Bericht Aufgabe 1

Nathalie Junker, Jannik Portz, Dennis Ritter

9. November 2015

Inhaltsverzeichnis

1 Aufgabenstellung

1.1 Einlesen eines Bildes

Es soll ein Programm geschrieben werden, das ein Bild von der Festplatte anzeigt. Beim Starten des Programms wird ein Dialog zum Wählen einer Bilddatei angezeigt. Wurde ein Bild ausgewählt, wird dieses in einem Fenster, das genauso groß ist wie das Bild, angezeigt.

1.2 Generierung eines Bildes

Es soll ein Programm geschrieben werden, das zunächst ein 640x480 Pixel großes Panel anzeigt, in dem auf einer schwarzen Grundfläche Pixel für Pixel eine rote Diagonale gezeichnet wird. Beim Verändern der Größe des Fensters soll diese Diagonale direkt verlängert bzw. verkürzt werden. Darüber hinaus soll das erzeugte Bild über einen Menüpunkt als Bilddatei abgespeichert werden können.

1.3 Matrix- und Vektorbibliothek

Basierend auf den vorgegebenen Klassendiagrammen sollen Klassen für eine quadratische Matrix, einen Vektor, einen Punkt und eine Normale (alle für den Einsatz im dreidomensionalen Raum) programmiert werden. Diese Klassen besit-

zen Methoden für mathematische Operationen mit Objekten der gleichen Klasse oder teilweise mit Objekten anderer Klassen der Mathematik-Bibliothek. Zu diesen Operationen gehören beispielsweise Multiplikation, das Bilden des Skalarprodukts von zwei Vektoren/Punkten/Normalen, die Reflexion eines Vektors an einer Normalen oder das Bilden des Kreuzprodukts zweier Vektoren. Darüber hinaus können bei der Matrix einzelne Spalten separat ausgetauscht werden. Alle der zu implementierenden Klassen sollen immutable sein, also nach der Initialisierung nicht mehr veränderbar.

2 Lösungsstrategie

Wir haben in unserer Gruppe versucht, die drei Teilaufgaben fair unter den Gruppenmitgliedern aufzuteilen. Darüber hinaus haben wir uns zwischendurch getroffen, um Fragen zu stellen, Probleme zu schildern, den anderen Gruppenmitgliedern die eigene Arbeit vorzustellen und zu erläutern und Tipps zu geben. Jeder hat sich die Arbeit der anderen Gruppenmitlieder angeschaut und ggf. an der ein oder anderen Stelle nachgebessert.

3 Implementierung

3.1 Einlesen eines Bildes

Um ein Bild anzuzeigen, haben wir die Klasse ChooseImageFrame geschrieben. Hier wird vorerst ein JFileChooser Dialog geöffnet, in dem der Benutzer eine Datei auswählen kann. Die auswählbaren Dateien haben wir durch einen FileFilter beschränkt, der nur Dateien akzeptiert, die auf .jpg oder .png enden. Außerdem akzeptiert er auch Verzeichnisse, damit der Benutzer durch den Öffnen-Button auch durch Verzeichnishierarchien navigieren kann.

Wenn eine entsprechende Datei geöffnet wurde, wird dem ChooseImageFrame ein ImagePanel hinzugefügt, das das Bild intern zeichnet und automatisch seine Größe an die Auflösung des ausgewählten Bildes anpasst.

3.2 Generierung eines Bildes

Wir haben ein ImageCanvas prgrammiert, das von java.awt.Canvas erbt. In der paint() Methode wird ein BufferedImage erzeugt über dessen einzelne Pixel (von links nach rechts und oben nach unten) iteriert. An Punkten, an denen der x und y Wert übereinstimmen, wird ein roter Pixel gemalt, an allen anderen ein schwarzer.

Das ImageCanvas wird dann in ein Panel und einen Frame eingesetzt, um das erzeugte Bild in einem Fenster anzuzeigen. Über den Menüpunkt 'Speichern' wird das Bild in eine png Datei geschrieben und auf der Festplatte gespeichert.

3.3 Matrix- und Vektorbibliothek

Die Mathematik-Bibliothek haben wir gemäß den Vorgaben in der Aufgabenstellung umgesetzt. Um sicherzugehen, dass die geforderten Abnahmekriterien zu jedem Zeitpunkt erfüllt sind, haben wir einen JUnit TestCase angelegt, in dem wir genau diese Szenarien durchlaufen. Somit können wir nach jeder Änderung

prüfen, ob diese durch nicht bedachte Abhängigkeiten die Abnahme gefährden. Darüber hinaus haben wir eine Klasse ParameterNullException geschrieben, die automatisch eine verständliche Exception-Message generiert, um Redundanz zu vermeiden.

4 Probleme / Schwierigkeiten bei der Bearbeitung

- Einarbeitung in LaTex
- Verteilte Zusammenarbeit mit git. Oftmals sind bei uns merge-Konflikte aufgetreten, die wir aus der Welt schaffen können, indem wir die Arbeit besser aufteilen bzw. Zeitfenster zur Optimierung der Lösung der anderen Gruppenmitglieder festlegen.
- Algorithmus zur Reflexion eines Vektors: Bei der Recherche im Internet sind uns viele verschiedene Formeln über den Weg gelaufen und wir hatten Probleme, die geeignetste zu finden und diese mit unserer Bibliothek zu implementieren.