

## Medientechnik - Programmierung 2

### Lernziele

- Farbräume
- Transformation in Java2D
- Swing-Kenntnisse vertiefen
- Java2D kennenlernen

### Aufgaben (20 Punkte)

#### Aufgabe 1: Farbräume (3 Punkte)

Erklären Sie die Eigenschaften der Farbräume HSL, RGB und CMYK und welche Werte man in den jeweiligen Modellen zur Definition einer Farbe angeben muss.

Speichern Sie Ihre Antworten in eine Datei **aufgabe1.txt**.

#### Aufgabe 2: Transformation in Java2D (4 Punkte)

Java2D ermöglicht die einfache Nutzung der drei Transformationsarten Translation, Rotation und Skalierung für beliebige geometrische Objekte.

1. Recherchieren Sie, was Translation, Rotation und Skalierung in der Computergraphik bedeuten und erklären Sie jede dieser Transformationen in jeweils mindestens einem Satz.
2. Probieren Sie Folgendes entweder direkt in Java2D oder auf einem Blatt Papier aus: Gegeben ist ein Quadrat mit einer Seitenlänge von 50 Pixeln. Dieses wird im Ursprung des Koordinatensystems (linke obere Ecke im Punkt 0,0) gezeichnet.
  - a. Davor wird der Befehl `translate(50, 50)` gefolgt von `rotate(Math.toRadians(45))` ausgeführt. An welchen Koordinaten befindet sich in der finalen Darstellung die linke obere Ecke des ursprünglichen Quadrats?
  - b. Jetzt wird die Reihenfolge dieser Transformationsbefehle vertauscht. Landet die linke obere Ecke des ursprünglichen Quadrats diesmal an einer anderen Stelle oder hat die Vertauschung der Transformationen keinen Einfluss darauf?

Speichern Sie Ihre Antworten in der Textdatei **aufgabe2.txt** und fügen Sie sie Ihrer Abgabe bei.

### Aufgabe 3: Polygonzeichner mit Java2D (12 Punkte)

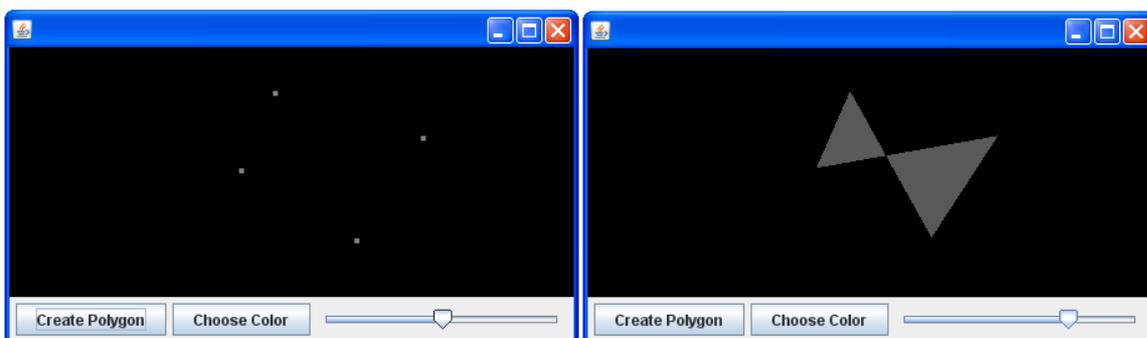
Realisieren Sie eine Java SWING Anwendung, die die Java2D Funktionalität nutzt. Dabei lernen Sie neue Interaktionsformen wie Reaktion auf Mouse-Events, das Einbinden von Slidern und das Zeichnen von 2D-Objekten kennen.

Die Punktzahl für die Teilaufgaben sind jeweils angegeben, die Restpunkte werden für eine lauffähige Abgabe und guten Programmierstil (z.B. Kommentare) vergeben.

Folgende Funktionalität soll von Ihrer Anwendung unterstützt werden:

- **In einem schwarzen Panel soll der Benutzer Punkte mit der Maus einzeichnen können. (3 Punkte)**  
Erstellen Sie eine Unterklasse von JPanel, in der Sie auf Mouse-Events reagieren. Überschreiben Sie die paint()-Methode und lassen Sie die Punkte und das Polygon dort zeichnen. Die eigentlichen Daten (also Punkte, Farbe, Transparenz etc.) soll wieder in einem Model gespeichert werden (also wieder MVC-Prinzip mit Observer und Observable).  
Dieses Panel müssen sie nun in einen größeren Container, z.B. ein JFrame, einfügen.
- **Über einen Button „Create Polygon“ wird aus diesen Punkten ein Polygon gezeichnet. (1 Punkt)**  
Die Graphics2D Klasse stellt eine Methode `void draw(Shape s)` zur Verfügung. Ein Shape hat mehrere Unterklassen, u.a. Polygon.
- **Über einen Button „Choose Color“ kann man die Füllfarbe ändern. (2 Punkte)**  
Binden Sie die Klasse JColorChooser aus dem javax.swing-Paket in Ihre Anwendung ein. Wählt ein Benutzer dort eine Farbe aus, soll das Polygon entsprechend abgeändert werden.
- **Über einen Slider (javax.swing.JSlider) kann man die Transparenz des Polygons ändern. (3 Punkte)**  
Die Transparenz eines 2D-Objektes kann man ändern, indem man das Graphics-Objekt vor dem Zeichnen ändert (Stichwort AlphaComposite). Für den JSlider benötigen Sie wieder einen Listener, der auf Änderungen des Sliders reagiert.

So sollte die Anwendung in etwa aussehen:



Schritt 1: Punkte zeichnen

Schritt 2: Polygon zeichnen

### **Bonusaufgabe – Abstract Art (3 Punkte)**

Man soll mehr als ein Polygon zeichnen und verändern können. Nach Zeichnen eines Polygons kann das nächste gezeichnet werden. Durch Anklicken eines Polygons kann man dessen Farbe und Transparenz ändern.

Speichern Sie Ihre Dateien wie unten angegeben ab und fügen Sie sie Ihrer Abgabe bei.

### **Abgabe (1 Punkt)**

Noch einige Hinweise zur Abgabe (um die Korrektur zu erleichtern):

- Die Textaufgaben (1+2) sollen mit dem entsprechenden Titel als .txt oder .pdf abgegeben werden
- Zur Programmieraufgabe sollen nur die Source-Dateien (.java) abgegeben werden. Zusätzlich soll eine lauffähige jar-Datei abgegeben werden. Diese erstellen Sie z.B. mit Eclipse unter „File“-> „Export“ -> „Java“ -> „Runnable JAR file“.
- Insgesamt soll die Abgabe dann so aussehen (bitte auch Ordnerstruktur beachten, natürlich sinnvollere Java-Klassennamen wählen):



- Zu guter Letzt noch als .zip-Datei verpacken und in UniWorX hochladen
- Als Motivation: Für eine richtige Abgabe gibt es ab sofort einen Punkt ☺

Geben Sie diese Lösung bis zum 22.05.08, 10:00 Uhr im UniWorX Portal (<http://www.pst.ifi.lmu.de/uniworx>) ab.