

Belegarbeit
"Sunset Pagoda"

Burenko Anton
s76905

Prof. Dr. Wolfgang Oertel
Computergrafik 1

Wintersemester 2018/19

Inhaltsverzeichnis

1	Aufgabenstellung	3
2	Bedienungsanleitung	4
3	Functionen	4
4	Viewports	6
4.1	Viewport von Sonne	6
4.2	Einstellbares Viewport	6
5	Figure	7
5.1	Würfel	7
5.2	Piramide	7
5.3	Pyramidenstumpf	7
6	Texture	8
7	Light	9
7.1	Ambiente Licht	9
7.2	Spotlicht	9
7.3	Richtungslicht	10
8	Animation	11
9	Interaction	12
10	Quellenverzeichnis	12

1 Aufgabenstellung

Schreiben Sie ein Programm in C/C++, das unter Verwendung von OpenGL, Vertex- und Fragment-Shadern folgende Aufgaben realisiert.

Aufgabe 1:

Geometrische Objekte: Erzeugen Sie eine interaktive zeitlich animierte Szene mit mehreren unterschiedlichen farblichen und texturierten dreidimensionalen geometrischen Objekten.

Aufgabe 2:

Beleuchtung: Beleuchten Sie die Szene mit mehreren Lichtquellen so, dass auf den Objekten unterschiedliche Beleuchtungseffekte sichtbar werden.

Aufgabe 3:

Ansicht: Stellen Sie die Szene gleichzeitig in verschiedenen Ansichten und Projektionen in mehreren Viewports des Anzeigefensters dar.

Aufgabe 4:

Programm: Stellen Sie das komplette Programm in Quelltextform als Visual-Studio-C++ - Projekt und in ausführbarer Form als exe-File derart bereit, dass die Lauffähigkeit auf den Computern des Praktikumlabor der Lehrveranstaltung gewährleistet ist.

Aufgabe 5:

Dokumentation: Fertigen Sie eine Systemdokumentation in Form eines pdf-Dokumentes von etwa 10 Seiten an, die Deckblatt, Gliederung, Aufgabenbeschreibung, Lösungsansatz, Lösungsumsetzung, Installations- und Bedienungsanleitung, einige Bildschirm-Snapshots, Probleme, Ergebnisse, Literatur- und Quellenverzeichnis enthält.

Aufgabe 6:

Abgabe: Demonstrieren Sie die Ergebnisse der Aufgaben 4 und 5 an einem Computer des Praktikumlabor der Lehrveranstaltung und übergeben Sie diese in einem Verzeichnis "*Name_Vorname_Bibliotheksnnummer*".

Zeitplan:

Die Ausgabe der Aufgabenstellung erfolgt zu Beginn der Lehrveranstaltungszeit. Die Abgabe der Ergebnisse erfolgt spätestens zum Ende der Lehrveranstaltungszeit.

2 Bedienungsanleitung

Das Programm verwendet OpenGL und Visual Studio, um die Szene zu kompilieren und auszuführen. Die Hauptbibliothek dafür sind opengl32, freeglut, glew32, FreeImage. Die müssen vorher durch Linker und C++ - compiler gebunden sein.

Die Texturen und alle .cpp sowie Headers und Shaders Dateien müssen in dem Verzeichnis "Programme" gespeichert werden. Welche Texturen es gibt kann man im Punkt "Texture" detailliert lesen.

3 Funktionen

main

int main(int argc, char argv)**

initialisiert das Program, bestimmt Window Size, Namen, und bindet Funktionen für Aufruf(z.B. keyboard, display).

generateBox24

void generateBox24()

generate Positionen, Farben, Normalen und Texturekoordinaten für einen Würfel. Außerdem speichert die für VAO und Position Buffers.

drawBox24

void drawBox24()

zeichnet den Würfel von gegebenen Positionen.

generatePiramideExTop

void generatePiramideExTop()

generate Positionen, Farben, Normalen und Texturekoordinaten für einen Dachobject. Außerdem speichert die für VAO und Position Buffers.

