GEORG-AUGUST-UNIVERSITÄT
GÖTTINGEN

Prof Dr. Winfried Kurth,
Alex Tavkhelidze

Praktikum Computergrafik, WiSe 21/22
Übungsblatt 2

- **Abgabefrist**: 29.11.2021 11:59:59
- Abgabe erfolgt per E-Mail an jeos@mail.com
- **Betreff**: CG21WS ÜB2
- **Erste Zeilen der E-Mail**: Autorennamen *(höchstens 2 Kursteilnehmer)* und Matrikelnummern
- Der **lauffähige Code** *(Quelltext)* soll **als Anhang** in der E-Mail mitgeschickt werden
- Der Quelltext muss dabei als **ZIP-Archiv** exportiert worden sein (*siehe die Anleitung in den Folien zu #1*)

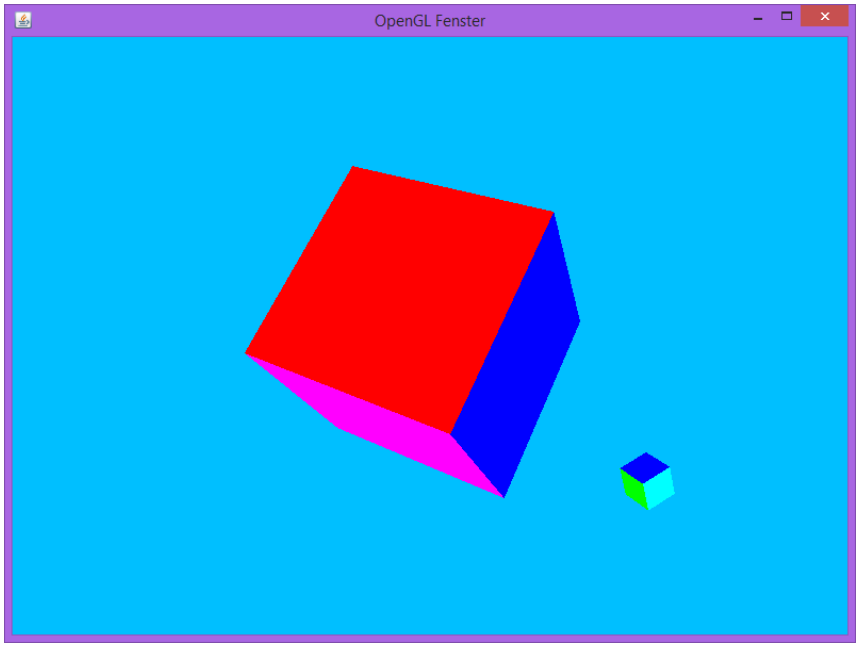
Quellen:

Aufgaben	http://www.uni-forst.gwdg.de/~wkurth/cg21_u02.pdf
Folien	http://www.uni-forst.gwdg.de/~wkurth/cg21_f02.pdf
Code-Frameworks	http://www.uni-forst.gwdg.de/~wkurth/cg21_c03.txt http://www.uni-forst.gwdg.de/~wkurth/cg21_c04.txt

Liste der Aufgaben:

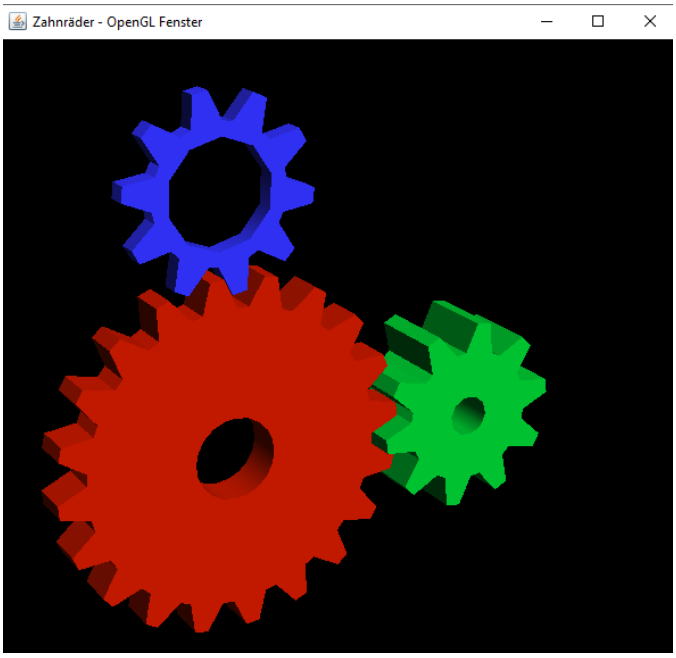
1. Zeichnen Sie einen drehenden Würfel, wobei die Rotationsgeschwindigkeit über eine Sinusfunktion definiert werden muss [d.h. steter wellenartiger Wechsel zwischen Beschleunigungs- und Bremsphasen].
   Im Laufe der Drehung sollten sich alle 6 Flächen des Würfels zeigen (also ohne „Mond-Effekt“ ☺) – dabei müssen die Flächen unterschiedlich gefärbt und zudem auch einfarbig aussehen (*also kein Farbverlauf*).

