LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

INSTITUT FÜR INFORMATIK
MEDIENINFORMATIK
PROF. DR. ANDREAS BUTZ, CHANGKUN OU, DAVID ENGLMEIER
COMPUTERGRAFIK 1, SOMMERSEMESTER 2020

Online-Hausarbeit 2: The Bread and Butter Stuff

Bearbeitungszeitraum: 13.07.2020 00:00 Uhr - 18.07.2020 23:59Uhr

General Informations

- There are three graded assignments allowing for a total score of 200 regular points and 20 bonus points. You will need 100 points to pass with 4.0, every additional 10 points increase the grade to the next step, while a score of 190 points or greater results in 1.0.
- It is prohibited to exchange solutions for the graded assignments with other students during the examination period. You must work on the graded assignments alone and independently and submit your own solution. If we discover any fraud or plagiarism in the submission, both parties will be excluded from the course.
- For programming tasks, you must organize and comment your code well by following additional comments/instructions in the code skeleton. Weird coding style or changing the provided template may not be well tested or graded. In addition, any comments in your code must be written in English.
- For non-programming tasks, you can answer in German or English or mix.
- If you have any questions regarding technical issues, please contact the course assistants immediately.

1 Overview

This is the second graded assignment for the CG1. You can achieve a maximum of 55 points from this graded assignment. The estimated completion time for this assignment is about 1 to 2 hours. Depending on your familiarity with the subject, your time might increase accordingly. If you struggle with one task, get yourself some fresh air and come back to it later. To reduce your time pressure, you can submit your work before 18.07.2020 23:59.

For our own information (and without any influence on the grade!), please roughly record the actual time (in hours) you spent on this assignment and let us know on the last page below your signature.

The aim of this assignment is to help those of you who do not like programming to still achieve a certain grade by demonstrating how they recognize and can apply concepts from class to examples and explain how and why, including transformation, camera, local illumination, rendering methods, and the OpenGL graphics pipeline.

Let's get started.

2 Warmup (3 Points)

If you look at the three spheres in Figure 1, you should immediately recognize with which family of illumination models they were generated. We have discussed 3 variations of this.

a) Please identify from left to right, which variation was used and tell us how you recognized this from the image. (3p)

Figure 1: The three spheres.

3 See your world with Phong's eyes (5 Points)

In class, we have discussed the Phong's illumination model

$$L_{\text{Phong}} = L_a + L_d + L_s = k_a I_a + k_d I_d \max(0, \mathbf{N} \cdot \mathbf{L}) + k_s I_s \max(0, \mathbf{R} \cdot \mathbf{V})^p$$

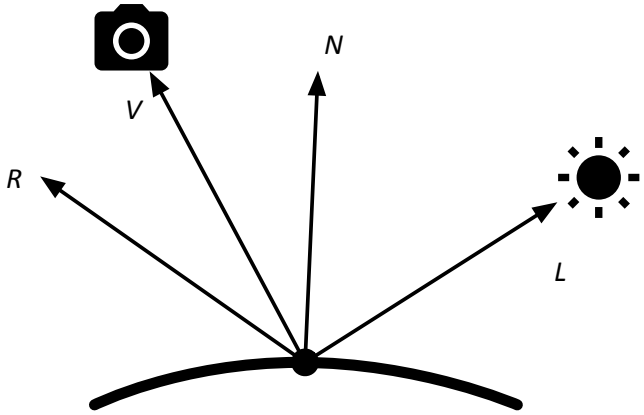

Figure 2: The Phong's illumination model.

with its three components for ambient, diffuse and specular reflection.

a) Look around your desk area, tell us 3 materials that you think can be well reproduced using this model, and give an estimate for the respective coefficients k_a, k_d, k_s on a scale from 0 to 1 (3p).

b) Then, pick 2 materials you think can not be described by this model and tell us why (2p).

4 Choosing Rendering Methods (5 Points)

Figure 3 illustrates a photo of a teapot in a box with colored walls. This is a physical mockup of a standard CG example, the so-called "Cornell Box".

a) Please identify 4 illumination effects in the photo and discuss, with which rendering approach they could be recreated and why (4p).

b) The light source on the box covers a larger area. What would change in the photo if it was a point light source, and why (1p)?

Figure 3: A photo of a teapot

5 Transformations (5 Points)

Why is it a good idea to use homogeneous coordinates? They require more memory, matrices are larger, more operations are needed to multiply a vector with a matrix in homogenous coordinates, So:

a) What is the advantage? (1p)

b) Give us an example of a scene graph, in which regular 3D coordinates would be more efficient (1p),

c) and another one, in which the advantage of homogeneous coordinates is big (1p),

d) and another one, in both cases, explain why (2p).

6 What's left if nothing's right? (8 Points)

Let's assume you have built a polygon model of a right hand at the position $(0, 0, -10)$. Your camera is at the position $(0, 0, 0)$ and looks along the negative Z axis, as discussed in class. You use one of the approaches from Section 2 to generate the image in Figure 4.

