity. Below is a rough categorization of the features based on their technical challenges:
    - Easy: GPU instancing; shader effects like toon shading, bloom, fog, etc.
    - Easy: normal mapping, displacement mapping, wireframe rendering
    - Medium: Deferred shading; Screen-space ambient occlusion; Environment lighting
    - Medium: Physics-based animation, e.g., particle systems, smoothed particle hydrodynamics;
    - Hard: Ray tracing features such as reflection, refraction, diffuse reflections.
    - Hard: Cascaded shadow maps; Depth of field effects
    - Hard: 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 request for permission to show your demo videos for reference to future deliveries of this module. You can opt-out if you do not want your work to be shown to the next class 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/vtbyjvngrzgz#e28726
- Trinity College Dublin provides professional Overleaf subscriptions for staff and students. https://www.overleaf.com/edu/tcd

Submit all your deliverables to **Blackboard**.

- Please avoid submitting large files to Blackboard during deadline as the server could be unresponsive.
- Instead, upload a large zip file of the project (if you have) to OneDrive, and submit the link in a txt file to Blackboard.

Timeline

- Project final submission: Monday, December 29, 2025 at 23:59 (midnight).

- Late submissions are accepted until Monday, 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 up to 10% can be rewarded to an excellent project implementation. Note that this cannot be used to make up other marks, i.e., the total of the project will be 60% max.
- For late submissions, 10% will be deducted from your marks, meaning that you will get a maximum of 50% 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.

- For each feature implemented, simple features will receive lower scores, while elaborate or well-designed features will receive more. The evaluation criteria provided shows the maximum marks that can be obtained for each requirement. Merely attempting a section does not imply the full score indicated.

- In rare occasions, the lecturer might request for a demo of the project if the submission package has insufficient details for a proper evaluation. If you fail to show up for the demo, you will be reported as absent and will receive a significant mark deduction to your project.

- Academic dishonesty and misconduct (e.g., plagiarism, fabrication) are strictly not tolerated. Doing so will result in a penalty for your project evaluation.

AI and vibe coding policies

- You are free to explore AI-generated code to assist your development. Please put a comment on which code section is generated by AI.

- You can explore AI pretrained models to create 3D assets for your application such as 3D geometry, textures. Give credits in such cases (which model is used for such content generation).

- You are not allowed to use AI to generate all the project deliverables. Doing so will result in a zero mark for your project.

- You are not allowed to use AI to generate the final report. Doing so will result in a zero mark for your report.

- It is fine to use some external libraries to load assets, again, please acknowledge this in the report.

- It is 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.

- Do not use a graphics engine (e.g., UE4, Unity, etc.) for this project. This is a test of your ability to program the basic 3D graphics functionality covered in class, so no higher-level libraries or engines should be used for rendering, animation, etc.

- 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.