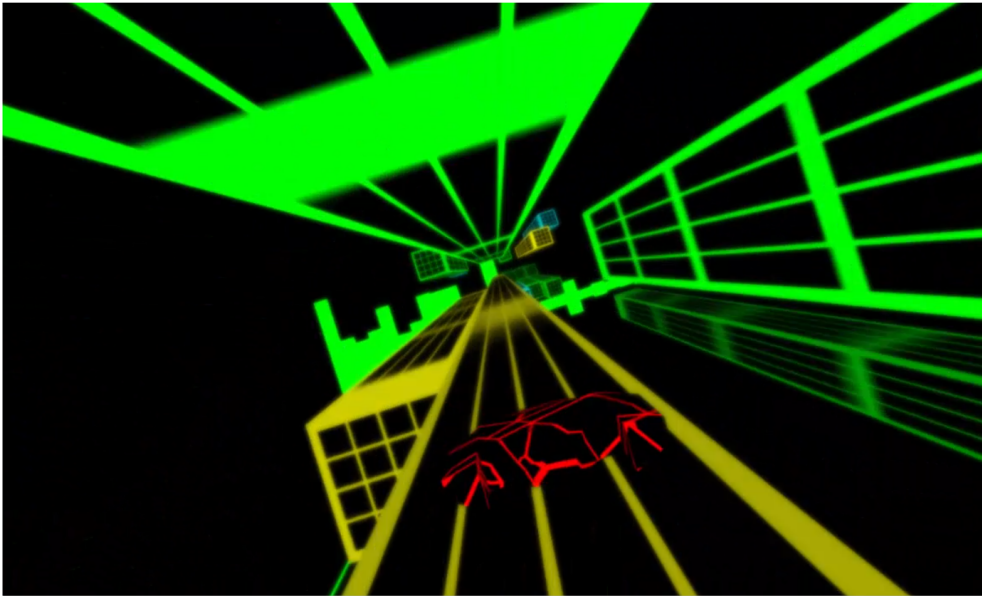


# Projektleitfaden CGA

Hinweise für das Abschlussprojekt in Computergrafik und Animation



## Vorgehensweise zum Abschluss des Moduls

Bis auf weiteres wird das Modul CGA mit einem Software-Projekt abgeschlossen. Beachten Sie dabei das folgende Vorgehen.

1

### EXPOSÉ (BIS 21.07.2024)

Die Exposévorlage muss ausgefüllt und an [cga-praktikum@gm.fh-koeln.de](mailto:cga-praktikum@gm.fh-koeln.de) geschickt werden. Die Vorlage ist unter diesem [Sciebo-Link](#) zu finden.

2

### ABMELDUNG IM PSSO

Abmeldezeitraum ist bis 31.07.2024

3

### ABNAHMETERMIN

(12.08.–16.08., 19.08.–23.08. & 26.08.–30.08.2024)

Nach der Anmeldung wird Ihnen ein Abnahmetermin in der gewählten Abnahmewoche des Moduls zugeteilt. Außerhalb dieser Zeiten sind keine Abnahmen möglich.

## FAQ

### Welche Prüfungsform wird angeboten?

Die Prüfungsform wird bis auf weiteres ein Projekt bleiben.

### Wie groß darf das Projektteam sein?

2-3 Personen

### Was passiert, wenn ein Teammitglied abspringt?

Die Anmeldung ist verbindlich. Springt ein Teammitglied danach ab, so bekommt er/sie die Note 5,0. Das Projekt der anderen wird ganz normal bewertet.

### Muss ich das Praktikum bestanden haben?

Ja, alle Teammitglieder müssen das Praktikum bestanden haben.

### Darf ich auch JAVA verwenden?

Nein.

## PROJEKT

4

Das Projekt muss eigenverantwortlich durchgeführt werden und darf ohne Rücksprache thematisch nicht vom Exposé abweichen. Außer den im Praktikum verwendeten Bibliotheken und Programmiersprachen (OpenGL, LWJGL, JOML, Kotlin, Standard SDK) sind keine weiteren Bibliotheken erlaubt. Code von Dritten muss im Code und in der Featureliste (siehe 5) aufgeführt werden und geht nicht in die Bewertung ein. Die Bewertungskriterien für das Projekt finden Sie weiter unten.

## ABGABE

5

Nutzen Sie dafür das folgende GitHub-Template-Repository und folgen Sie den dortigen Angaben: [https://github.com/CGA-TH-Koeln/CGA\\_Project](https://github.com/CGA-TH-Koeln/CGA_Project)  
Das Projekt muss **spätestens bis Freitag vor der Abnahmewoche als GitHub-Repository mit Featureliste und Erklärvideo** abgegeben werden. Dafür fügen Sie **thflo, umuesse** und **Igrebe1 als Collaborators** hinzu. Nach der Abgabe dürfen keine inhaltlichen Änderungen mehr vorgenommen werden, außer Bug Fixing oder Refactoring.

