

Übung 03b: 3D-Visualisierungen mit Python

Ziel der heutigen Übung ist das Erstellen einer 3D-Animation der Laufkatze (verschiebbarer Körper mit Pendel). Dazu sind drei Python-Dateien vorgegeben:

model.py

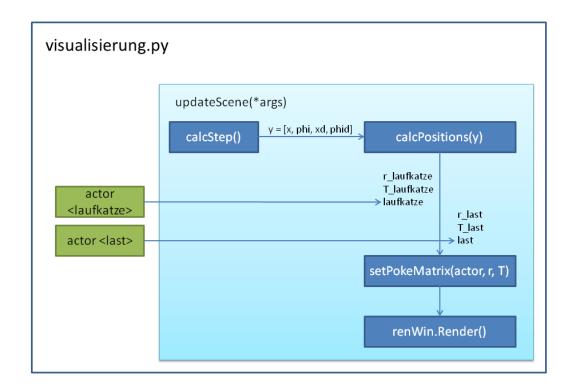
Enthält das Differentialgleichungssystem (Funktion rhs) und den Solver sowie eine Funktion zur Berechnung eines einzelnen Integrationsschrittes (calcStep).

functions.py

Enthält die Funktion calcPositions zur Berechnung der Position und Orientierung der Körper aus dem Simulationsergebnis sowie die Funktion setPokeMatrix, welche die Lageinformation an die Geometrieobjekte übergibt.

visualisierung.py

Das Hauptskript. Hier werden die Geometrieprimitive angelegt und die Szene erstellt. Dazu ist die Funktion updateScene vorgegeben. Um eine Animation zu erhalten, muss sie wiederholt aufgerufen werden. Innerhalb von updateScene wird ein Simulationsschritt gerechnet, aus dem Ergebnis die Lage der Körper berechnet, diese wiederum an die Geometrieobjecte (actors) übergeben und schlieSSlich das Bild neu gerendert, siehe Ablaufschema:





Aufgaben

 Vervollständigen Sie die Funktion calcPositions(y) in der Datei functions.py. Berechnen Sie die Lage und Orientierung der Laufkatze und der Last aus dem Ergebnis y eines Simulationsschrittes. Da die Laufkatze nur translatorisch verfahren kann, entspricht ihre Orientierung der 3x3 Einheitsmatrix (np.eye(3)). Die Position der Last berechnet sich nach der Formel:

$$r = [x + l\sin(\varphi), 0, -l\cos(\varphi)] \tag{1}$$

Zur Berechnung der Orientierungsmatrix R_y der Last siehe Folie 10.

Hinweis: Orientieren Sie sich bei der Variablenbezeichnung am Rückgabewert der Funktion (vorgegeben).

- 2. Vervollständigen Sie die Funktion setPokeMatrix(actor, r, T) in der gleichen Datei. Hier ist die Matrix poke elementweise mit den Werten aus r und T zu befüllen (siehe Folie 10).
- 3. Legen Sie in der Datei visualisierung.py einen Quader für die Laufkatze an (x=0.3,y=0.1,z=0.1). Fügen Sie einen Mapper und einen Actor für den Quader hinzu
- 4. Wiederholen Sie Punkt 3 für die Last. Erzeugen Sie aber diesmal einen Würfel mit der Kantenlänge 0.1
- 5. Fügen sie beide actors zum Renderer hinzu (ren.AddActor(...))
- 6. Importieren Sie die benötigten Funktionen zur Aktualisierung der Szene (siehe Schema oben)
- 7. Vervollständigen Sie die Funktion updateScene, beachten Sie dabei die Rückgabewerte der einzelnen Funktuionen und die Reihenfolge der Argumente!
- 8. Testen Sie die Animation. Der Befehl iren.CreateRepeatingTimer(20) gibt das Aktualisierungsintervall in ms an, also die Geschwindigkeit der Animation.

Zusatzaufgaben

- 1. Erzeugen Sie einen Zylinder der das Seil darstellt (r=0.01, h=0.5). Der Zylinder erhält die gleiche Orientierung wie die Last, muss jedoch noch entsprechend der Pendellänge verschoben werden (Position r extra berechnen)
- 2. Ersetzen Sie die Visualisierung der Laufkatze durch die stl-Datei im Verzeichnis /data. Fügen Sie auSSerdem den Haken als stl-Datei hinzu. Nutzen Sie dazu den vtkSTLReader (s. Folien 13). Der Haken erhält die selbe Position und Orientierung wie die Laufkatze.