drawPiramideExTop

void drawPiramideExTop()

zeichnet das Dachobject von gegebenen Positionen.

generatePiramide

void generatePiramide()

generate Positionen, Farben, Normalen und Texturekoordinaten für eine Piramide. Außerdem speichert die für VAO und Position Buffers.

drawPiramide()

void drawPiramide()

zeichnet eine Piramide von gegebenen Positionen.

init

void init()

initialisiert Shaders, Texturen, Tiefenpuffer, ruft alle generate* Funktionen und alle Locations für *uniforms*

lichtBerechnung

void lichtBerechnung()

berechnet ModelViewProjection, ModelView und NormalMatrix und übergibt die Variablen als *uniforms*.

drawFigures

void drawFigures()

wählt richtige Texturen und zeichnet alle Objecte der Szene.

display

void display()

stellt View, Viewports bzw. drawFigures

keyboard

void keyboard(unsigned char theKey, int mouseX, int mouseY)

implementiert keyboard interaction(siehe Interaction).

mouse

void mouse(int mouseX, int mouseY)

implementiert mouse interaction(siehe Interaction).

4 Viewports

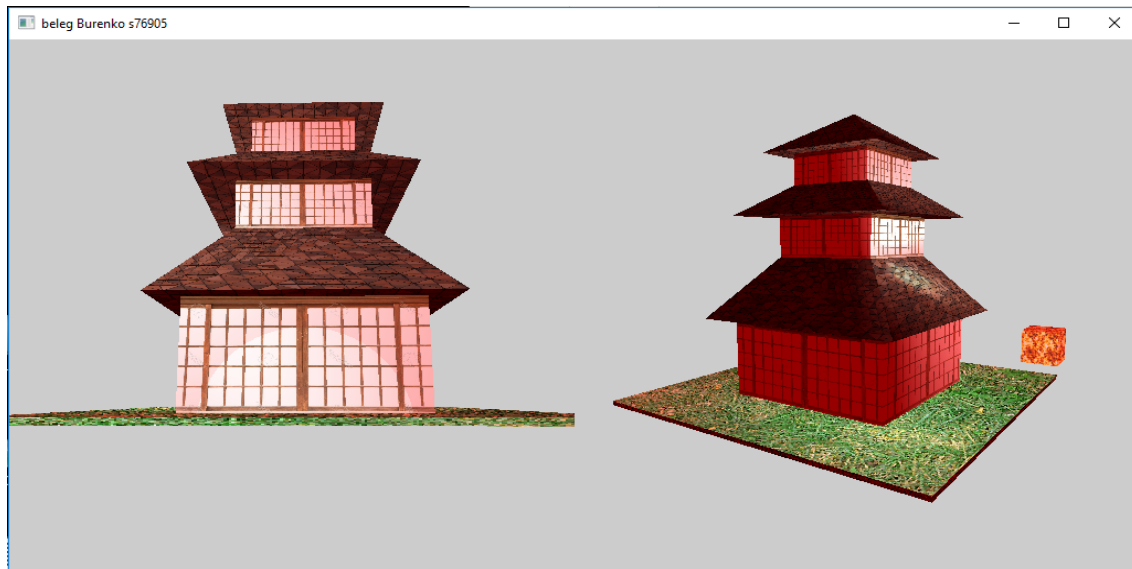
In der Szene sind 2 Viewports aktiv.

4.1 Viewport von Sonne

Das Viewport kann man links sehen. Die "Kamera" bewegt sich mit der Sonne zusammen nach oben und unter. Aus dem Sicht kann man sehr gut das Spotlight sehen.

4.2 Einstellbares Viewport

Das Viewport kann man rechts sehen. Die "Kamera" bewegt sich durch Bewegen vom Maus des Benutzers. Aus dem Sicht kann man alles sehen. Außerdem ist es mit dem Viewport möglich das Richtungslicht auf dem Dach bemerken.

