COSC363: Computer Graphics

Assignment 1

(Max. Marks = 20; Due date: 1 April 2021)

Synchronized Coupled Animation

1. Introduction

Synchronized coupled animations are formed by combining two or more different animation sequences in such a way that the objects move in a specific pattern producing an interesting and visually appealing motion. Examples of such animations can be found in the following YouTube videos:

https://www.youtube.com/watch?v=2zmkQr2j7z8

https://www.youtube.com/watch?v=V-cCrX7vw8Q

Several other sophisticated and more complex forms of synchronized animations can be found here:

https://youtu.be/SZtTupRKSKY

Examples of synchornized animations will also be shown during lectures when the assignment tasks are discussed.

2. Task Description

In this assignment, you will develop a graphics application using OpenGL-2 to display a scene, which among other things contains a synchronized animation model as the centerpiece. You are not required to create highly complex models or animation sequences for this assignment. The assignment task can be broadly divided into four sections:

Minimum requirements (10 marks): The scene should contain a set of minimum features as outlined in Section 3.

Extra features (5 marks): You can implement a set of extra features and gain up to a maximum of 5 marks. A few examples of such features are given in Section 4.

Rendering quality (3 marks): Up to 3 marks are allocated for the overall rendering quality of the scene, including animations (see Section 5).

Report (2 marks): You should also prepare a brief report containing a few screenshots and a description of the scene (see Section 6).

3. Minimum Requirements (max. 10 Marks)

3.1 (7 Marks) The scene should contain a few models that continuously move according to a synchronized animation sequence. The models could be constructed using a set of polygonal primitives, GLUT objects or surface design methods (see Section 7).

3.2 (1 Mark) At least one of the objects must be modelled as a sweep surface.

3.3 (1 Mark) The viewer must be able to navigate the camera through the scene using arrow keys. The up and down arrow keys must move the camera forward and backward respectively, the left arrow key should change the direction of motion towards left by a certain angle, and the right arrow key should change the direction of motion towards right. You may define additional functions for processing keyboard, special key, and mouse inputs as needed. Please do not use passive mouse motion function or the idle function for user interaction.

3.4 (1 Mark) At least three different textures must be used in the scene. It is not necessary to texture map all surfaces. Please do not use very large images (>10MB) as textures. The textures used in a skymap will be counted as one texture.

4. Extra Features (max. 5 Marks)

A list of possible features, and the approximate marks they would each gain if implemented correctly (subject to a maximum of 5 marks) are given below.

- Planar shadows cast by at least one object (1 mark)
- A spot light on a moving/rotating object (1 mark). The movement of the spotlight should be clearly visible.
- Physics based (e.g. gravity) animation. Give relevant equations in your report. (1-2 marks depending on the complexity of the Physics model)
- A surface shape generated using a mathematical formula (e.g. paraboloids) or a complex design process. Please provide sufficient details in the report. (1-3 marks depending on the complexity of the model)
- Collision detection (1 mark)
- Sky box (1 mark)
- Particle systems (1-2 marks depending on the complexity of the model)
- (This is not a program feature) You will be given an extra mark if you also include a proper CmakeLists.txt file that builds the executable from the source code, along with your submitted files.

The marks associated with each feature should be taken to be indicative of the time and/or effort required to implement that feature. The list given above should not be taken as the complete set of features that can be implemented.

5. Rendering Quality (max. 3 Marks)

In the context of this assignment, rendering quality refers to the visual aesthetic quality of the models and animations displayed by the program. For example, slow and jittery animations, incorrectly rendered sky boxes and shadows, and improper illumination may cause marks to be deducted for poor rendering quality.

6. Report (Max. marks: 2, Max. number of pages: 4)

The report should include the following sections:

- Your name and student number.
- A brief description of the scene.
- At least two screenshots showing important aspects of the scene or animations.
- A brief description of each extra feature implemented, including relevant equations. Please also include any diagrams, sketches etc. used in the design of the models.
- A full list of control functions (keyboard, mouse, special keys) defined for interacting with the scene.
- Build commands or instructions for compiling and running the program.
- References to sources of textures, algorithms, equations etc., used in your work. You need not include references to material provided in the course (lab/lecture material).

You may include more than 4 pages in the report, only if absolutely necessary. Please submit your report as a single **PDF** file.

7. Models and Animations

The object models in the scene may be constructed using any of the following methods:

- By transforming and combining a set of (at least three) GLUT/GLU objects such as spheres, cubes, cylinders etc. to form a composite model.
- By using a set of polygonal surfaces comprising of quads and triangles.
- By using surface generation methods such as sweep surfaces, surfaces of revolution etc.

You are not required to design or use highly complex mesh models for this assignment. Designing objects using modeling packages such as Max, Blender, Lightwave, etc., will not give you any extra points. Downloading pre-built mesh models from the Internet and using them in the program will also not give you any marks.

You may use parts of lab code and resources (models, images, image loading functions). Models and animations developed in the lab will give you marks only if significant changes or enhancements have been made to them.

8. Program Development:

- Please do not use source codes of programs from online repositories, tutorials, and books.
- Develop your program in C/C++ language using only OpenGL 2 API. Please do not use OpenGL 4 code (vertex/fragment shaders etc.), or extensions (e.g. ARB, EXT etc.) that are not part of the standard OpenGL API.
- **Please make sure that your program can be compiled and run on CSSE lab machines.**
- Your submission will not be marked for code readability or optimization. However, you are encouraged to prepare well documented and readable code.
- You will gain marks only for objects/features that are correctly and clearly displayed. A particular feature may be implemented correctly in code, but if its effect is not clearly seen on the screen, you will not receive marks for that feature.

9. Assignment Submission

Submit electronically (using *Learn*), the source code and all supplementary files (models, images etc.) needed to run the program. Please also submit your report in PDF format. The files should all be packaged together and submitted as a single .zip file.

This is not a group project. Your assignment submission must represent your own individual work. In particular, students are not permitted to share program source code in any way.

Important dates:

- Due date and time: 11:55pm, Wed, **1 April 2021**
- Dropdead date and time, with late penalty 7.5% (1.5 Marks): 11:55pm, Mon, 8 April 2021.

10. Miscellaneous

1. Check regularly on the *Learn* system forums for spec updates and clarifications.

2. Standard departmental regulations regarding dishonest practices apply.