Media Informatics /
Graphics & Geometric Computing

Prof. Dr. Marcel Campen
Steffen Hinderink

UNIVERSITÄT OSNABRÜCK

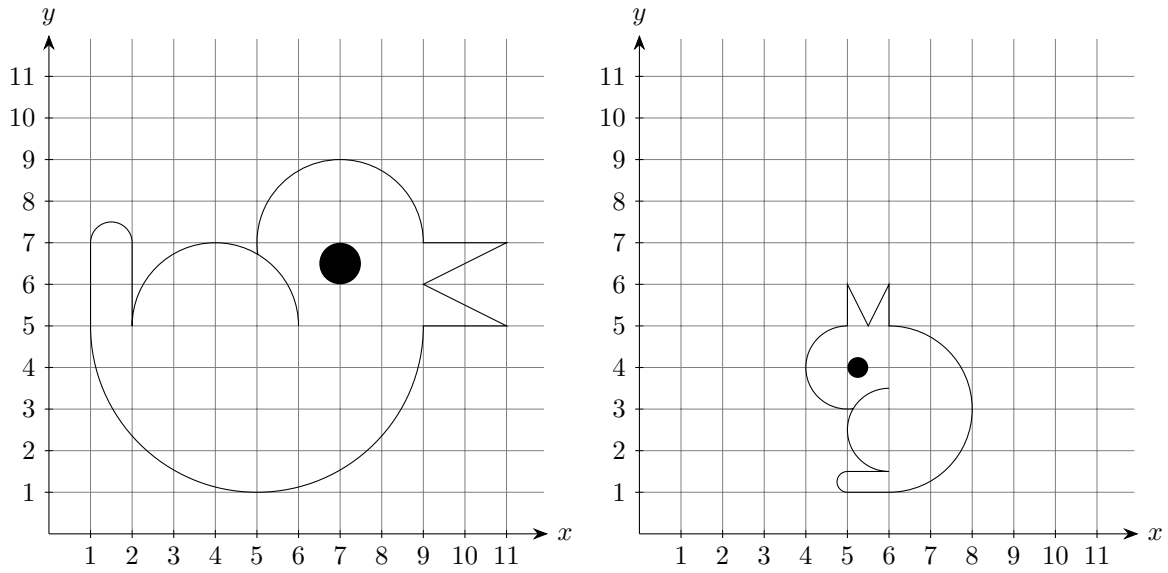
Computergrafik (SoSe 2023)

Blatt 1

fällig Sonntag **30. April**, 24:00

Verspätet eingereichte Abgaben können nicht berücksichtigt werden.

Aufgabe 1 (Theorie: Lineare & Affine 2D Transformationen)



(a) (7P) Bestimmen Sie eine Reihe von vier elementaren Transformationsmatrizen (Translation, Rotation, Skalierung (uniform oder nicht-uniform), Translation), so dass das Produkt dieser Matrizen die links dargestellte *Ente* auf den rechts dargestellten *Hasen* abbildet! Geben Sie sowohl die vier elementaren Matrizen als auch ihr Produkt, d. h. die finale Transformationsmatrix, an! Benutzen Sie dabei keine Abkürzungen oder symbolische Schreibweisen für die Matrizen, sondern geben Sie die einzelnen Einträge der Matrizen jeweils explizit an. Denken Sie daran, erweiterte Koordinaten zu benutzen. Wir nutzen, wie auch in der Vorlesung, die Konvention, dass Punkte/Vektoren von rechts an Matrizen multipliziert werden.

(b) (3P) Geben Sie die Bilder der Einheitsvektoren ($(1,0,0)$ und $(0,1,0)$) sowie des Ursprungs unter dieser Transformation an, d.h. die Vektoren oder Punkte, auf die diese durch die Transformation abgebildet werden.

Media Informatics /
Graphics & Geometric Computing

Prof. Dr. Marcel Campen
Steffen Hinderink

UNIVERSITÄT OSNABRÜCK

Aufgabe 2 (Praxis: Lineare & Affine 2D Transformationen)

Öffnen Sie die Datei 01.html in einem Browser und in einem Editor (beliebiger reiner Text-Editor oder JavaScript-Editor/Entwicklungsumgebung). Wann immer Sie im Folgenden Ergänzungen am JavaScript-Quelltext in dieser Datei vorgenommen und gespeichert haben, aktualisieren Sie die Seite im Browser um das Ergebnis zu sehen.

Ergänzen Sie den vorgegebenen Quelltext in dieser Aufgabe nur wie unten angegeben. Nehmen Sie keine Änderungen am bereits vorhandenen Quelltext vor.

(a) (3P) Ergänzen Sie die Funktion `createPoints`, die ein Array von 2D-Punkten erzeugt. Die Funktion soll mindestens vier Punkte zurückgeben. Diese können eine beliebige Figur bilden – Sie können die Punkte "hardcoden" oder prozedural erzeugen. Die Koordinaten der Punkte sollten in allen Dimensionen auf den Wertebereich von -250 bis 250 beschränkt sein. Der Ursprung liegt in der Mitte des Canvas-Elements.

Für die Repräsentation eines Punktes ist die Klasse `Point` bereits vordefiniert. Ein neuer Punkt kann also durch `new Point(x, y)` erzeugt werden. Die zusätzliche `1` (erweiterte Koordinaten) wird dabei intern automatisch angehangen. Ein solcher Punkt kann dann per `.push(...)` an das Array angehangen werden.

Öffnen Sie die Datei nun in einem Browser, sollten Sie eine Darstellung der Punkte sehen.

(b) (7P) Ergänzen Sie die Funktion `transformationMatrix`, die eine 3×3 Transformationsmatrix (wie unten definiert) zurückgibt, mit der 2D-Punkte in erweiterten Koordinaten transformiert werden können. Diese wird dann auf die definierten Punkte angewendet, so dass sich die Darstellung der Punkte im Browser entsprechend ändert.

Die zu berechnende Transformation soll Translation, nicht-uniforme Skalierung und Rotation gegen den Uhrzeigersinn (in dieser Reihenfolge) vereint durchführen. Nutzen Sie dabei als Translationsvektor, Skalierungsfaktoren, und Rotationswinkel die Werte, die Sie in folgenden Variablen finden:

- `transX`
- `transY`
- `scaleX`
- `scaleY`
- `rot`

Die Werte dieser Variablen werden automatisch verändert wenn die Slider, die Sie im Browser unterhalb des Canvas-Elements finden, betätigt werden, so dass Sie interaktiv durchprobieren können, ob Ihre Transformation sich wie erwartet verhält.

Für die Verwaltung der Matrizen verwenden Sie die Klasse `Matrix` und ihre Methoden.

Aufgabe 3 (Bonus: Kolorierung der Punkte)

Modifizieren Sie den Code so, dass den einzelnen Punkten Farben zugeordnet werden (z. B. "hardcoded", abhängig von der Entfernung zum Ursprung, abhängig von der Punktnummer oder eines sonstigen von Ihnen gewählten Kriteriums), und diese dann beim Zeichnen der Punkte genutzt werden. Für das Umschalten steht die Variable `colored` zur Verfügung.