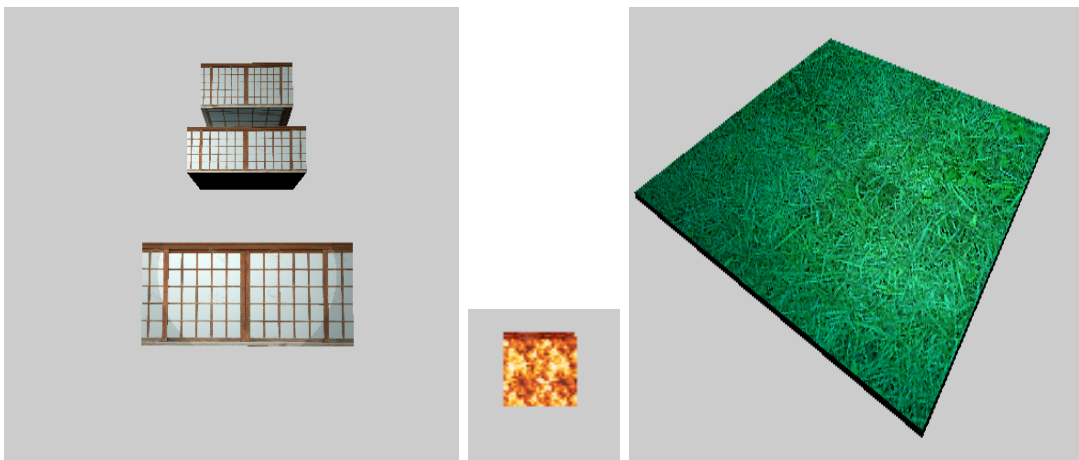


5 Figure

Für diese Szene sind 3 Figuren verwendet: ein Würfel, eine Piramide und einen Pyramidenstumpf. Alle Figuren haben eine neutrale Farbe, die danach mit einem Texture überdeckt wird.

5.1 Würfel

Der Würfel wird für Grass, Sonne und Gebäudeblocke verwendet.

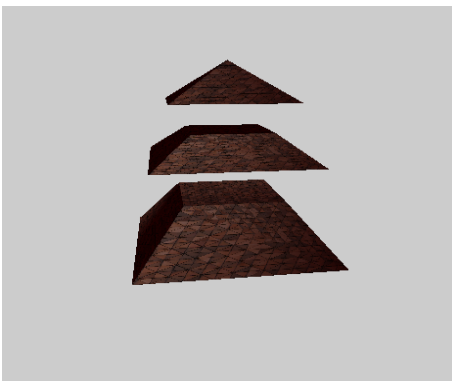


5.2 Piramide

Die Piramide ist nur für Dach von Pagode verwendet.

5.3 Pyramidenstumpf

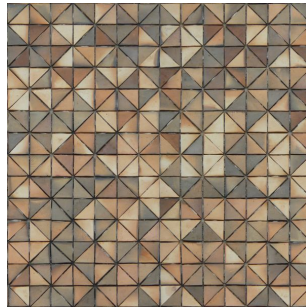
Der Pyramidenstumpf ist für den Zwischendach von Pagode verwendet.



6 Texture

Es gibt folgende Texturen: roof, wall, grass and sun.

roof:



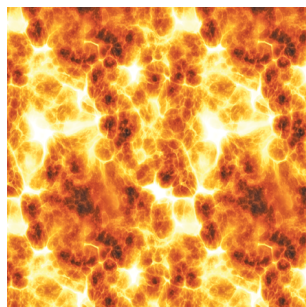
wall:



grass:



sun:



7 Light

Die Szene enthält drei Lichtquellen: Ambiente licht, Spotlicht und Richtungslicht. Alle Lichtquellen sind in dem Fragment Shader implementiert.

7.1 Ambiente Licht

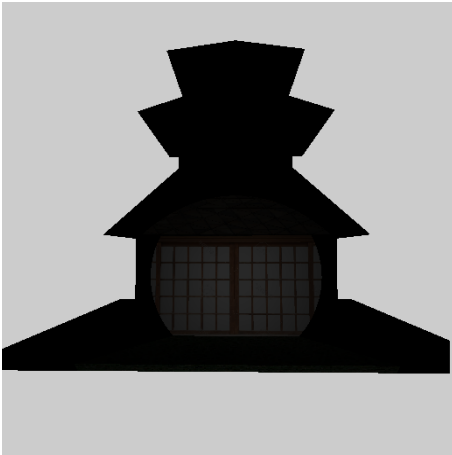
Position Ambiente Licht gibt es überall auf der Szene.

In dieser Szene wechselt es die Farbe von Weiß auf Rot und umgekehrt(siehe **Animation**), um den Sonnenuntergang zu simulieren.

7.2 Spotlicht

Position Bewegt sich mit der Sonne zusammen.

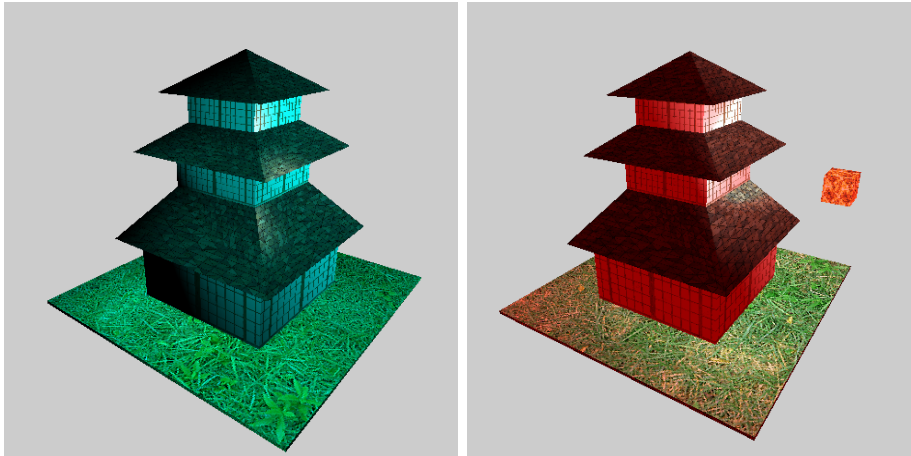
Das Spotlicht sieht wie ein Lichtkreis von der Richtung von der Sonne aus. Die Lichtquelle bewegt sich mit dem Licht zusammen (siehe **Animation**). Die Weiße Farbe wird vom Zenter bis Rand immer schwächer.



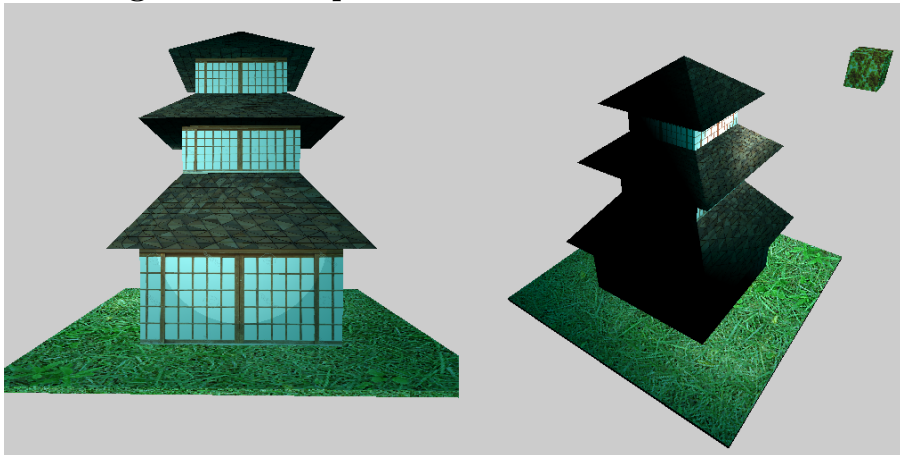
7.3 Richtungslicht

Position (4, 4, 4) Orientiert von der Sonne rechts oben.

Richtungslicht kann man sehr gut als Blicke auf dem Dach beachten. Die Blicke kann man nur auf zwei Seiten sehen, weil das Licht nur aus einer Seite die Pagode beleuchtet. Am besten ist es zu sehen wenn man das Spotlight ausmacht (siehe **Interaction**).



