

# Praktikum Computergrafik, WiSe 20/21

## Übungsblatt 4

- **Abgabefrist:** 25.01.2021 11:59:59
- Abgabe erfolgt per E-Mail an [jeos@mail.com](mailto:jeos@mail.com)
- **Betreff:** CG21WS ÜB4
- **Erste Zeilen der E-Mail:** Name der Autoren (*höchstens 2*) und Matrikelnummern
- Der **lauffähige Code** (*Quelltext*) soll **als Anhang** in der E-Mail mitgeschickt werden
  - Die folgenden Sprachen sind zugelassen: *Java*(JOGL/LWJGL), *C/C++*, *Python*(PyOpenGL), *C#*
  - **Keine** high-level Bindungen (*Beispiel: Python/ModernGL*) sind erlaubt
  - Bei JOGL: muss als **ZIP-Archiv** exportiert worden sein (*siehe die Anleitung in den Folien zu #1*)

### Quellen:

<b>Aufgaben</b>	<a href="http://www.uni-forst.gwdg.de/~wkurth/cg20_u04.pdf">http://www.uni-forst.gwdg.de/~wkurth/cg20_u04.pdf</a>
<b>Folien</b>	<a href="http://www.uni-forst.gwdg.de/~wkurth/cg20_f04.pdf">http://www.uni-forst.gwdg.de/~wkurth/cg20_f04.pdf</a>
<b>Code-Frameworks</b>	<a href="http://www.uni-forst.gwdg.de/~wkurth/cg20_c14.txt">http://www.uni-forst.gwdg.de/~wkurth/cg20_c14.txt</a> <a href="http://www.uni-forst.gwdg.de/~wkurth/cg20_c15.txt">http://www.uni-forst.gwdg.de/~wkurth/cg20_c15.txt</a> <a href="http://www.uni-forst.gwdg.de/~wkurth/cg20_c16.txt">http://www.uni-forst.gwdg.de/~wkurth/cg20_c16.txt</a>
<b>Texturbilder</b>	<a href="http://www.uni-forst.gwdg.de/~wkurth/cg20_c01.zip">http://www.uni-forst.gwdg.de/~wkurth/cg20_c01.zip</a>

**Liste der Aufgaben:**

1. Erstellen Sie den Code, der die folgendermaßen beschriebene Szene aufbaut:

- **Maschenboden**, der in der ganzen Szene verlegt wird (also die Szene soll mit dem Bodenhorizont rundum versehen sein);
  - Aus einfacheren Objekten zusammengebaute **zwei**farbige **Verkehrsampel mit dem Sockel** (zu diesem Moment ist nur erforderlich, ein **passendes 3D-geometrisches Objekt zu basteln**, das später als eine Verkehrsampel texturiert und weiter ausgestattet wird) – *dafür kann man beispielsweise einen [Kegel aus der GLUT-Bibliothek](#), einen [Zylinder aus der GLUT-Bibliothek](#) und **zwei** aufeinander gestapelte gleichgroße Würfel verwenden;*
  - Bogenförmige **Brücke** - ihr könnt (*kein Muss*) bereits gebasteltes Zahnrad (siehe das Übungsblatt #2) verwenden (also zur Hälfte querschneiden und die Zahngröße etwas verkleinern);
  - Einigermaßen realistisch parametrisierter **Nebel-Effekt**;
  - Über die ganze Szene verstreute, unterschiedlich große **Rechtecke**, die senkrecht zum Boden stehen (zeichnet nicht allzu viele Rechtecke – sonst muss Instancing mitimplementiert werden, was aber kein Ziel der Aufgabe ist) – für diesen Zweck (also für eine Zufallsverteilung) kann man beispielsweise die Methode nextFloat() der Klasse **java.util.Random** einsetzen;  
**Achtung:** kein Rechteck darf auf der Autobahn sowie sehr nah zur Ampel stehen
  - Gleichmäßige tastaturgesteuerte Bewegung (d.h. mit einem konstanten Schritt) des Betrachters in 6 Richtungen durch die Szene;
  - Gleichmäßige 360°-Rotation der Blickrichtung in 4 Richtungen – steuerbar durch die entsprechenden Tastaturtasten.
-

## 2. Texturieren Sie die Objekte in der [Aufgabe #1](#):

- Maschenboden (*nur ein sichtbar begrenzter Teil davon*):
  - versehen mit der Textur (**PNG-Bild**) eines realistisch aussehenden flachen Terrains (*findet also ein passendes Texturbild im Netz*)
- Terrain (*ein längliches Rechteck auf dem Maschenboden*):
  - versehen mit der Textur (**PNG-Bild**) einer Autobahn mit 3-4 Fahrstreifen, sodass sie unter der oben erwähnten Brücke geht. Die Verkehrsampel muss auch gleich am Fahrbahnrand und im wesentlichen Abstand von der Brücke stehen
- verstreute Rechtecke:
  - versehen Sie diese mit dem transparenten **PNG-Bild** eines Baumes

**Anregung:** Sie können versuchen, mithilfe der „[face culling](#)“-bezogenen OpenGL-Befehlen (ggfs. könnte auch der boolesche Befehl [glIsEnabled](#) mitbenutzt werden) und deren Parameter, diese Rechtecke von beiden Seiten beliebig (nach Ihrem Wunsch) zu texturieren;

- Verkehrsampel (nur der Kopfteil):
  - Texturieren Sie die Seiten der oben genannten **zwei** Würfel mit den passenden **PNG-Bildern** (die Straßenseiten: von ausgeblendeten [Ampelmännchen](#), sonst: einfach **schwarzgefärbt**), damit die Ampel realistisch aussieht

**Anschließend** rüsten Sie die Ampel mit der folgenden mausgesteuerte Funktionalität aus:

- **beim einmaligen Klick** wird die Ampel ein/ausgeschaltet
- **beim Doppelklick** wird die Ampelphase gewechselt

Die Basis dieser Funktionalitäten ist die Ersetzung entsprechender Texturbilder.

---

**Bilderquelle:** passende und frei verwendbare **PNG-Bilder** sind im Internet erhältlich [einige werden auch über StudIP hochgeladen] - beispielsweise durch die [erweiterte Suche auf Google Bilder](#) (*wählen Sie nur **frei nutzbare** Bilder unter dem **Filterfeld** „Nutzungsrechte“*).

3. Zeichnen Sie einen drehenden Würfel mit den durch die Methode `draw3D()` der Klasse `TextRenderer` beschrifteten Flächen:

