CMSC 427 Computer Graphics
Programming Assignment 4: Materials, Texture, and Transformations
Assignment Release Date: April 5, 2017
Due Date: **Saturday, April 22, 2017 at 11:59pm**
Project Submission:

1) Delete all intermediate files (run the command make distclean) and the Makefile created during the compilation process. Also delete the executable file.
2) Delete all of the object models included in the original starter code.
3) Place all files for the project in a folder and ZIP up the folder. You will submit your project via the submit server. To submit a zip file, login to the submit server webpage and look for the link to make a *web submission*.

Assignment Description

This assignment has two parts. A written part, in which you must answer questions about the starter code. The written part counts toward your written assignment grade and this programming assignment counts toward your project grade. Note that the written questions are in a separate assignment on the course web page and are due earlier than this programming portion. For each of the features below, you are to include screen captures demonstrating your implementation of each operation with before and after. Make sure your write-up includes a concise but accurate technical description of what you implemented. The TA should be confident of your implementation based on your report and should be able to verify it quickly. In addition, for each operation you will use a single mesh Car.obj mesh in the subfolder Mercedes. You may make hard references in your code to geometry in this mesh. We will only use this mesh for testing.

Implementation

- Camera Parameter Animations (implement all)
  - Smoothly animate FOV parameter between 20 and 100 degrees, explain what aspect of the camera is changing in your documentation file.
  - Smoothly animate the near plane between 1 and 500, explain what you see in your documentation file.
  - Smoothly animate the far plane between 500 and 1000, explain what you see in your documentation file.
  - Smoothly animate the camera position about the look center in the look up direction. Use quaternions for the rotation.

- Materials (implement all)
  - Cycle through each material, drawing only triangles from one material at a time. Print the material name in a message box.
  - Animate the triangles for the selected material so they move up out of the plane of the car and back down. Note, the rest of the car must be displayed at this time.

- Groups (implement all, also consider extra credit)
  - Cycle through each group in the obj file, drawing only triangles from one group at a time. Print the group name in a message box.
  - Animate each of the wheels so they rotate as if the car is moving forward. HINT: You will need to use the previous task to find the group of triangles that correspond to the spoked wheel. You can hard code the object names in your implementation. Note also that the wheel is in the coordinate frame of the car, not its own centered coordinate frame.
  - ***Extra Credit (up to 10%):*** Animate the front wheels of the car so they move left and right as if the car is swerving left and right. The wheels must move left and right while they are also rotating, like a real car. In addition animate the car so that it moves as it swerves in a figure

eight pattern. HINT: You will need to have this transformation operate on more than one object. You can hard code the object names in your implementation.

Project Grading

There are 2 parts that will be graded. 1) The write-up and 2) your code. The write-up should include screenshots demonstrating each operation you chose to implement. There should be 2 screenshots per operation – one before and one after applying the operation to the mesh. Screenshots are required for all operations. The TA will use an undisclosed set of meshes to test your implementation as well so we encourage you to find more meshes online to test your program. We have provided you with several meshes to start off with.

Your code should be well commented in header files and source files. Note that it will also be judged subjectively based on the simplicity and clarity of implementation. An implementation that is easy to understand, but has few minor bugs will be scored higher than a messy implementation with the same number of minor bugs.

Documentation Grading

Please include documentation in the form of a Markdown (https://github.com/adam-p/markdown-here/wiki/Markdown-Cheatsheet) document with your assignment. It should be a text file in the folder ./cmsc427 called DOCUMENTATION.md. You will be graded on the completeness, clarity, and conciseness of your documentation. Your documentation should links to inline images showing the before and after of your image processing operations. It should also include an explanation of the algorithm you implemented. The more complete and clear your explanation is in this documentation the easier it will be for us to assign partial credit. It is in your interest to be as concise and clear as possible. Please make sure you only use relative links in your mark down file. If the TA has difficulty finding your images and the links, you will lose points. It is additionally suggested that you convert your Markdown file to .PDF so the TA.

Additional Submission Details

- Exactly one person of your team should be responsible for the submission. The latest submission will be the one your team be graded on.
- Your document should be properly written with markdown notations, and with correct extension names: it should be either .md or .markdown, not .txt. (If you prefer an extended version of Markdown (e.g. GitHub flavored Markdown or Pandoc's Markdown), it is also fine if you have included your PDF file.)
- It's recommended (required if you are using extended version of Markdown) to generate a PDF file for your markdown document. and put it under the same directory with same basename (i.e. DOCUMENTATION.pdf), take a look at it to make sure the layout is intended and screenshots are properly included. If the PDF file is missing, your document will be converted from your DOCUMENTATION.md or DOCUMENTATION.markdown to a PDF file using pandoc before grading.
- Most online markdown editors would have the option to export your document as PDF files, and there are tools that do this Markdown-PDF conversion online.
- It's not recommended to include your images as website URLs or encode your images into URLs. and do not use local or private URLs otherwise I can't find your screenshots. You should include your images as separated files in your submission.
- Make sure to crop your screenshot properly, it should be enough to show your work, but not too much information (e.g. another screen with youtube or twitch opened).
- In your submission there should be just project files, screenshots, documents and other things as requested. Top-level zip files are fine, but do not add any other level of archive in your submission, which means things like *.zip, *.7z are not recommended, compressing data more than once barely gives you a smaller file anyways.

- For your project directory, it's recommended to use just a combination of alphabetic and numeric characters and underlines (or in other words, your project directory name should be an ASCII string accepted by regular expression \w+), otherwise qmake would have some trouble with project path.
- Your source code should be compatible with C++11 standard and cross-platform:
    - one way to make your compiler keep an eye on this is to ensure c++11 appears in the *.pro file of your project in CONFIG += ... line (which it should have been if you download the latest starter project). also try to address compiler warnings instead of suppressing them.
    - be consistent with file names, which means if you name one of your header Foo.h, you should write your include directive like #include "Foo.h", not #include "foo.h" or #include "FOO.H"