## ABNAHME

6

In der Abnahme des Projekts haben Sie **max. 8 Minuten Zeit ihr Projekt vorzustellen**. Dabei sollten sie das Projekt kurz vorführen und sich dann ein bis zwei Features aussuchen, deren Implementierung Sie anhand Ihres Quellcodes erklären. Danach folgt noch eine **Fragerunde von ca. 7-13 Minuten Länge. In dieser Fragerunde sollte jeder des Teams in der Lage sein, sämtlichen Code erklären zu können**. Teile die nicht zufriedenstellend erläutert werden, können zu Punktabzug führen.  
Die Note wird Ihnen wenige Tage später per Email mitgeteilt.

## ANSPRECHPARTNER

Bei Fragen und Problemen wenden Sie sich bitte an eine der folgenden Kontaktadressen:

- Allgemeine Fragen von generellem Interesse zum Abschlussprojekt bitte im Forum des ILU-Kurses stellen.
- [cga-praktikum@gm.fh-koeln.de](mailto:cga-praktikum@gm.fh-koeln.de) für terminliche Wünsche, Exposé-Abgabe sowie sonstige Fragen.



## Bewertungsschema

Punkte	Note
0 - 6	Nicht bestanden
7	4,0
8	3,7
9	3,3
10	3,0
11 - 12	2,7
13 - 14	2,3
15 - 16	2,0
17 - 18	1,7
19 - 20	1,3
> 20	1,0

## Bewertungskriterien

Im Folgenden werden die Bewertungskriterien für das Abschlussprojekt aufgelistet sowie das zu den einzelnen Features passende Benotungsschema. Die Teilpunktzahlen können abhängig von der Abnahme abgewertet werden, wenn sich herausstellt, dass die umgesetzten Konzepte nicht verstanden wurden.

Feature	Sehr gute Lösung (3-X Punkte)	Passable Lösung (1-2 Punkte)	Nicht akzeptable Lösung (0 Punkte)
Rendering	Mehrere externe, parametrisch erstellte oder eigene Modelle wurden eingebunden und zu einer sinnvollen Szene zusammengestellt.	Verschiedene Modelle sind wahllos ohne größeren Zusammenhang dargestellt.	Keine Geometrie, lediglich die Geometrie aus den Praktika oder nur einfache geometrische Formen wurden hardgecodet und dargestellt.
Shader-programmierung	Wechsel zwischen verschiedenen Shadern möglich, bspw. Phong und Toon-Shader oder anderen. Dynamische Parameter in Vertex- und/oder Fragmentshader sinnvoll eingesetzt, bspw. für Animation, Texturblending oder andere Effekte.	Wechsel zwischen verschiedenen einfachen Shadern möglich, bspw. Phong-Shader oder Visualisierung der Normalen. Keine dynamischen Parameter verwendet.	Nur ein Standardshader aus Vorlesung / Praktikum.
Transformationen	Siehe „Passable Lösung“ plus zusätzliche Implementierung komplexer abhängiger Animationen mittels mehrstufiger Transformationshierarchien.	Mehrere Objekte bewegen sich automatisch oder interaktiv mittels abhängiger Transformationen.	Die Verwendung des Szenegraphen geht nicht über das Praktikum hinaus.
Kamera	Siehe "Passable Lösung" plus interaktiver Wechsel zwischen einer perspektivischen und orthographischen Kamera möglich oder Wechsel zwischen einer Orbitkamera oder Fly-Through-Kamera möglich	Die Kamera wurde um interaktive Features erweitert. Zum Beispiel eine Zoom-Funktion.	Nur die Kamera aus dem Praktikum.
Texturen	Siehe „Passable Lösung“ plus weitere Features wie Verwendung von Advanced Textures, oder Texturmatrizen zur Animation.	Die Verwendung der Texturdaten geht über das Praktikum hinaus. Beispiel: Skybox (Cubemap) oder parameterabhängige Texturierung (Texturemixing)	Es wurden keine bzw. einfache Texturen wie im Praktikum verwendet.
Idee	Das Projekt implementiert eine klare und kreative Idee, bspw. Visualisierung von Strömungsdaten, Implementierung eines kleinen Spiels, etc.	Das Projekt implementiert eine simple Idee im Sinne eines 1st- oder 3rd-Person Walk-Throughs, d.h. man bewegt sich einfach nur durch eine Szene.	Das Projekt hat keine tiefergehende Idee umgesetzt.
Weitergehende Konzepte	Es wurden komplexe Algorithmen umgesetzt: <ul style="list-style-type: none"> <li>• Shadow Mapping</li> <li>• Deferred Shading</li> <li>• Inverse Kinematics</li> <li>• Physics</li> <li>• ...</li> </ul>		