3. Praktikum

Jörn Loviscach

Versionsstand: 14. Juni 2020, 21:23

Ein Windpark soll in 3D visualisiert werden.

Legen Sie in der XAML-Datei im `Grid` einen `Viewport3D` an. Geben Sie diesem einen Namen, aber keine Breite und keine Höhe, so dass er automatisch das Fenster füllt.

`Viewport3D` erlaubt, statische und animierte 3D-Szenen zu erstellen. Das Standardkoordinatensystem liegt dabei so: Der Ursprung liegt in der Mitte des `Viewport3D`, die x-Achse zeigt nach rechts, die y-Achse nach oben (!) und die z-Achse aus dem Bildschirm heraus (rechtshändiges System). Konstruieren Sie Ihre Szene so, dass die Zahlenangaben aller 3D-Koordinaten in der Einheit Meter sind.

Im Konstruktor von `MainWindow` geben Sie dem `Viewport3D` eine `PerspectiveCamera`. Diese benötigt diverse Einstellungen: kleinstmöglicher und größtmöglicher Abstand, Position, Richtung senkrecht nach oben, Blickrichtung, Sichtfeld.

Dem `Viewport3D` geben Sie als Kind ein neues Objekt der Klasse `ModelVisual3D` und diesem als `Content` ein neues Objekt der Klasse `Model3DGroup`. Dieses kann als Kinder zum Beispiel Lichter, `GeometryModel3D`-Objekte und weitere `Model3DGroup`-Objekte enthalten (entsprechend zu Ordnern in Ordnern).

Erzeugen Sie ein Licht-Objekt der Klasse `DirectionalLight`. Stellen Sie dessen Farbe und Richtung ein.

Legen Sie in der Klasse `MainWindow` eine noch leere Methode `MeshGeometry3D ErzeugeRöhre(double länge, double radius)` und eine noch leere Methode `Model3DGroup ErzeugeTurbine(double x, double z, double nabenhöhe, double rotordurchmesser)` an.

Rufen Sie im Konstruktor von `MainWindow` in einer doppelten `for`-Schleife die Methode `ErzeugeTurbine` so auf, dass in der `Model3DGroup` des `ModelVisual3D` 5×5 Turbinen angelegt werden.

Füllen Sie die Funktion `ErzeugeRöhre(double länge, double radius)` testweise zunächst so, dass sie keine Röhre erzeugt, sondern ein Rechteck, das sich in x-Richtung von `0` bis `länge` erstreckt und in y-Richtung von `-radius` bis `+radius`.

Dazu erzeugen Sie eine `MeshGeometry3D`, dem Sie die vier Eckpunkte als `Positions` und die vier jeweiligen Normalenvektoren als `Normals` hinzufügen. Außerdem haben Sie anzugeben, wie der Grafikchip aus den vier Eckpunkten Dreiecke bilden soll. Dazu fügen Sie an `TriangleIndices` für jedes Dreieck die Indizes seiner drei Eckpunkte hinzu. Die Reihenfolge der Indizes muss dabei so sein, dass dass Dreieck von vorne (d. h. von außen) betrachtet gegen den Uhrzeigersinn durchlaufen wird.

Füllen Sie nun die Methode `ErzeugeTurbine` testweise zunächst so, dass Sie dort nur den Turm anlegen. Erzeugen Sie dazu eine `Model3DGroup`, geben Sie dieser eine `TranslateTransform3D`, um die Turbine an der richtigen Stelle zu platzieren.

Für die weiße Oberfläche erzeugen Sie ein `DiffuseMaterial` mit einer weißen `Brush`. Geben Sie einem neuen `GeometryModel3D` dieses Material und als `Geometry` eine Röhre aus Ihrer Methode `ErzeugeRöhre`. Stellen Sie diese Röhe aufrecht, indem sie der `GeometryModel3D` eine `RotateTransform3D(new AxisAngleRotation3D(new Vector3D(0.0, 0.0, 1.0), 90.0))` geben.

Behandeln Sie nun das `KeyDown`-Ereignis des `Window`, so dass die Kameraposition mit den vier Cursortasten verändert werden kann.

Komplettieren Sie nun die Methode `ErzeugeRöhre` so, dass sie statt nur eines Vierecks wirklich Röhren aus Hunderten von Dreiecken erzeugt.

Komplettieren Sie dann die Methode `ErzeugeTurbine` so, dass sie zusätzlich einen Rotor aus drei Röhren erzeugt.

Geben Sie dem Rotor eine Dreh-Animation auf Basis des folgenden Musters:

```
DoubleAnimation animation = new DoubleAnimation();
animation.From = 0.0;
animation.To = - 360.0;
animation.Duration = new Duration(TimeSpan.FromSeconds(5.0);
animation.RepeatBehavior = RepeatBehavior.Forever;
drehung.BeginAnimation(AxisAngleRotation3D.AngleProperty,
                                                    animation);
```

Diese Anleitung ist bewusst nicht vollständig – als Übung dafür, die automatische Code-Vervollständigung zu nutzen und die richtigen Fragen an die Suchmaschinen zu stellen.