Assignment – Final Project

Grade: 50/60 of the project grade

The purpose of this assignment is to allow you to extend the theory you learned in lectures by creating a winter wonderland visualization with crowds.

Figure 1: example ideas for the visualization

DEADLINES

There are two deadlines for the final project:

- Deadline 1 is 6th December 2022. On that date, you must have all the required features completed and shown on a YouTube video with accompanying detailed report. Note that there will be no demos this year, as we have too large a class.. so the you-tube video should be detailed. Please consider providing a voiceover and/or overlaid text/arrows etc.
- Deadline 2 is 6th January 2023. On that date, you will show your final demo, including all features, highlighting the advanced features, on another detailed YouTube video and an accompanying detailed report. Your marks for the advanced features will be based on these.

1. Outline

- This assignment is strictly **individual** (no groupwork).
- You are required to create a 3D winter wonderland visualization with animated 3D snow-people/reindeer/robins using modern shader-based OpenGL (version 3 or higher). Specifications at the end.
- Your project will be demonstrated in your report and recorded video, but you may additionally be required to demonstrate your working program to the lecturer, if requested

For each 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 can be between 3-5 pages, depending on how many images/figures/pseudocode you add.
2. all code source files and assets for the winter wonderland visualization demo in a ZIP file
3. a YouTube link to a demonstration video-recording of your demo (videos can be recorded using a tool such as FRAPS or Nvidia Shadowplay). NB: The demonstration should be maximum 5 minutes long and should clearly show all features (e.g., demonstration of each part of the hierarchical animation, changing the ambient, diffuse, and specular terms to demonstrate phong shading, clearly visible objects, etc.). You don't need to show your code, just the results. Please consider providing a voiceover and/or overlaid text/arrows etc. **If it can't be seen on the demo, it can't be graded.**
4. A mandatory declaration that the work on both the programming and written assignments are 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, 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
- 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 sure to attend labs and ask the demonstrators for help.

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 winter wonderland visualization can be any kind you like, but the project must have the following basic functionality:

- 3-dimensional objects and views (note: fixed top-down orthographic view not allowed).
- User interaction and camera-control (already graded during mid-term)
  a. 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.
- A Hierarchical animated creature relevant to the theme (**already graded during mid-term**)
- A crowd of animated snow-people/reindeer/robins. The crowd can be implemented in any way that you like, boids can be implemented as an Advanced Feature and having a hierarchical character for each crowd member is also an Advanced Feature. At a minimum, the creatures should be translating around the scene, e.g., forwards and backwards. **(15% for crowd)**
- Texture-mapping your scene and creatures using an image file (e.g., jpg). (**approx. 15%)**
- Implementation of the Phong Illumination model. Specular highlights must be visible. **(approx. 15%)**
  a. Multiple light sources (at least 2, can be point, directional, or spotlight)
  b. Multiple different material properties (at least 5 on 5 different objects – e.g., you could have several crowd characters with different material properties)
  c. Normal must be transformed correctly
  d. Shading must use a combination of ambient, diffuse, and specular lighting

<u>This basic functionality is worth approximately 45% of the project mark</u>

The final 55% (approx.) will be given for advanced features, with approximately 15% per perfectly-executed feature – simple features will receive less, elaborate or well-designed features will receive more. You have some scope to put effort into your favourite aspects of 3D graphics, or experiment with new ideas.

<u>Advanced Features</u> can include the following, or indeed others that you think of:

- More complex boids/crowd animation
- Hierarchical creatures that move realistically in your crowd
- Advanced rendering/lighting effects
- Advanced animation
- Collision detection
- Simulated dynamics or physics of any sort
- Intelligent Characters
- Great models that you made yourself
- Height-mapped terrain
- Procedurally generated terrain or meshes
- 3D “picking” to select game objects with the mouse cursor

- More advanced texturing effects; multi-texturing, environment-mapping, bump-mapping, specular-mapping, etc.
- Partially transparent geometry using alpha-blending
- Smoke of fog
- On-screen control panels (using QT or other library)
- cube mapping for sky-boxes, environment mapping, or refraction
- ??? – your own feature ideas

Note: The [approximate] 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. You should aim to include bonus feature(s) and will be penalised if you have not attempted more than the basic functionality.