KIT307/KIT608 Computer Graphics & Animation
Assignment 2: Animated Movie

Due Monday 16th October, 11:55pm

Introduction

For this assignment you will create a short animated movie. The assignment will assess your understanding of 3D programming and animation.

Task Outline

This assignment consists of four tasks. All students must complete tasks 1, 2, but may choose one of tasks 3 and 4 to complete. There will be no additional marks for completing both of tasks 3 and 4. Task 3 is intended for students who are more technically minded, while task 4 is for more creative students.

Task 1: Animated Movie 50%

For this task you will create a short animated movie. The movie should contain approximately 30 seconds of animation. Extended periods with little animation will not count towards the 30 seconds (if you need periods like this for dramatic effect, extend the overall length of the movie).

The specific requirements for the movie are:

- 30 seconds of animation
- Must tell a "story" (not just random animation)
- Must include one or more sweep objects. At least one sweep object must meet the same requirements as for Task 1 in Assignment 1, but may be a different object
- Must include one or more *animated* composites. At least one composite must meet the same requirements as for Task 2 in Assignment 1, but in addition must have individually animated components. Again, this may be a different object to that submitted for Assignment 1

Task 2: Textured Object 25%

For this task you must create a textured object. You may use any image for the texture, or create the image yourself. This object does not have to be included in the movie (for example, you may feel that the texture spoils the aesthetics of the movie). If the object is not included, make sure that this is clearly indicated to the marker.

The specific requirements are:

- The object must consist of more than six polygons that are not coplanar (i.e. more complex than a cube)
- Texture coordinates may not be calculated by any of the `calcUV*` methods currently implemented in the Mesh class or trivial derivatives of these methods. Coordinates may be calculated procedurally using an alternative method, or may be set manually
- The texture(s) must contain recognisable details, not just a plain colour, or "noise", or a repeating pattern

Task 3: Programming Extension 25%

For this task you will complete a significant programming extension. The specifics of your extension must be negotiated with the lecturer. Some examples of extensions that would be considered "significant" are:

- Development and use of multi-light pixel shaders
- Shadow implementation (e.g. drop shadows or shadow mapping shaders)
- Mesh deformation
- Animated textures
- Debugging tools
- Development tools (e.g. simple terrain editor or mesh loader)

Task 4: Extended Movie 25%

To be assessed for this task, your movie must contain 60 seconds of animation, plus meet the following *additional* requirements:

- Must include at least one additional *animated* composite. This composite must meet the same requirements as for **Task 5** in Assignment 1, but in addition must have individually animated components. Again, this may be a different object to that submitted for Assignment 1
- Must have a sound track

Application Setup

Your application should be setup to run the movie immediately. Any additional components that need to be assessed should be included in an additional application class with clear comments in the Main method showing how to run this application (and comments explaining what additional items are included in this application for assessment – e.g. textured object, programming extension).

Do not neglect the camera setup. The default camera settings are not likely to be the best for your scene. In particular, choose a near clip plane as far from the camera as possible. Please disable the `KeyboardController` – it is a movie, not an interactive animation.

Please stop all animation (including camera movement) at the end of your movie, and preferably do something to clearly indicate that the movie has completed (e.g. credits).

Submission Process

Please follow the submission process below very carefully, it is your responsibility to ensure that the correct files (and only the correct files) are submitted in working order. The marker will have very limited time to correct any errors. If the project cannot be made to run correctly in a reasonable amount of time, your assignment will not be marked.

- Remove any unused files from the resources folder
- Check that the project still runs as expected
- In Visual Studio, select "Clean Solution" from the build menu. This is to avoid uploading object files and to ensure a clean build when the marker runs your program
- Check that the project still runs as expected
- Choose "Clean Solution" from the build menu again. Do not run your program again after this stage
- Quit Visual Studio and zip your assignment folder
- Fill out an assignment cover sheet (electronically)
- Submit the zip and the cover sheet via MyLO
- **After submitting, unzip your submission to a new location and quadruple check that it works!**
-

Assignment Individuality

You are encouraged to use relevant code from the T3D framework, lectures or tutorials, as the starting point for any new classes that you create, but must clearly acknowledge the origin of such code (in the comments). Similarly, if you find code-snippets or other helpful information on-line, this also needs to be acknowledged in the comments.

However, this is an individual assignment, which should otherwise be your own individual work. It is expected that you will work with and get (limited) help from other students, but may not use any part of their code or the code of previous students in your solution.

Marking

Synopsis of the task and its context

The assignment is worth 25% of the unit mark, and marked out of 100.

All testing of code will be done using Visual Studio on Windows as provided in the School labs.

Your code will be tested both by running what you submit and by viewing your code. Your code may also be run using separate test code.

Match between learning outcomes and assessment criteria for the task:

Unit learning outcomes	
On successful completion of this unit, you will be able to:	Task criteria
1. Demonstrate techniques for generating, storing, displaying and manipulating 2D and 3D graphical data;	All
2. Identify and analyse trends in graphics hardware and software, and assess the likely impact of future developments in this area;	-
3. Implement low-level graphics code to create and animate 2D and 3D graphics.	All
4. Use creative, computational and critical thinking in the selection and implementation of graphics and animation techniques	All
5. Contribute to an open source community, in accordance with industry best practice.	-

Criteria	HD (High Distinction)	DN (Distinction)	CR (Credit)	PP (Pass)	NN (Fail)
	You have:	**You have:**	**You have:**	**You have:**	**You have:**
1. Animated Movie ***Weighting 50%***	Developed an animated movie that: • Has at least 30 seconds of animation • Contains a sweep object that meets requirements • Contains a composite with individually animated components that meets requirements • Tells a story	Developed an animated movie that contains a sweep and composite that meet requirements and tells a story, but only 15- 30 seconds of animation	Developed a movie that tells a story and has 15+ seconds of animation, but is missing either a suitable sweep or composite	Developed a movie that has 15+ seconds of animation, but is missing a suitable sweep and composite	Developed a movie that contains less than 15 seconds of meaningful animation
2. Textured Object ***Weighting 25%***	Constructed a textured object that: • Has more than 6 non-coplanar polygons • Has UV coordinated calculated procedurally or manually. The chosen method must be mostly original code. • Uses a texture with sufficient detail to determine if UV coordinated are calculated correctly	Constructed a textured object that meets 2 of the three HD requirements			Constructed a textured object that meets less than 2 of the HD requirements
3. Programming Extension ***Weighting 25%***	• Correctly implemented a significant programming extension • The extension is implemented in a way that complements the design of T3D	Correctly implemented a significant programming extension	Partially implemented a significant extension, or implemented a significant extension with minor errors, or implemented a minor extension correctly		Made some progress on a significant extension
4. Extended Movie ***Weighting 25%***	Extended your animated movie so that it: • Has at least 60 seconds of animation • Contains at least one advanced composite with individually animated components • Tells a compelling story • Is aesthetically pleasing • Has a sound track	Extended your animated movie so that it: • Has at least 60 seconds of animation • Contains at least one advanced composite with individually animated components • Tells a story • Is aesthetically pleasing • Has a sound track	Extended your animated movie so that it: • Has at least 60 seconds of animation • Contains at least one advanced composite with individually animated components • Tells a story • Has a sound track		Made a movie that contains more than 60 seconds of animation and tells a story and has a sound track