

„Prüfungs-Essay“ – Aufgabenstellung

Die Veranstaltung wird in diesem Semester aufgrund der Corona-Einschränkungen in Form eines **individuellen** schriftlichen Essays bewertet, welches eine Beschreibung Ihrer Programmierlösung der praktischen Übung beinhaltet und diese anhand der untenstehenden Fragen in den Kontext des Vorlesungsinhalts stellt. Das Essay wird benotet und ersetzt zusammen mit dem Programmcode sowohl die Übungsabgabe als auch die Präsenzklausur. Sie geben somit zwei Dateien ab: Das Essay sowie ein Archiv mit Ihrem Programmcode der Übung.

Aufgabe:

In Ihrem Essay beschreiben Sie Ihre Lösung der praktischen Programmierübungen und stellen diese anhand der untenstehenden Fragen in den Kontext der Vorlesung.

- A. **Beschreiben Sie Ihre Programmierlösung:** Wie ist das Programm aufgebaut, welche Klassen und Funktionen implementieren welche Übungsaufgabe? Verweisen Sie entsprechend auf Ihren eingereichten Quellcode.
- B. Beschreiben Sie, welche **Schwierigkeiten und Erfolge** Sie bei der Implementierung hatten: An welchen Stellen haben Sie mehr Zeit benötigt, warum, welche Probleme gab es zu lösen? (Wir benoten nicht Ihre Schwierigkeiten, sondern die Darstellung und Reflektion.)
- C. Zeigen Sie **Screenshots der Lösungen aller Teilaufgaben** (also der Textausgabe der ersten Übung, des Colourmappings der zweiten Übung, etc.). Zeigen Sie ggf. auch Screenshots, die eventuelle Schwierigkeiten zeigen (also auch, was nicht funktioniert hat).
- D. **Beschreiben Sie Ihre Wahlaufgabe** am Ende der Übung: Welche Technik haben Sie zusätzlich zu den vorgegebenen Übungsaufgaben implementiert? Was lernen Sie über die Daten durch die implementierte Technik (welche Eigenschaften der Daten sind nun sichtbar, die vorher nicht zu sehen waren)?
- E. **Beantworten Sie die untenstehenden Fragen** und stellen Sie Ihr Programm somit **in den Kontext der Vorlesung**. (WICHTIG: Beachten Sie die Randomisierung durch Ihre Matrikelnummer!) Kennzeichnen Sie klar im Text, welche Frage wo beantwortet wird und referenzieren Sie entsprechend Vorlesungsfolien und Zeilen in Ihrem Programmcode.

Bitte beachten Sie: Die **Note** ergibt sich zu **30% aus Teilen A bis D**, zu **70% aus Teil E**. Um zu bestehen, müssen alle Aufgabenteile bearbeitet sein, der Programmcode muss vorliegen, **UND** es müssen in Teil E mindestens 50% der Punkte erreicht sein (das entspricht in etwa dem ursprünglichen Plan einer kürzeren Abgabe als Übungsnachweis sowie einer Präsenzklausur).

Layout und Umfang:

- Bei der Übung konnten Sie in Gruppen an Ihrem Quellcode arbeiten. **Das Essay hingegen wird Ihre individuelle Leistung sein.** Hier schreiben Sie also in Ihren eigenen Worten, **identische Texte müssen wir somit als Plagiat werten.**
- Ihr Quellcode wird nicht benotet, sondern nur Ihre Beschreibung, Einordnung und Ihre Reflektion aufgetretener Probleme. Vorliegen des Quellcodes ist aber Voraussetzung für das Bestehen.
- Schreiben Sie möglichst technisch-wissenschaftlich und „nüchtern“. Also kein „Erfahrungsbericht“, sondern ein **wissenschaftlich-technischer Artikel**.
- Verwenden Sie für Ihren Essay **Schriftgröße 11**.
- Der **Umfang** des Essays soll **11 (elf) Seiten Text inklusive Abbildungen** und Referenzen **nicht überschreiten**. Seiten, die das Limit überschreiten, werden nicht bewertet.

