

Wintersemester 2013/14

Übungen zu Computergraphik - Blatt 2

Abgabe am 29. 10. 2013

Aufgabe 1 (Vektoren, 2 Punkte)

Finde einen Vektor, der orthogonal zu $\mathbf{v} = (-1, 2)$ ist.

Aufgabe 2 (Vektoroperationen, 2+2 Punkte)

Gegeben seien zwei Vektoren $\mathbf{b}, \mathbf{c} \neq 0$ im \mathbb{R}^n .

a) Geben Sie an, welche Bedingung Vektor \mathbf{a} erfüllen muss, damit

$$\mathbf{a} \times \mathbf{b} = \mathbf{a} \times \mathbf{c}$$

gilt. (Es gibt noch eine weitere Lösung außer der trivialen Lösung $\mathbf{a} = 0!$) Dies zeigt, dass $\mathbf{a} \times \mathbf{b} = \mathbf{a} \times \mathbf{c} \not\Rightarrow \mathbf{b} = \mathbf{c}!$

b) Zeigen Sie: falls $\forall \mathbf{a} : \mathbf{a} \cdot \mathbf{b} = \mathbf{a} \cdot \mathbf{c}$ dann folgt daraus $\mathbf{b} = \mathbf{c}$.

Aufgabe 3 (Abstandsberechnung, 4 Punkte)

Gegeben zwei Geraden $L_1 : \mathbf{x} = \mathbf{Q}_1 + t \mathbf{d}_1$ und $L_2 : \mathbf{x} = \mathbf{Q}_2 + s \mathbf{d}_2$ im Raum; Geben Sie einen genauen Lösungsweg zur Bestimmen des minimalen Abstandes der Geraden voneinander an.

Aufgabe 4 (C++-Programmierung, 5 Punkte)

Die folgenden Aufgabenteile sollen in das Framework (siehe Webpage) integriert werden. Die Dateinamen sind gleich benannt wie die Klasse. Schauen Sie sich dazu zuerst die Folien zur C++-Wiederholung auf der VL-Homepage an und bei Bedarf weitere Quellen zu C++ (Links finden siehe auch auf der VL-Homepage). Wie in den C++ Folien erklärt, sollen die Deklarationen in eine *.h und die Definition in eine *.cpp Dateien geschrieben werden.

Implementieren Sie eine Klasse `Vec3` zur Repräsentation von 3D-Vektoren mit Floating-Point-Zahlen als Werten. Die Klasse sollte folgende Funktionalität bieten:

```
class Vec3
{
public:
    Vec3( ); // constructor, initialized each component with 0
```

```

    Vec3( float x, float y, float z ); // constructor, initialize with (x,z,y)
    Vec3( const Vec3& v ); // copy constructor

    float normalize( );
private:
    // Ueberlegen Sie sich die interne Instanzvariable
};

Vec3 cross( const Vec3& lhs, const Vec3& rhs );
float norm2( const Vec3& v ) ;

```

Die Wirkung der Funktionen sei wie folgt:

- `Vec3 cross(const Vec3& lhs, const Vec3& rhs);` Berechnet das Kreuzprodukt/Vektorprodukt zweier Vektoren.
- `float norm2(const Vec3& v) ;` Gibt die euklidische Norm des Vektors zurück.
- `float normalize();` Normalisiert einen Vektor und gibt zusätzlich die ursprüngliche Länge des Vektors zurück.

Tip: Schreiben Sie ein kleines Programm zum Testen Ihrer Klassen. Verwenden Sie sinnvolle Eingabewerte für Ihre Tests.