

# 4. Aufgabenblatt zur Veranstaltung *Computergrafik und Visualisierung II* im Sommersemester 2020

Bearbeitungszeitraum: 09.05.-29.05.2020

Hochschule für Technik und Wirtschaft Dresden, Friedrich-List-Platz 1, 01069 Dresden

Prof. Dr. Marco Block-Berlitz, Rainer Uhlemann

Im vierten Teil der Veranstaltung werden wir eine kleine Gameenginevorlage erarbeiten und diese mit GLSL und einem eigenen Schwarmkonzept kombinieren. Hinweis: Die Lösungen der praktischen Übungsaufgaben können als Gruppe (max. 3) bearbeitet werden und sind spätestens am **29.05.2020** einzureichen. Die Einreichung erfolgt als zip-Datei, in der das komplette ausführbare Eclipseprojekt und ein PDF zu den Lösungserläuterungen vorhanden sind.

## 1 Einarbeitung in die Grundlagen

Die Buchprogramme<sup>1</sup> laufen in Eclipse und wir haben bereits etwas Erfahrung mit OpenGL [2] und GLSL [3] gewinnen können. Eure Nachfragen und Lösungen haben mir gezeigt, dass Ihr in der aktuell sehr schwierigen Lage wirklich sehr gut bei der Sache seid, die Inhalte fleissig erarbeitet habt und wir uns inhaltlich auf die Zielgerade bewegen können. Im Zuge der kommenden Aufgaben und in Bezug auf Eure Nachfragen und Hinweise habe ich das 1. Buchkapitel noch einmal überarbeitet. Es wurde um die **Entwicklung einer kleinen Gameengine** und den Aspekt **Steuerungsverhalten** erweitert. Diese Abschnitte sollt Ihr zunächst durcharbeiten. Ihr findet das neue Kapitel und den überarbeitete Codesammlung bei OPAL. Weiterhin sind Foliensätze zum Schwarmverhalten und der Engineentwicklung bei OPAL zu finden.

Das überarbeitete Buchkapitel beinhaltet sicherlich noch den einen und anderen Fehler oder es fehlen notwendige Erläuterungen und Hinweise. Jeden der einen Fehler entdeckt und mir mitteilt, werde ich gerne in der kommenden Auflage namentlich bei den Danksagungen erwähnen (Fabian ist schon dabei!).

## 2 Übungsaufgaben

Zur Erinnerung: Die Übungsaufgaben haben den Zweck, Sie auf die Klausur vorzubereiten. Alle Aufgaben sollten durch das Buch zur Veranstaltung beantwortet werden können [1].

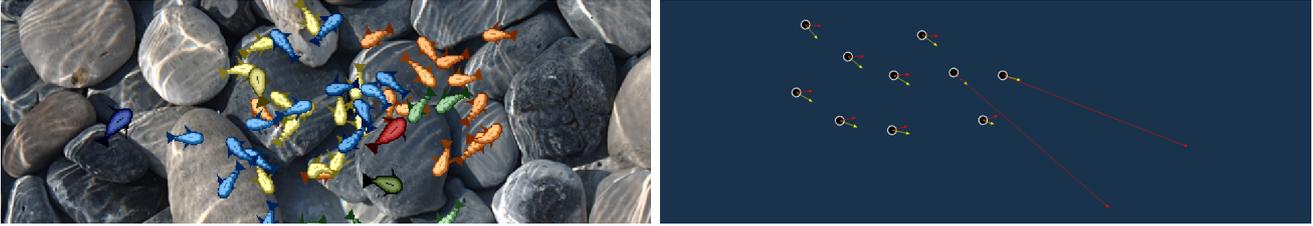
1. Beschreiben Sie allgemein, was unter dem Konzept Steuerungsverhalten verstanden wird.
2. Warum ist es sinnvoll, die beiden Methoden `render` (Objekt wird gezeichnet) und `update` (Objektposition, -orientierung, ... wird angepasst) zu trennen?
3. Welche Rolle spielt das Strategy-Pattern bei der Integration von Verhalten? Welchen Vorteil erhalten wir bei der Verwendung?
4. Erläutern Sie die Berechnung (anhand einer Skizze und Formeln) der konkreten Steuerungsbeispiele *folgen* (seek), *ankommen* (arrive) und *fliehen* (flee).
5. Erläutern Sie detailliert, wie Schwarmverhalten durch das von Craig Reynolds 1986 vorgestellte, zusammengesetzte Steuerungsverhalten bestehend aus drei Kräften imitiert werden kann. Wie lassen sich die Kräfte kombinieren?
6. Beschreiben Sie die Euler-Integrationsmethode zur Transformation von Steering-Kräfte-Modellen in Positionsänderungen von Objekten.

---

<sup>1</sup>[http://www.vividus-verlag.de/beleuchtung\\_und\\_rendering/index.html](http://www.vividus-verlag.de/beleuchtung_und_rendering/index.html)

### 3 Praktischer Teil

1. [15 Punkte] Denken Sie sich ein Projekt zum Thema Schwarmverhalten oder ein interessantes Partikelsystem aus. Erstellen Sie ein grobes Klassendiagramm. Das Projekt soll ebenfalls visuell skizziert werden.
2. [20 Punkte] Implementieren Sie Ihr vorgestelltes Projekt. Die zur Verfügung gestellten Codebeispiele können dabei die Grundlage Ihrer Projektidee bilden und dürfen gerne Verwendung finden.



Integrieren Sie die notwendigen Methoden für die Realisierung Ihres Projekts. Achten Sie dabei auf ein sinnvolles Konzept! Achten Sie weiterhin auf die individuellen Bewegungs- und Handlungsmöglichkeiten der Agenten bzw. Partikel.

3. [11 Punkte] Visualisieren Sie Ihren Schwarm in einem FragmentShader. Dazu könnten Sie beispielsweise die Individuen durch einfache OpenGL-Primitive konstruieren.

Um eine Übersicht zu den bisherigen und noch möglichen Punkten der Aufgabenzettel zu geben, habe ich folgende Tabelle für Euch:

Aufgabenzettel	erreichbare reguläre Punkte	erreichbare Zusatzpunkte
1	30	0
2	59	20
3	15	40
4	46	0
5	0	50
Summen	<b>150</b>	<b>110</b>

Geplant sind dabei insgesamt **fünf** Arbeitsblätter mit insgesamt 150 Punkten und maximal 110 Zusatzpunkten.

Bitte bleibt gesund!

### Literatur

- [1] Block-Berlitz, M.: „Warum sich der Dino furchtbar erschreckte: Lehrbuch zu Beleuchtung und Rendering mit Java, LWJGL, OpenGL und GLSL“, vividus Wissenschaftsverlag, 2019
- [2] Shreiner D., Woo M., Neider J., Davis T.: „OpenGL Programming Guide“, 6. Auflage, Pearson Education, Addison-Wesley Verlag, 2008
- [3] Rost R.J., Licea-Kane B.: „OpenGL Shading Language“, 3. Auflage, Pearson Education, Addison-Wesley Verlag, 2010