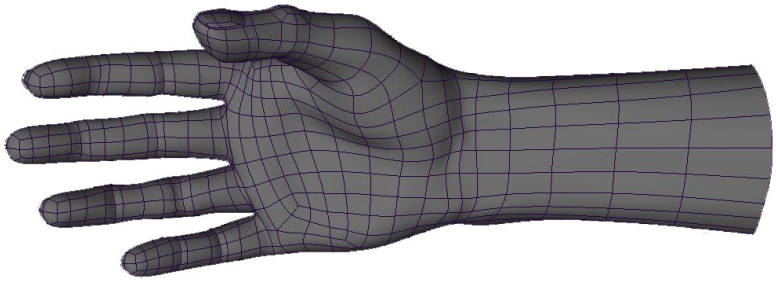

Figure 4: A polygon model of a right hand.

Now you transform your model of a right hand with the following matrix:

$$\begin{pmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

a) What do you see in the image and from which side (3p)?

b) Why is this probably not what you wanted (1p)?

c) What goes wrong (1p)?

d) Does this have to do with an optimization method for the camera? Which and how (1p)?

e) Is there a way to fix this using transformations only (1p)?

f) Is there a different way to fix this (1p)?

7 The automatic camera (eliminating human error ;-) (14 Points)

In photography, many things have been made automatic (exposure and brightness, focus, depth of field, ...). One reason why there are still bad photos around, is the human who is pointing the camera the wrong way and, for example, cuts off legs, makes the horizon tilted, or adds unnecessary space around a person. In computer graphics, we can finally get rid of this last source of errors (the human) and build a camera that always shows the entire scene filling the entire frame, and with a perfectly level horizon. Imagine you would like to program such a camera. In the exercises, you have already learned about a very helpful data structure (AABB) to make your calculations simple. You can assume that such an AABB is precomputed for the group "world" of all objects in the scene and it has the

boundaries $(x_{min}, x_{max}, y_{min}, y_{max}, z_{min}, z_{max})$. Let's assume a human left the camera somewhere in space and showing a badly framed picture. Let's call the camera position (x_c, y_c, z_c).

a) List the steps and calculations you need to do to view the entire scene centered in the image with the camera looking along the negative Z axis and a level horizon (5p).

b) Assume you want to remain on the same side of the scene instead of turning the camera to point along -Z. You want to still center the entire scene and have it roughly fill the frame with a level horizon. What do you do in this case? Is the AABB still the best possible data structure? What alternative would you ask for to make your life easier? (6p)

c) In this second case above, getting the horizon level will fail in two specific cases. Why is this, and what is the concept discussed in the exercises which relates to it? (3p)

8 A shady place (10 Points)

You want to recreate the scene shown in Figure 5 using the programmable OpenGL shader pipeline as performance-efficiently (!) as possible so that it may be used for large virtual environments. The circles in the image highlight important features of the scene.

a) How would you utilize the four categories of shaders in the center of the OpenGL pipeline for rendering the scene in the most efficient way? Find a task for each shader. Justify your 4 answers and (if applicable) link them to the highlighted elements in the figure and some specific details that construct the scene (8p).

b) Let's say you want to add performance-efficient physical interaction with the grass blades (for example a character that walks the scene). Which shader would you use, and why? (2p).

Figure 5: A rendered scene.

9 Submission Instructions (Bonus: 5 Points)

Make sure that your submission is complete. In particular, the signed "Erklärung über die eigenständige Bearbeitung" must be included in your submission. A submission without signature will not be graded, and a re-submission is NOT possible. Your signature must be a non-editable element in the document. Specifically, you can either print the document, sign then rescan the document as a PDF file that replaces this document with your hand signature; Or, you can also add your electronic signature in this document by using the Preview app on macOS or Adobe Acrobat Reader DC on Windows (Linux user can find their own tools). Following the additional submission instructions below will reward you with 5 bonus points.

- Use the `ANSWERS.md` file to fill your answers (1p) and use Markdown syntax to structure your answer (1p) for bullet points and ect.
- Use LaTeX-syntax to insert mathematical formula in the `ANSWERS.md` file (2p). If you are not familiar regarding how to use LaTeX-syntax in a Markdown file, you can find two quick guides ("Quick Start" and "Markdown Reference") in the Help menu of the typora editor.
- Rename your folder name to `abgabe2-<your matriculation number>`. Then submit your solution as a `.zip` file via Uni2Work (1p).

As an example, assume your matriculation number is 12345678, then your ZIP filename should be `abgabe2-12345678.zip`, and the decompressed folder structure should look exactly like this:

```
abgabe2-12345678/
├── ANSWERS.md
└── README.pdf
```

Erklärung über die eigenständige Bearbeitung

Online-Hausarbeit 2

Ich erkläre hiermit, dass ich die vorliegende Arbeit vollständig selbstständig angefertigt habe. Quellen und Hilfsmittel über den Rahmen der Vorlesungen/Übungen hinaus sind als solche markiert und angegeben. Ich bin mir darüber im Klaren, dass Verstöße durch Plagiate oder Zusammenarbeit mit Dritten zum Ausschluss von der Veranstaltung führen.

Ort, Datum Unterschrift

Ich habe insgesamt etwa _____ Stunden Arbeitszeit für diese Abgabe aufgewendet. (Diese Information ist freiwillig, rein informativ und hat keinen Einfluss auf die Benotung.)