

Diff: Übung 04b

Unterschiede zwischen Vorgabe und Lösung

Um die Musterlösung leichter nachvollziehbar zu gestalten wurden die Unterschiede zur Quelltext-Vorgabe mit Hilfe des Programms `diff` grafisch aufbereitet.

Legende:

- Grau: unveränderter Text (nur auszugsweise)
- Grün: neue Zeilen
- Gelb: veränderte Zeilen
- Rot: gelöschte Zeilen (kommt hier nicht vor)

Hinweis: nicht aufgeführte Dateien wurden nicht verändert.

Dieses Dokument wurde mit Hilfe von [diff2html](#) erstellt.

diff -u ../kurs04b/uebung/code/functions.py ../kurs04b/uebung/loesung/functions.py

../kurs04b/uebung/code/functions.py	../kurs04b/uebung/loesung/functions.py
11 def calcPositions(y):	11 def calcPositions(y):
12 """	12 """
13 Position und Orientierung der Koerper im Raum berechnen (im raumfesten	13 Position und Orientierung der Koerper im Raum berechnen (im raumfesten
14 Koordinatensystem). y ist das Ergebnis der Integration [x, phi, xd, phid]..	14 Koordinatensystem).
15	15
16 r - Position (3-Vektor)	16 r - Position (3-Vektor)
17 T - Orientierung (3x3-Matrix)	17 T - Orientierung (3x3-Matrix)
22 # Geometrieparameter - Pendellaenge	22 # Geometrieparameter - Pendellaenge
23 l = 0.5	23 l = 0.5
24	24
25 # Zustaende x und phi aus Argument - Geschwindigkeiten nicht benoetigt.	25 # Zustaende auf Poitionsebene - Geschwindigkeiten werden nicht benoetigt.
26 # ...	26 x = y[0].
	27 phi = y[1]
27	28
28 # Lage und Orientierung der Laufkatze berechnen	
29 # ...	
30	29
31 # Lage und Orientierung der Last.	30 # Lage der Laufkatze.
32 # ...	31 r_laufkatze = np.array([x,0,0]).
	32 T_laufkatze = np.eye(3)
	33
	34
	35 # Lage der Last
	36 r_last = np.array([x+l*np.sin(phi),0,-l*np.cos(phi)])
	37 T_last = np.array([[np.cos(phi), 0, -np.sin(phi)],
	38 [0, 1, 0],
	39 [np.sin(phi), 0, np.cos(phi)]])
33	40
34 return (r_laufkatze, T_laufkatze, r_last, T_last)	41 return (r_laufkatze, T_laufkatze, r_last, T_last)
35	42
55 poke = vtk.vtkMatrix4x4()	62 poke = vtk.vtkMatrix4x4()
56	63
57 # Matrix elementweise befuellen	64 # Matrix elementweise befuellen
58 # for...	65 for row in range(3):
59	66
60 # Positionsvektor	67 # Positionsvektor
61 .	68 poke.SetElement(row,3, r[row]).
62	69
63 # Orientierung	70 # Orientierung
64 .	71 for col in range(3):
	72 poke.SetElement(row, col, T[row,col])
65	73
66 # Matrix an Actor uebergeben	74 # Matrix an Actor uebergeben
67	75

```
6 / actor.PokeMatrix(poke)
```

```
\ No newline at end of file
```

```
Only in ../kurs04b/uebung/loesung/: functions.pyc
```

```
Only in ../kurs04b/uebung/code/: interaktiv.py
```

```
Only in ../kurs04b/uebung/code/: laufkatze.stl
```

```
Only in ../kurs04b/uebung/loesung/: model.pyc
```

```
diff -u ../kurs04b/uebung/code/visualisierung.py ../kurs04b/uebung/loesung/visualisierung.py
```

```
../kurs04b/uebung/code/visualisierung.py
```

```
1  
2  
3  
4  
5  
6 from vtk.util.colors import *  
7  
8  
9 # importe einfuegen.  
10 # ...  
11  
12  
13 #-----  
14  
15 # Quader fuer Laufkatze mit Abmessungen x,y,z anlegen  
  
  
  
  
  
  
  
  
  
16  
17  
18 #-----  
19  
20 # Wuerfel fuer Last
```

```
75 actor.PokeMatrix(poke)
```

```
../kurs04b/uebung/loesung/visualisierung.py
```

```
1  
2  
3  
4  
5  
6 from vtk.util.colors import *  
7  
8  
9 from model import calcStep.  
10 from functions import calcPositions, setPokeMatrix.  
11  
12  
13 #-----  
14  
15 # Quader fuer Laufkatze mit Abmessungen x,y,z anlegen  
16 lkSource = vtk.vtkCubeSource()  
17 lkSource.SetXLength(0.3)  
18 lkSource.SetYLength(0.1)  
19 lkSource.SetZLength(0.1)  
20  
21 # Mapper  
22 lkMapper = vtk.vtkPolyDataMapper()  
23 lkMapper.SetInputConnection(lkSource.GetOutputPort())  
24  
25 # Actor  
26 laufkatze = vtk.vtkLODActor()  
27 laufkatze.SetMapper(lkMapper)  
28  
29  
30 #-----  
31  
32 # Wuerfel fuer Last  
33 lastSource = vtk.vtkCubeSource()  
34 lastSource.SetXLength(0.05)  
35 lastSource.SetYLength(0.05)  
36 lastSource.SetZLength(0.05)  
37  
38 # Mapper  
39 lastMapper = vtk.vtkPolyDataMapper()  
40 lastMapper.SetInputConnection(lastSource.GetOutputPort())  
41  
42 # Actor  
43 last = vtk.vtkLODActor()  
44 last.SetMapper(lastMapper)  
45
```

	46 last.GetProperty().SetColor(red)
21	47
22	48
23 #-----	49 #-----
:	:
30 iren.SetRenderWindow(renWin)	56 iren.SetRenderWindow(renWin)
31	57
32 # Actors zum Renderer hinzufuegen	58 # Actors zum Renderer hinzufuegen
33 # ...	59 ren.AddActor(laufkatze).
	60 ren.AddActor(last)
34	61
35 # Hintergrundfarbe und Fenstergroesse	62 # Hintergrundfarbe und Fenstergroesse
36 ren.SetBackground(0.1, 0.2, 0.4)	63 ren.SetBackground(0.1, 0.2, 0.4)
:	:
50 ""	77 ""
51	78
52 # einen Simulationsschritt berechnen	79 # einen Simulationsschritt berechnen
53 # ...	80 t, y = calcStep().
54	81
55 # Positionen und Orientierungen der Koerper berechnen	82 # Positionen und Orientierungen der Koerper berechnen
56 # ...	83 r_laufkatze, T_laufkatze, r_last, T_last = calcPositions(y).
57	84
58 # Koerper updaten	85 # Koerper updaten
59 # ...	86 setPokeMatrix(laufkatze, r_laufkatze, T_laufkatze).
	87 setPokeMatrix(last, r_last, T_last)
60	88
61 # Bild neu rendern	89 # Bild neu rendern
62 renWin.Render()	90 renWin.Render()