

LEHRSTUHL FÜR RECHNERTECHNIK UND RECHNERORGANISATION

Aspekte der systemnahen Programmierung bei der Spieleentwicklung

Projektaufgabe – Aufgabenbereich Bildverarbeitung (A210)

1 Organisatorisches

Auf den folgenden Seiten finden Sie die Aufgabenstellung zu Ihrer Projektaufgabe für das Praktikum. Die Rahmenbedingungen für die Bearbeitung werden in der Praktikumsordnung ¹ festgesetzt, die Sie auch über die Praktikumshomepage ² aufrufen können.

Der **Abgabetermin** ist der **1. Februar 2016 (23:59 MEZ)**. Die Abgabe erfolgt per Git in das im Gitlab für Ihre Gruppe eingerichtete Projektrepository. Bitte beachten Sie unbedingt die in der Praktikumsordnung angegebene Liste von abzugebenden Dateien!

Wie in der Praktikumsordnung beschrieben, sind die Aufgaben relativ offen gestellt. Besprechen Sie diese innerhalb Ihrer Gruppe und konkretisieren Sie die Aufgabenstellung. **Der erste Teil Ihrer Projektpräsentation** ist eine kurze Vorstellung Ihrer Aufgabe in einer Tutorübung in der Woche **12.12.2016-16.12.2016**. Wählen Sie bitte eine Übung, in der mindestens ein Team-Mitglied angemeldet ist.

Erscheinen Sie bitte **mit allen Team-Mitgliedern** und bereiten Sie einen Kurzvortrag mit folgenden Inhalten vor:

- 1 Folie: Vorstellung der Team-Mitglieder
- 2 Folien: Zusammenfassung der Aufgabenstellung

In der Übung wird eine Beamer vorhanden sein, an den Sie Ihr eigenes Notebook anschließen können. Alternativ können Sie auch Ihre Präsentation als PDF Datei auf einem USB-Stick mitbringen. Der Kurzvortrag wird von einer Person gehalten.

Bei Fragen/Unklarheiten in Bezug auf den Ablauf und die Aufgabenstellung wenden Sie sich bitte an Ihren Tutor.

Mit freundlichen Grüßen
Die Übungsleitung

PS: Vergessen Sie nicht, sich rechtzeitig in TUMonline zur Prüfung anzumelden. Dies ist Voraussetzung für eine erfolgreiche Teilnahme am Praktikum im laufenden Semester.

¹<http://wwwi10.lrr.in.tum.de/~buettner/GEP-ASP/GEP-ASP-Praktikumsordnung.pdf>

²<http://www.lrr.in.tum.de/lehre/>

2 Image Processing

2.1 Überblick

Im Zuge Ihrer Projektaufgabe werden Sie theoretisches Wissen aus der Mathematik im Anwendungszusammenhang verwenden, um einen Algorithmus auf Ihrem Raspberry Pi 3 implementieren. Sie konzentrieren sich dabei auf das Feld des *Image Processing*, in welchem Pixelbilder, wie sie typischerweise Digitalkameras produzieren, als Eingabe für bestimmte Algorithmen verwendet werden und mathematische Überlegungen dadurch sichtbar gemacht werden.

2.2 Funktionsweise

In dieser Aufgabe befassen Sie sich mit der Mandelbrotmenge. Diese ist gegeben durch iterative Anwendung der komplexen Gleichung

$$z_{i+1} = z_i^2 + c \quad (i \geq 0)$$

und Betrachtung des Wertes z_i innerhalb der komplexen Ebene. Dabei wählt man $z_0 = 0$ und Real- sowie Imaginärteil von c gemäß der Koordinaten in der komplexen Ebene. Durch Variation von a und b in $c = (a+bi)$ für $a \in [-2; 1]$ und $b \in [-1; 1]$ und Betrachtung von z_n für große n lässt sich so die charakteristische Mandelbrotmenge in der komplexen Ebene darstellen.

