

Übung 04a: Objektorientierte Programmierung in Python

In dieser Übung wird die Animation aus Übung 3b (3D-Visualisierung mit Vtk) wieder aufgegriffen. Die Funktionen zur Simulation des Systems sowie der Berechnung von Position und Orientierung der Körper sind bereits vorgegeben. Diesmal sollen die Visualisierungsobjekte als eigene Klassen implementiert werden. Dazu wurde in der Datei `primitives.py` bereits die Basisklasse `Geometry` angelegt. Diese ist zunächst zu vervollständigen. Danach sollen Klassen für Quader, Kugel und Stl-Datei implementiert werden, die von dieser Basisklasse erben. Schließlich sind noch einige wenige Anpassungen im Hauptskript (`visualisierung.py`) vorzunehmen um die neuen Klassen verwenden zu können.

Übungsaufgaben

1. Erweitern Sie die Klasse `Geometry` um die Methoden `setOrientation` (Matrix als Argument) und `setPositionAndOrientation` (Vektor und Matrix als Argumente)
2. Erweitern Sie die Klasse `Geometry` um die Methoden `setColor` und `setOpacity` (je ein Argument). Diese vereinfachen den Zugriff auf die Eigenschaften der intern verwendeten `vtkLODActor` Instanz. Die Farbe und Transparenz werden mit `actor.GetProperty().SetColor(...)` bzw. `actor.GetProperty().SetOpacity(...)` gesetzt.

3. Erstellen Sie eine Klasse `Sphere` in Analogie zur Klasse `Cube`, die von `Geometry` erbt und als Quelle `tk.VtkSphereSource` nutzt. Als Argumente im Konstruktor sollen der radius und optional die Auflösung der Geometrie stehen. Zur Erinnerung:

```
part = vtk.vtkSphereSource()
part.SetRadius(5)
part.SetThetaResolution(20)
part.SetPhiResolution(20)
```

4. Erstellen Sie in gleicher Weise eine Klasse `File`, die im Konstruktor einen Pfad zu einer Datei erwartet. Erinnerung:

```
part = vtk.vtkSTLReader()
part.SetFileName("part.stl")
```

5. Legen Sie in Hauptskript zwei Körper für Laufkatze (Quader mit $0.3 \times 0.1 \times 0.1$) und Last (Kugel mit $r=0.03$) als Instanzen Ihrer Klassen an, übergeben Sie diese an den Renderer und nutzen Sie die von Ihnen implementierte Methode `setPositionAndOrientation` zum Aktualisieren der Körper innerhalb der Funktion `updateScene`.
6. Tauschen Sie die Geometrieprimitive der Körper nach Belieben aus und variieren Sie Farbe und Transparenz der Objekte. Im Unterverzeichnis `<data>` befindet sich eine stl-Datei für die Visualisierung der Laufkatze.

