UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA
Denkleiers • Leading Minds • Dikgopolo tša Dihlalefi

Department of Computer Science
Faculty of Engineering, Built Environment & IT
University of Pretoria

COS344 - Computer Graphics

Practical 4 Specification: Translucency and Light

Release Date: 17-02-2024 at 06:00

Start By Date: 07-04-2024

Due Date: 28-04-2024 at 10:00

Total Marks: 48

Contents

1 General Instructions

- ***Read the entire assignment thoroughly before you start coding.***
- This assignment should be completed individually, no group effort is allowed.
- **To prevent plagiarism, every submission will be inspected with the help of dedicated software.**
- Be ready to upload your assignment well before the deadline, as **no extension will be granted.**
- If your code does not compile, you will be awarded a mark of 0. The rendering output of your program will be primarily considered for marks, although internal structure may also be tested (eg. the presence/absence of certain functions or classes).
- Failure of your program to successfully exit will result in a mark of 0.
- Note that plagiarism is considered a very serious offence. Plagiarism will not be tolerated, and disciplinary action will be taken against offending students. Please refer to the University of Pretoria's plagiarism page at http://www.ais.up.ac.za/plagiarism/index.htm.
- You are allowed to use any standard of C++.
- The usage of ChatGPT and other AI-Related software is strictly forbidden and will be considered as plagiarism.
- No pre-build objects and textures may be used. All objects and textures that you need to use must be created by yourself.
- You must use OpenGL version 3.3 for this practical.

2 Overview

For this practical, you will need to render a semi-translucent glass wall with a light shining through it with the light also being projected onto a sheet. You will also be able to see what effect the number of vertices has on the shading and performance of your rendering.

3 Your Task:

For this practical, you will need to render the following scene depicted in Figure 1: The **red sheet** is the background sheet onto which the light will be projected on and will be referred to as the `backsheet`. The **green block** is a semi-translucent block that should be positioned in the middle between the light and the `backsheet`. The `green block` will be referred to as the `glass`. The **blue dot** represents the point-like light source and is referred to as the `light`. The **grey sheet** below will be referred to as the `bottomsheet`.

In the sections below the different requirements are laid out.

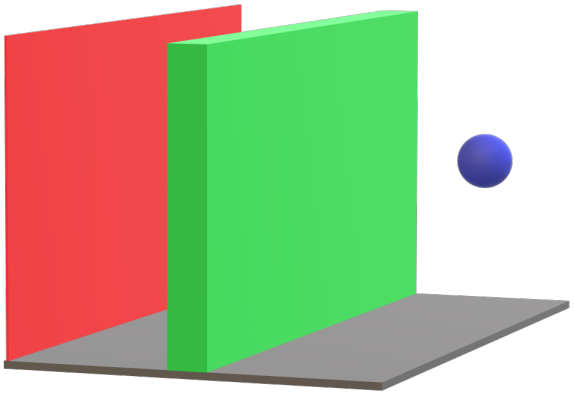

Figure 1: Depiction of Render

3.1 Shape requirements

Your rendering should contain the following shapes:

- Backsheet
  - A configurable number of smaller rectangles which should represent a solid wall.
  - Due to rounding errors with the low-level representation of numerical data, at a certain angle you will be able to see through small slits in the wall. This will be taken into account when marking, you will not lose marks because of the rounding errors.
  - The number of rectangles will be specified as a command line argument.
- Glass
  - The glass should be a semi-translucent box that has the same height and length as the backsheet.
  - The faces of the glass should be rendered in the correct order such that a the glass appears translucent.
  - It can be assumed that the light does not reflect off of the glass but instead just shines through it.
  - The glass should not have a thickness of more than 0.2 units within the default clipping window.
  - The glass should be placed in the middle between the light and the backsheet.
- Bottomsheet
  - This is a simple rectangle that should span all the way from the backsheet, under the glass and to the same Z plane as the light's initial position.
  - The bottomsheet should have the same length as the backsheet and glass.

3.2 Colour requirements

You will need to comply with the following color requirements:

- Backsheet
  - Assign two keys on your keyboard (of your choosing) to control the colour of the backsheet.
  - Using these keys, you will need to be able to cycle through the following colours:
    - Red
    - Green
    - Blue
    - White
    - Black
    - 5 other distinct colours of your choosing.
- Glass
  - Assign two different keys on your keyboard (of your choosing) to control the colour of the glass.
  - Using these keys, you will need to be able to cycle through the following colours:
    - Red
    - Green
    - Blue
    - White
    - Black
    - 5 other distinct colours of your choosing.
  - Assign two further keys to control the alpha value of the glass.
  - Make use of RGBA and assign the colour buffer to take in RGBA colours[1] instead of the standard RGB[2] colours.
- Light
  - Assign two different keys on your keyboard (of your choosing) to control the colour of the glass.
  - Using these keys, you will need to be able to cycle through the following colours:
    - Red
    - Green
    - Blue
    - White
    - 5 other distinct colours of your choosing.

[1] A vec4 value.
[2] A vec3 value.