2.3 Aufgabenstellungen

Ihre Aufgaben lassen sich in die Bereiche Konzeption (theoretisch) und Implementierung (praktisch) aufteilen. Sie können (müssen aber nicht) dies bei der Verteilung der Aufgaben innerhalb Ihrer Arbeitsgruppe ausnutzen. Alle Antworten auf konzeptionelle Fragen sollten in Ihrer Ausarbeitung erscheinen. Besprechen Sie nach eigenem Ermessen außerdem im Zuge Ihres Vortrags einige der konzeptionellen Fragen. Die Antworten auf die Implementierungsaufgaben werden durch Ihrem Code reflektiert.

2.3.1 Theoretischer Teil

- Berechnen Sie Real- sowie Imaginärteil von z_{i+1} in Abhängigkeit von Real- sowie Imaginärteil von z_i explizit. Dies ist die Iterationsvorschrift, die Ihr Algorithmus verwenden soll.
- Sofern z_i beschränkt ist, färben Sie den Pixel, der zu Real- sowie Imaginärteil von c korrespondiert schwarz, da er zur Mandelbrotmenge gehört. Wie verfahren Sie in dieser Hinsicht mit Pixeln, die zu Koordinaten korrespondieren, für die z_i divergiert?
- Der Algorithmus lässt sich leicht parallelisieren. Entwerfen Sie einen parallelen Algorithmus, der die SIMD-Einheit des BeagleBoards verwendet.

2.3.2 Praktischer Teil

- Implementieren Sie in der Datei mit dem Assemblercode eine Funktion

```
void mandelbrot(unsigned int r_start, unsigned int r_end, unsigned  
int i_start, unsigned int i_end, unsigned int res, unsigned char  
*img)
```

welche den Ausschnitt der Mandelbrotmenge in der komplexen Ebene, der auf der reellen Achse durch `r_start` und `r_end` und auf der imaginären Achse durch `i_start` sowie `i_end` gekennzeichnet wird, übergeben bekommt. Als weitere Parameter erhält die Funktion die gewünschte Auflösung (anzahl der Samplepunkte) `res` sowie einen Zeiger auf einen Speicherbereich der groß genug ist, um die berechneten Bitmap-Daten des Ergebnisses zu speichern `img`. Legen Sie Ihrer Ausgabe eine sinnvolle Farbpalette zugrunde.

- Wie Sie wissen, dürfen Sie alle I/O-Operationen im Rahmenprogramm in C durchführen. Wir schreiben Ihnen kein Ausgabeformat vor, legen Ihnen aber nahe, mit unkomprimierten, 24-bit BMP-Dateien zu arbeiten. Das Schreiben des Headers und der Daten kann daher in C erfolgen.

2.3.3 Bonusaufgabe

Die direkte Implementierung der Iterationsvorschrift für die Mandelbrotmenge resultiert in Bildern, die stufenweise Artefakte aufweisen. Entwerfen und implementieren Sie zusätzlich einen Algorithmus, der Bilder ohne Artefakte generiert.

2.4 Checkliste

Die folgende Liste soll Ihnen als Gedächtnisstütze beim Bearbeiten der Aufgaben dienen. Sollten Sie eine Reverse-Engineering-Aufgabe erhalten haben, sind diese Punkte für Sie weitestgehend hinfällig.

- Verwenden Sie keinen Inline-Assembler.
- I/O-Operationen dürfen grundsätzlich in C implementiert werden.
- Sie dürfen die Signatur der in Assembler zu implementierenden Funktion nur dann ändern, wenn Sie dies (in Ihrer Ausarbeitung) rechtfertigen können.
- Denken Sie daran, das Laufzeitverhalten Ihres Codes zu testen (Sichere Programmierung, Performanz).
- Verwenden Sie NEON/SIMD-Befehle, wenn möglich.
- Fügen Sie Ihrem fertigen Quelltext Anweisungen hinzu, wie das Projekt kompiliert werden kann.

- Eingabedateien, welche Sie generieren, um Ihre Implementierungen zu testen sollten mit abgegeben werden.
- Bonusaufgaben (sofern vorhanden) müssen nicht implementiert werden.