- Formatieren Sie Ihren Essay mit einer **Kopfzeile**, die Ihren Namen und Ihre Matrikelnummer sowie die aktuelle Seitennummer enthält.
- ACHTUNG: Die Leitfragen sind zum Teil durch Ihre **Matrikelnummer randomisiert** – lesen Sie also aufmerksam und stellen Sie sicher, die richtigen Aufgaben und ggf. numerischen Werte zu verwenden.
- **Kennzeichnen Sie klar im Text, welche Frage wo beantwortet wird.** Wir werden nicht nach ungekennzeichneten Antworten suchen.
- Wird danach gefragt, geben Sie bitte klar die entsprechenden Stellen in Ihrem Code an (Datei, Methodenname, Zeilennummer).
- Geben Sie gerne als **Referenzen die entsprechenden Vorlesungsfolien** an (bitte im Format „(V-T6F23)“ für Thema 6 Folie 23) bzw. Stellen aus der bereitgestellten oder anderweitigen Literatur.
- Geben Sie jeder **Abbildung eine Nummer** (Abb. 1, Abb. 2, etc.) und nutzen Sie diese Nummer als Referenz im Text. **Referenzieren Sie alle Abbildungen im Text** – Abbildungen, auf die nirgends verwiesen, „existieren nicht“.

Abgabe:

- Das **Essay im PDF-Format** sowie der Programmcode der **Übung als ZIP Archiv** (also zwei Dateien) müssen bis **Montag, 26. Juli 2021, 23:59** digital abgegeben werden. Frühere Abgaben sind jederzeit möglich.
- Die **Abgabe erfolgt über Moodle** (die Funktion wird rechtzeitig freigeschaltet, Informationen folgen). Bitte machen Sie sich vor der Deadline mit der Abgabefunktion vertraut. **Verspätete Abgaben werden nicht akzeptiert**, wir empfehlen also sehr, nicht bis zur letzten Minute zu warten. Falls es technische Probleme gibt, sind wir am **Abgabetag bis 18:00 per E-Mail zu erreichen** – bitte senden Sie uns in diesem Fall eine Telefonnummer, unter der wir zurückrufen können.

Fragen zur Einordnung Ihres Programms in den Vorlesungsstoff (Teil E)

Für eine volle Punktzahl (100 Punkte) gehen Sie bitte auf sämtliche Fragen ein.

Kennzeichnen Sie klar im Text, wo Sie welche Frage beantworten (z.B. indem Sie einem Satz „Antwort 1.1.“, „Antw. 1.1.“ etc. voranstellen).

Einige Fragen sind mit Ihrer Matrikelnummer randomisiert. Die Kennzeichnung ist folgendermaßen:

Ziffern sind in der Reihenfolge 1234567 bezeichnet. Ist Ihre Matrikelnummer also „6372884“, so ist die 1. Ziffer eine „6“, die 2. Ziffer eine „3“, die 7. Ziffer eine „4“. Die 7. Ziffer ist in dem Beispiel gerade.

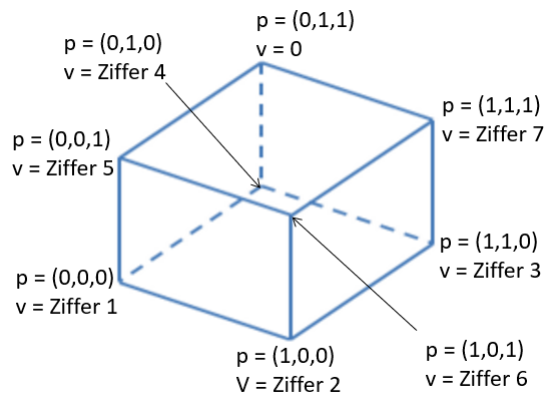
Steht in der Frage „(6. Ziffer gerade) TEIL A, (6. Ziffer ungerade) TEIL B“ so wählen Sie bitte den entsprechenden Teil aus.

1. Allgemein

- 1.1. **(3 P)** Wie haben Sie das Konzept der Visualisierungs-Pipeline in Ihrem Programm umgesetzt? Geben Sie ggf. Codestellen an.

2. Gitter und Rekonstruktion

- 2.1. **(3 P)** Für die Streamlines haben Sie trilineare Interpolation implementiert. Beschreiben Sie unter Angabe der Quellcodezeilen und Vorlesungsfolien, wie Ihre Methode zu der Beschreibung in der Vorlesung korrespondiert.
- 2.2. **(5 P)** Für die Gitterzelle in Ihrem Cartesischen Gitter mit den Gitterpunktskoordinaten „p“ (siehe Abbildung) sind an den acht Gitterpunkten als Werte „v“ (siehe Abbildung) die **Ziffern Ihrer Matrikelnummer** bzw. bei $p=(0,1,1)$ eine „0“ gespeichert (siehe Abbildung). Beschreiben Sie **alle Zwischenergebnisse** und das **Resultat** Ihres Codes bei der **Interpolation an die Position (0.2/0.4/0.6)** innerhalb der Gitterzelle.