2. Ergänzen Sie den Code durch die Anwendung von weiter unten angegebenen Befehlen (*siehe Tabelle auf der Seite 4*), sodass neben dem drehenden Würfel und nach einer bestimmten Zeit (*also nicht sofort beim Ausführen des Codes*) ein anderer, in die Gegenrichtung drehender und deutlich kleinerer Würfel für eine Weile erscheint, dann verschwindet und dieses Verhalten stets wiederholt:



Anregung: versuchen Sie möglichst komplexere Animationen mit wenigeren Codezeilen zu erzeugen.

3. Zeichnet ein drehendes **Zahnrad** (*es ist dabei gleichgültig, welche Drehungsrichtung, Drehungsgeschwindigkeit und sonstige Parameter* [äußere/innere Radien, Art und Anzahl von Zähnen] *ihr wählt*):



Hinweise:

- ihr könnt ein Zahnrad durch die folgendermaßen parametrisierte Methode definieren:

```
public static void zahnrad (GL2 gl,
                            float innerer_radius,
                            float äußerer_radius,
                            float breite,
                            int zähne,
                            float zahn_tiefe)
```

- verwendet dabei **GL_QUAD_STRIP** und **GL_QUADS** (*siehe Folien #1*)

Befehl	Beschreibung
1) Transformationen:	
glScalef https://wiki.delphigl.com/index.php/glScale	Zuständig für die Skalierung (*zB für die Erzeugung von kleineren / größeren Objekten in der Szene*)
glTranslatef https://wiki.delphigl.com/index.php/glTranslate	Zuständig für die Verschiebung (*zB fürs Zeichnen von Objekten neben anderen / Verschiebung von Objekten in die Tiefe*)
glRotatef https://wiki.delphigl.com/index.php/glRotate	Zuständig für die Drehung (*für die Rotation um einen bestimmten Winkel und eine bestimmte Achse, sowie auch um einen beliebigen Ortsvektor*)
2) Speichern / Laden der aktuellen Matrizen:	
glLoadIdentity https://wiki.delphigl.com/index.php/glLoadIdentity	Lädt die Einheitsmatrix als eine aktuell anzuwendende Matrix (*zB fürs Rücksetzen der aktuellen Matrix nach Transformationen; wird auch als Anfangsbefehl vor dem Zeichnen benutzt*)
glPushMatrix https://wiki.delphigl.com/index.php/glPushMatrix	Speichert die aktuelle Matrix (ermöglicht die einfache Wiederverwendung nach dem anschließenden Aufruf des Befehls glPopMatrix – zB fürs Setzen eines weiteren Objekts)
glPopMatrix https://wiki.delphigl.com/index.php/glPushMatrix	Nimmt die zuletzt gespeicherte Matrix weg vom Stapel und setzt sie als eine aktuelle Matrix ein
3) Anwendung von basischen Operatoren https://www.java-tutorial.org/operatoren.html **und Kontrollstrukturen in Java:** https://www.java-tutorial.org/kontrollstrukturen.html	
if-Anweisungen und/oder **switch-Anweisungen** und/oder **Schleifen** und/oder **arithmetische und logische Operatoren**	