

COMPUTERGRAFIK 1. BERICHT ZUR AUFGABE 1

ALEXANDER BUYANOV (806984) AND PHILLIP REDLICH (791806)

1. BERICHT

Bei der Vorbereitung zum praktischen Teil der ersten Übung sollten wir uns mit der verteilten Versionsverwaltung von Dateien mit "GIT" vertraut machen, die java.awt.image API kennenlernen und unsere Mathematikkenntnisse zum Thema Vektoren und Matrizen auffrischen.

Im ersten Anwendungsteil der Aufgabe mussten wir zwei kleine Programme schreiben.

Das erste Programm sollte ein Bild von der Festplatte öffnen, und es in einem Fenster anzeigen. Um ein Bild zu suchen haben wir die JFileChooser Klasse aus der JAVA API benutzt. Darüber hinaus haben wir ein Filter für die Auswahl der JPG und PNG Bilder verwendet. Die erste Methode run() wartet auf das Bild, das der User von der Festplatte auswählen muss. Wenn das Bild ausgewählt wurde, startet die zweite Methode showImage(), welche das Image mit Hilfe der BufferedImage Klasse im Fenster anzeigt. Diese Aufgabe war für uns, programmiertechnisch und zeitlich, kein großes Problem und hat uns gut an die Thematik herangeführt.

Das zweite Programm sollte ein schwarzes Bild mit einer diagonalen roten Linie erzeugen ohne triviale Methoden von der Graphics-Klasse zu verwenden. Hier war auch eine SWING-Komponente zu verwenden. Das Programm kann darüberhinaus das erzeugte Bild auf der Festplatte als eine Datei im PNG-Format speichern. Für die Erzeugung eines Bildes haben wir die Klassen BufferedImage, WritableRaster und ColorModel benutzt. An diesem Punkt in der ersten Übung sind wir auf kleine Probleme gestoßen, aber nachdem wir verstanden haben wie die Bild-Erzeugung funktioniert, haben wir unser Ziel erreicht. Um das Bild speichern zu können, haben wir die Klasse ImageSaver erstellt. Nachdem der Benutzer auf den Menüpunkt ?Speichern? klickt, wird eine neue Instanz der ImageSaver-Klasse erzeugt. Der Nutzer kann den Pfad in dem das Bild gespeichert werden soll wählen. Es wird überprüft, ob das Bild im PNG-Format ist und wenn ja, wird es erfolgreich auf der Festplatte gespeichert.

Der letzte Teil der ersten Übung beinhaltete die Implementierung von einer Vektoren- und Matrizen-Bibliotheken. Dazu sollten die Rechenmethoden der Objekte als Methoden umgesetzt werden. Um dies zu erreichen haben wir zuerst die Musterklassen erstellt ohne die Methoden zu überschreiben. Danach haben wir begonnen tatsächliche Berechnungen einzufügen. Wir haben hier mit den einfachen Rechenmethoden begonnen und die etwas anspruchsvolleren Formeln später

zunächst nachvollzogen und ebenfalls implementiert. Die Implementierung selbst wies wenig Besonderheiten auf. Sie setzt lediglich die tatsächliche Berechnung in Code um. Die einzige wirkliche Schwierigkeit lag für uns in der Implementierung der Methode `reflectedOn()`, da hier die Methoden zusammen angewendet werden mussten, was teilweise durch die Verschachtelung zu Fehlern führte. Dies klärte sich nach erster falscher Implementierung jedoch schnell auf. Der Zeitbedarf war grundsätzlich nicht allzu groß, was die reine Implementierung angeht. Das testen hat z.B. länger gedauert.

Anhand diesen Aufgaben haben wir die Grundlagen von der Bilderzeugung in der Programmierung kennengelernt und unser Grundverständnis für Matrizen und Vektoren gefestigt.