  - Note the glass colour should effect the backsheet's colour when looking through the glass.
  - The glass should also affect the colour of the light that is projected onto the backsheet.

- Bottomsheet
  - This sheet needs to be a standard grey colour and should not change by the light[3]

3.3 Light requirements

Your light should have the following requirements:

- The light should be a point light.
- You should at least make use of the pseudo code described in Chapter 4 of the prescribed textbook to calculate the lighting for a point source.
- The lights projection will only be displayed in the backsheet.

3.4 Transformation requirements

In this section, the transformation or animation requirements for your rendering are discussed. Keep the animation rates small such that it allows you to press the key multiple times before going out of camera view.

3.4.1 Rotations

- When the `W key` is pressed, the scene (including the light) needs to rotate about x-axis in an anti-clockwise direction to the positive x-axis.
- When the `S key` is pressed, the scene (including the light) needs to rotate about x-axis in a clockwise direction to the positive x-axis.
- When the `A key` is pressed, the scene (including the light) needs to rotate about y-axis in an anti-clockwise direction to the positive y-axis.
- When the `D key` is pressed, the scene (including the light) needs to rotate about y-axis in a clockwise direction to the positive y-axis.
- When the `E key` is pressed, the scene (including the light) needs to rotate about z-axis in an anti-clockwise direction to the positive z-axis.
- When the `Q key` is pressed, the scene (including the light) needs to rotate about z-axis in a clockwise direction to the positive z-axis.

[3]This is done to not overwork lower end computers.

3.4.2 Translation

- When the `up arrow key` is pressed, the light's y-axis position needs to move up the backsheet.
- When the `down arrow key` is pressed, the light's y-axis position needs to move down the backsheet.
- When the `left arrow key` is pressed, the lights's x-axis position needs to move left across the backsheet.
- When the `right arrow key` is pressed, the light's x-axis position needs to move right across the backsheet.
- When the `m key` is pressed, the light needs to move closer to the backsheet.
- When the `n key` is pressed, the light needs to move further away from the backsheet.

Note, through all the transformations of the light, the basic scene setup should remain still, apart from the position of the light.

4 Examples

NB: The glass in the example is "floating" for the sake of demonstrating how the light affects the background

The following is an example of what the scene should look like given that the glass is green, the background is blue and the light is white: Figure 2 has the light far away from the scene and has

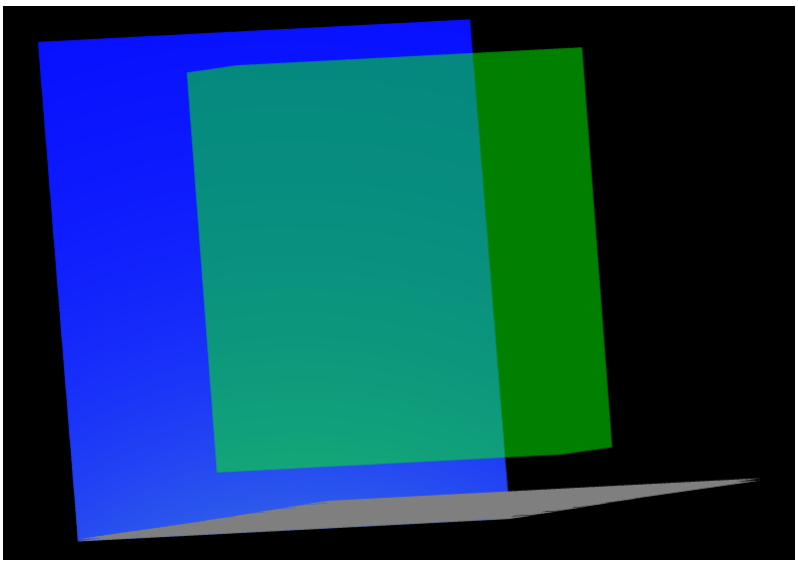

Figure 2: Light is far away from the scene

minimal visual effect on the backsheet, while Figure 3 the light is closer and creates a circle effect on the backsheet.