Richtungslicht und Spotlight zusammen

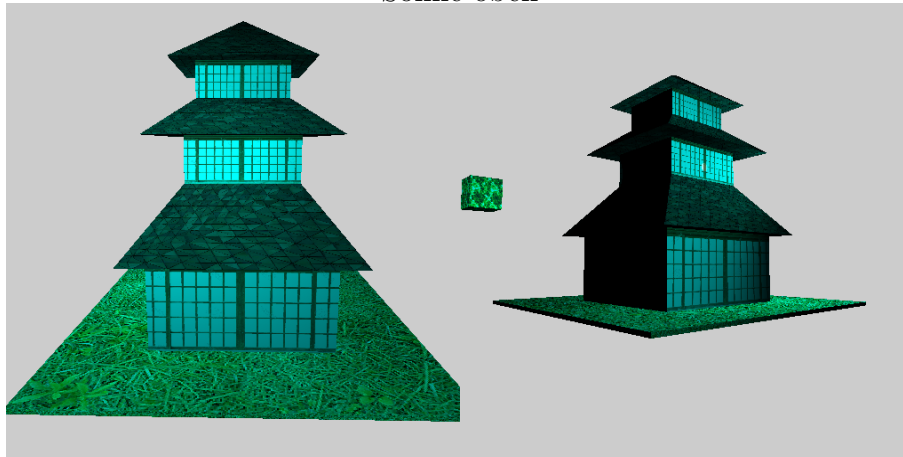


8 Animation

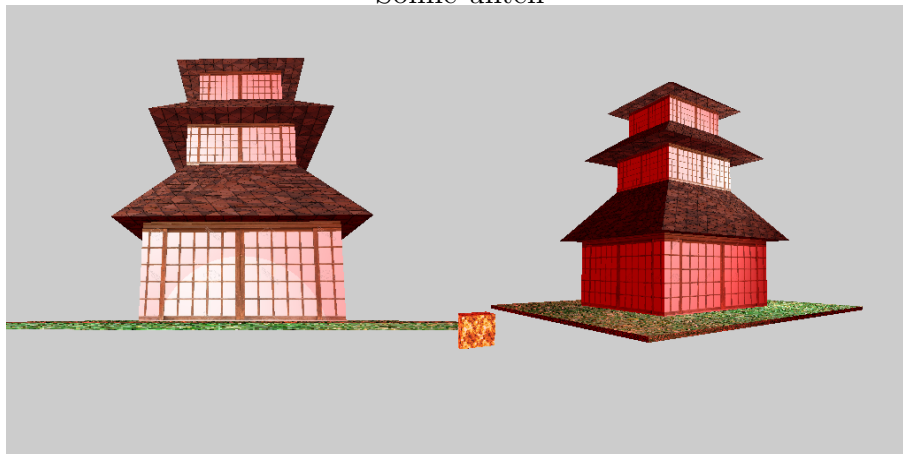
Animation kann man schon am Anfang der Szene sehen. Die Sonne bewegt sich an der Y-Achse nach oben bzw. unten.

Außerdem ändert sich die Farbe dabei, um Sonnenuntergang zu simulieren. Wenn die Sonne schon unter Niveau vom Grass-block bzw. der Erde ist wird die Farbe rot.

Sonne oben



Sonne unten



9 Interaction

Die Szene kann man mit Mouse sowie mit Keyboard steuern.

Berechnungen dazu findet man in Funktionen keyboard sowie mouse.

Keyboard mapping

Z ... kleiner

V ... größer

D ... Tiefenbuffer an/aus

1 ... Richtungslicht und Ambiente Licht an/aus

2 ... Spotlicht an/aus

10 Quellenverzeichnis

OpenGL SuperBible by Sellers, Wright Jr., Haemel

OpenGL Programming Guide : The Official Guide to Learning OpenGL, Version 2

Computergraphik 1 Vorlesungsmaterial by Prof. Dr. Wolfgang Oertel