- 2.3. **(5 P)** Was muss an Ihrem Programm geändert werden, falls die Daten nicht auf einem Cartesischen Gitter, sondern auf einem (**Matrikelnummer 6. Ziffer gerade**) **kurvilinearen** bzw. (**6. Ziffer ungerade**) **unstrukturiertem** Gitter gegeben sind? Wie finden Sie die Gitterzelle an einer bestimmten Position im Raum, was ändert sich bei der Interpolation?

3. Computergrafik

- 3.1. **(3 P)** Wo und wie taucht das Konzept der Grafikpipeline in Ihrem Programm auf?
- 3.2. **(3 P)** Wo tauchen Koordinatentransformationen in Ihrem Programm auf, und welche? Wozu werden diese benötigt?
- 3.3. **(5 P)** Ändern Sie in ihrem Code die Darstellung so, dass Sie eine orthografische Projektion erhalten. Benennen Sie die Codestelle und vorgenommene Änderung. Stellen Sie von einer beliebigen Szene die neue Projektion mit der ursprünglichen nebeneinander (zeigen Sie beide Bilder) und erklären Sie die Unterschiede.
- 3.4. **(4 P)** In welcher Art sind die Themen „Depth test“, „Texturing“, und „Blending“ in Ihrer Übung aufgetaucht? Hatten Sie Schwierigkeiten, und wie haben Sie diese gelöst?

4. Skalarfeld-Visualisierung

- 4.1. **(5 P)** In der Übung haben sie den Marching Squares Algorithmus zur Darstellung von Isolinen in 2D implementiert. Beschreiben Sie, welche Änderungen an Ihrem Programm nötig wären (geben Sie die Codestellen an), um zusätzlich 3D Isoflächen darzustellen, und zwar (**Matrikelnummer 7. Ziffer gerade**) mit **Marching Cubes** bzw. (**7. Ziffer ungerade**) mit **einem Raycaster**. Welche Vor- und Nachteile hat der gewählte Ansatz verglichen mit der jeweils anderen Möglichkeit, Isoflächen zu visualisieren?
- 4.2. **(4 P)** Wofür wäre ein „Asymptotic Decider“ in Ihrem Programm nützlich, und wo würden Sie ihn implementieren (geben Sie die Codestelle an)?
- 4.3. **(5 P)** Laden Sie einen Screenshot Ihrer Colourmapping-Übung auf den „Colour vision deficiency emulator“ unter <http://hclwizard.org/cvdemulator/> hoch. Kopieren Sie die resultierenden Abbildungen aller verfügbaren Emulationen in Ihre Abgabe. Würden Sie Ihre Darstellung anders interpretieren, wenn Sie eine der emulierten Farbwahrnehmungen hätten?
- 4.4. **(4 P)** Beschreiben Sie, welche Probleme bei der in der Übung gewählten Farbskala in Bezug auf Wahrnehmungslinearität auftauchen. Was müsste an Ihrem Programm geändert werden, um Farbskalen im HCL Raum zu ermöglichen (geben Sie die Codestelle an)?
- 4.5. **(6 P)** Nutzen Sie das in der Vorlesung genannte Webtool unter <https://hclwizard.org/creator/> und erstellen Sie drei Farbskalen mit der folgenden „Type of palette“ (oben rechts auf der Seite): (1) „Basic: Qualitative“, (2) „Basic: Sequential (multi-hue)“, (3) „Basic: Diverging“. Experimentieren Sie und wählen Sie die **Parameter frei**, jedoch mit **folgenden Bedingungen**: (a) „HUE 1“ = Matrikelnummer **Ziffern 567 geteilt durch 3**; (b) „CHROMA 1“ = **Ziffern 34** Ihrer Matrikelnummer; (c) „NUMBER“ = **Ziffern 23** Ihrer Matrikelnummer (**oder 40 falls der Wert über 40 liegt**).
Fügen Sie **Bilder** sowie die **Werte alle gewählten Parameter** (also HUE, CHROMA, ...) Ihrer **drei Farbskalen** in Ihre Ausarbeitung ein und diskutieren Sie, aus welchen Gründen die Skalen für die Visualisierung in der Übung geeignet sind oder nicht und was ggf. verbessert werden muss.