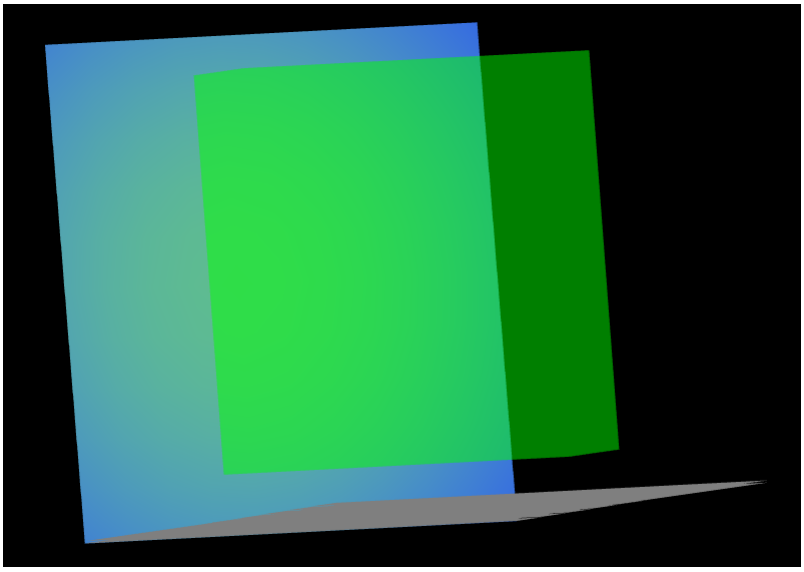

Figure 3: Light is shining onto the scene

5 Marking rubric

The following rubric will be used to mark your submitted assignment. Note you will be demoing the practical during the practical sessions on your own computer or a lab computer. Please see Table 1 for the rubric. Note: 1 mark will be subtracted for each transformation if the render is moved back to the center before a new transformation is applied.

6 Bonus marks

There are 10 bonus marks available. Things that can be done for bonus marks are:

- To sort your glass faces use the Binary Search Tree Partitioning algorithm (5 marks).
- Add 5 different types of material finishes to the backsheet (1 mark for each material finish).
- Adding a second light (2 marks).
- Adding edges (with glLines) to the glass such that each face can be easily distinguished (2 marks).
- Allowing the light to also affect the glass causes a spectral lighting effect on the glass (5 marks).
- Adding stained glass regions to the glass. For you to obtain your marks you will need to add ray tracing such that the light that intersects the different regions affects the light's projection on the backsheet. For each new region you correctly add you will get 2 marks.

7 Implementation Details

- You need to use OpenGL version 3.3 for this practical.

- You may **not** use any of the build-in mathematical libraries within the `glm` package. This included matrix arithmetic. *Hint: You may use your practical 1 in this practical.*
- You may **not** use any of the built-in OpenGL functions to generate the shapes for you. You need to create each shape from first principles.
- You may **not** use any of the built-in OpenGL functions to perform the transformations of the shapes. You need to transform each shape from first principles either explicitly or by using the matrix arithmetic techniques discussed in class.
- You may only use the following C++ and OpenGL libraries:
  - stdio.h
  - stdlib.h
  - iostream
  - iomanip
  - cmath
  - sstream
  - GL/glew.h
  - GLFW/glfw3.h
  - glm/glm.hpp

  You may also use the `shader.hpp` and `glad.c` files that assist with compiling and linking of shaders. Your code should be able to be compiled without the assistance of an IDE, i.e. the project needs to be able to be compiled from terminal.
- All your helper classes and files needs to be in the same directory of the main.cpp.
- Ensure that the title of the window of your program is your correct student number.

8 Submission

You are required to submit on ClickUp under the appropriate submission link. In the archive that you submit, include a makefile and compiling instructions such that the program can be compiled and executed by the markers if needed. *Failure to upload to ClickUP will result in you forfeiting all marks for this practical.* No exceptions will be made on this matter.

9 Demo Instructions

1. You will first be required to download your submission from ClickUP.
2. You will then demo your practical to the tutor.
3. In the presence of the tutor, you will be required to upload the archive you downloaded from ClickUP to FitchFork. *Failure to upload to FitchFork will result in you forfeiting all marks for this practical.* No exceptions will be made on this matter.

Assessment Criteria	0	1	2	3	4
Shape requirements (10 marks)					
Backsheet	Not configurable number of rectangles		Configurable number of rectangles but rendered incorrectly		Configurable number of rectangles and rendered correctly
Glass: Shape	Not rendered correctly		Box like shape but faces rendered in the incorrect order		Box like shape and faces are rendered in the correct order
Glass: Position	Not positioned correctly	Positioned correctly			
Bottomsheet	Not rendered correctly	Rendered correctly			
Colour requirements (18 marks)					
Backsheet	Single non changeable colour	At least 5 of the specified colours and controllable	All 10 specified colours and controllable		
Glass: Colours	Single non changeable colour	At least 5 of the specified colours and controllable	All 10 specified colours and controllable		
Glass: Alpha value	Alpha values not changeable or present		Alpha value is changeable and has visible effect		
Glass: Effects the backsheet	Glass has no effect on backsheet colour				Glass has an effect on the backsheet colour
Light: Colours	Single non changeable colour		At least 5 of the specified colours and controllable		All 9 specified colours and controllable
Light: Effect on backsheet	Light has no effect on backsheet colour		Light has effect on backsheet colour but ignores the glass colour		Light has effect on backsheet colour and incorporates the glass colour
Light requirements (12 marks)					
Lighting type	There is no lighting effect	There is a light but it is the wrong type (3 marks)			Correct point light (12 marks)
Transformation requirements (8)					
Rotations	No rotation		Entire scene rotates around at least one axis	Entire scene rotates around at least two axis	Entire scene rotates around three axis
Translation (light effect)	No light translations	Light translates but not in the correct directions relative to the scene	Light translates correctly along only one axis relative to the scene.	Light translates correctly along only two axis relative to the scene.	Light translates correctly along only thee axis relative to the scene.

Table 1: Marking rubric