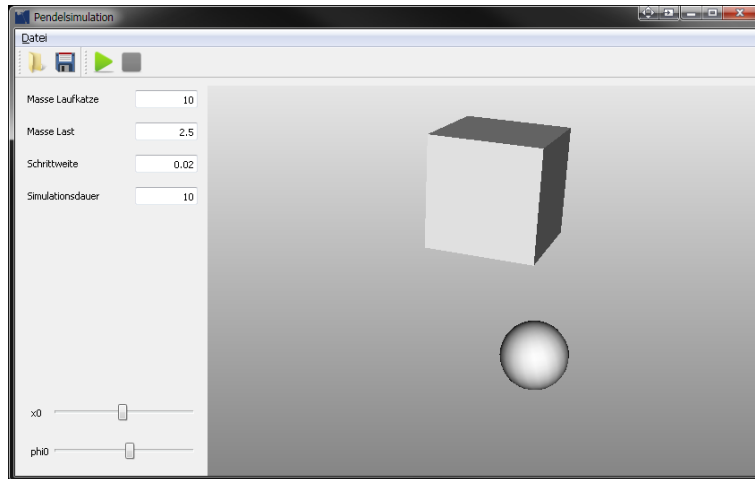


Übung 6b: GUI Programmierung mit PyQt4 - Teil II

Diese Übungsaufgabe dient als Zusammenfassung der bisher behandelten Themen und kombiniert die numerische Simulation mit einer 3D-Visualisierung in einer graphischen Benutzeroberfläche. Die Parameter des mechanischen Systems und dessen Startwerte können über Widgets vom Benutzer eingestellt werden. Die Werte können in eine Datei gespeichert und wieder geladen werden. Dafür wurden zwei Actions angelegt und als Toolbar und Menüeinträge verankert. Zwei weitere Actions dienen zum Starten und Beenden der Simulation. Die Visualisierung ist direkt an die Simulation gekoppelt und wird nach jedem Zeitschritt aktualisiert.



Für das Beispiel wurden bereits bekannte Elemente aus der Lehrveranstaltung aufgegriffen (numerische Simulation, Speichern von Daten als config-File, LineEdit Widgets, 3D-Visualisierung etc.) und zu einer Anwendung kombiniert (main.py). Durch Auslagerung von Funktionen / Klassen in Pakete und Module wurde eine hohe Wiederverwendbarkeit angestrebt. Der eigentliche Programmcode der main.py ist dadurch sehr kompakt (ca. 100 Zeilen ohne Kommentare). Die Funktionalitäten sind wie folgt in den bereitgestellten Modulen enthalten:

main.py	Hauptprogramm mit der Gui-Klasse
model.py	Differentialgleichungssystem rhs(...) – Zustandsableitung calcPositions(...) – Berechnung Orientierung und Lage der Körper aus q
widgets.py	Widgets (eigene Klassen), die das Hauptprogramm verwendet EingabeZahl - Label + TextEdit für Parametereingabe ParameterMaske - Block aus vier EingabeZahl-Widgets, realisiert Speichern und Laden des Parametersatzes VtkScene - Widget für die 3D-Darstellung und Animation StartwertSlider - Label + Slider für Startwertmanipulation
primitives.py	Klassen für Geometrieprimitive: Sphere, Box, File

Übungsaufgabe 1:

1. Schreiben Sie das Programm so um, dass nur eine Action für Play und Pause verwendet wird. Diese soll je nach Zustand ihr Icon ändern. Legen Sie dazu eine Variable an, die den aktuellen Zustand der Animation speichert und beim „Triggern“ der Action ausgewertet wird.
2. Ermöglichen Sie, dass die Animation auch über das Kontextmenü des VtkScene Widgets gesteuert werden kann.
3. Fügen Sie zwei Slider zum Setzen der Startwerte des Systems hinzu (x_0 , ϕ_0). Erzeugen Sie dazu zwei Instanzen der Klasse StartwertSlider (befindet sich in widgets.py) im Konstruktor der Klasse Gui in der Datei main.py. Fügen Sie die Slider in das vorhandene

Gridlayout unterhalb der Parameter-Textfelder ein (s. Abbildung). Verbinden sie das Signal `valueChanged(int)` der Slider-Objekte mit der Methode `setInitialValues` der Gui-Klasse und lesen sie die aktuellen Werte der Slider (`x0`, `phi0`) innerhalb dieser Funktion aus.

Hinweis: die `QSlider`-Instanz für das Verbinden des Signals ist als Attribut über `myStartwertSlider.slider` zu erreichen.