5. Vektorfeld-Visualisierung

- 5.1. **(5 P)** Beschreiben Sie den Unterschied zwischen Streamlines und Streaklines und beschreiben Sie, welche Änderungen in Ihrem Programm nötig wären, um Streaklines zu visualisieren. Geben Sie die Codestellen an (Datei und Zeilennummer), an denen Änderungen nötig wären.
- 5.2. **(3 P)** Welche Rolle spielt die Position der Seedpoints für Streaklines? Ist der Jobard & Lefer Algorithmus noch nützlich für Streaklines? Warum?
- 5.3. **(5 P)** Der von uns verwendete Tornado Datensatz wurde auch in dem Paper von Kanzler et al. (2016) verwendet, das ich im Moodle bereitgestellt habe. Stellen Sie in Ihrer Ausarbeitung eine **beliebige Abbildung** des Tornados **aus dem Paper** mit einer **Abbildung aus Ihrer Übung** nebeneinander und diskutieren Sie die Unterschiede: Welche Teile und

Merkmale des Strömungsfeldes sind wie erkennbar, was ist gut erkennbar und wo gibt es Probleme, wie könnten die Darstellungen verbessert werden?

6. Direct Volume Rendering

- 6.1. **(6 P)** Nehmen Sie an, Sie haben einen Raycaster für DVR in Ihr Programm eingebaut. Wie erstellen Sie Ihre **individuelle Transferfunktion** (es gibt viele Möglichkeiten), um die Windgeschwindigkeit zu visualisieren? **Zeichnen Sie ein Diagramm** mit der Geschwindigkeit auf der x-Achse und den ROT/GRÜN/BLAU/ALPHA-Werten (RGBA) auf der y-Achse (analog zu z.B. im Buch von Telea (2015), Fig. 10.10). **Denken Sie sich die Kurven der RGBA-Werte aus und beschreiben Sie**, wie Sie die resultierende Visualisierung erwarten würden.
- 6.2. **(3 P)** Wäre der HCL-Farbraum auch für die Transferfunktion in Ihrem Programm besser geeignet als der RGB-Farbraum?
- 6.3. **(3 P)** Welche Sampling-Schrittweite wählen Sie für den Raycaster? Erläutern Sie den Zusammenhang mit der trilinearen Interpolation. Was passiert, wenn zu viele bzw. zu wenige Samples verwendet werden?
- 6.4. **(4 P)** Erklären Sie in Ihren eigenen Worten, was die „Front-to-back strategy“ beim DVR mit „Schwarzchild’s equation“ zu tun hat.
- 6.5. **(3 P)** Hat das Phong-Beleuchtungsmodell eine Relevanz für DVR? Welche, bzw. warum nicht?

7. Informations-Visualisierung und Visual Analysis

- 7.1. **(3 P)** Wie würden Sie „Focus & Context“ implementieren, um die 3D Streamlines besser interpretierbar zu machen?
- 7.2. **(5 P)** Wie würden Sie „Brushing & Linking“ implementieren, um die 3D Streamlines besser interpretierbar zu machen? Wie kann z.B. die Transparenz interaktiv angepasst werden oder Linien ausgedünnt werden? Welche Arten von InfoVis Plots sind hier für das Brushing geeignet und warum?
- 7.3. **(5 P)** Michael Böttinger hat in seinem DKRZ-Vortrag ein Beispiel für interaktive visuelle Analyse (IVA) mittels Parallel Coordinate Plots (PCP) und Scatterplotmatrizen (SPM) gezeigt (Folien 23-25 seines Vortrags). Diskutieren Sie, wie **Sie IVA von PCP und SPM kombiniert mit Direct Volume Rendering** nutzen können, um die Daten in Ihrer Übung zu analysieren. Welche Variablen legen Sie auf die Achsen des PCP und der SPM? Wie erwarten Sie sehen die entstehenden PCP und SPM Darstellungen aus? Was brushen Sie und welchen Effekt erwarten Sie auf die DVR Darstellung?