

Diff:

Differences between given skeleton and solution

In order to make the sample solution easier to understand, the differences between it and the given skeleton source code were highlighted with the help of the program `diff`.

Legend:

- Gray: unchanged text (only excerpts).
- Green: new lines
- Yellow: changed lines
- Red: deleted lines

Note: Files not listed have not been changed.

This document was created with the help of [diff2html](#) erstellt.

Gemeinsame Unterverzeichnisse: ../course12-gui-part2/exercise/code/.idea und ../course12-gui-part2/exercise/solution/.idea.

diff -u ../course12-gui-part2/exercise/code/main.py ../course12-gui-part2/exercise/solution/main.py

../course12-gui-part2/exercise/code/main.py	../course12-gui-part2/exercise/solution/main.py
:	:
37 self.iv_slider_x = cw.IVSlider("x0", limits=(-1000, 1000))	37 self.iv_slider_x = cw.IVSlider("x0", limits=(-1000, 1000))
38 self.iv_slider_phi = cw.IVSlider("phi0", limits=(-180, 180))	38 self.iv_slider_phi = cw.IVSlider("phi0", limits=(-180, 180))
39	39
	40 # slider for pendulum length (task 3)
	41 self.slider_l = cw.IVSlider("l", limits=(0, 300))
	42
40 # layout	43 # layout
41 self.hbox = QtWidgets.QGridLayout()	44 self.hbox = QtWidgets.QGridLayout()
42 self.hbox.addWidget(self.parameter_mask, 0, 0)	45 self.hbox.addWidget(self.parameter_mask, 0, 0)
:	:
44 self.hbox.addWidget(self.iv_slider_x, 1, 0)	47 self.hbox.addWidget(self.iv_slider_x, 1, 0)
45 self.hbox.addWidget(self.iv_slider_phi, 2, 0)	48 self.hbox.addWidget(self.iv_slider_phi, 2, 0)
46	49
47 self.hbox.addWidget(self.scene, 0, 1, 3, 1) # with rowspan=3, colspan=1	50 # task 3
	51 self.hbox.addWidget(self.slider_l, 3, 0)
	52
	53 self.hbox.addWidget(self.scene, 0, 1, 4, 1) # with rowspan=4, colspan=1
48 self.centralwg.setLayout(self.hbox)	54 self.centralwg.setLayout(self.hbox)
49	55
50 # actions for the File menu (Open, Save, Exit)	56 # actions for the File menu (Open, Save, Exit)
:	:
70)	76)
71	77
72 # save the status whether simulation is running (task 2)	78 # save the status whether simulation is running (task 2)
73 # XXX = False	79 self.is_playing = False
74	80
75 # create instance attributes for the coordinates of the mechanical	81 # create instance attributes for the coordinates of the mechanical
76 # system (x, phi)	82 # system (x, phi) and pendulum length l (task 3)
77 self.x = 0	83 self.x = 0
78 self.phi = 0.25 * np.pi	84 self.phi = 0.25 * np.pi
	85 self.l = 1
79	86
80 # two actions for controlling the simulation	87 # two actions for controlling the simulation
81 self.actn_start_anim = QtWidgets.QAction(self)	88 self.actn_start_anim = QtWidgets.QAction(self)
:	:
88 self.actn_stop_anim.setIcon(QtGui.QIcon("../data/stop.png"))	95 self.actn_stop_anim.setIcon(QtGui.QIcon("../data/stop.png"))
89 self.actn_stop_anim.triggered.connect(self.stop_animation)	96 self.actn_stop_anim.triggered.connect(self.stop_animation)
90	97
	98 # new action for toggling the simulation on and off
	99 self.actn_toggle_anim = QtWidgets.QAction(self)
	100 self.actn_toggle_anim.setText("Play")
	101 self.actn_toggle_anim.setIcon(QtGui.QIcon("../data/play.png"))
	102 self.actn_toggle_anim.triggered.connect(self.toggle_animation)
	103
91 # assemble the menus	104 # assemble the menus
92 self.menu_file = self.menuBar().addMenu("&File")	105 self.menu_file = self.menuBar().addMenu("&File")
93 self.menu_file.addAction(self.actn_open)	106 self.menu_file.addAction(self.actn_open)
:	:
106 self.toolbar_sim = QtWidgets.QToolBar("Simulation")	119 self.toolbar_sim = QtWidgets.QToolBar("Simulation")
107 self.toolbar_file.setIconSize(QtCore.QSize(24, 24))	120 self.toolbar_file.setIconSize(QtCore.QSize(24, 24))
108 self.addToolBar(self.toolbar_sim)	121 self.addToolBar(self.toolbar_sim)
109 self.toolbar_sim.addAction(self.actn_start_anim)	122 self.toolbar_sim.addAction(self.actn_toggle_anim)
110 self.toolbar_sim.addAction(self.actn_stop_anim)	123
	124 # task 2: do not add the actions for starting and stopping (now we use toggle)

		125	# self.toolbar_sim.addAction(self.actn_start_anim)
		126	# self.toolbar_sim.addAction(self.actn_stop_anim)
111		127	
112	# connect slider-change-signals with respective slots	128	# connect slider-change-signals with respective slots
113	self.iv_slider_x.slider.valueChanged.connect(self.setx)	129	self.iv_slider_x.slider.valueChanged.connect(self.setx)
:		:	
115	self.iv_slider_phi.slider.valueChanged.connect(self.setphi)	131	self.iv_slider_phi.slider.valueChanged.connect(self.setphi)
116	self.iv_slider_phi.slider.valueChanged.connect(self.draw_cart_pendulum)	132	self.iv_slider_phi.slider.valueChanged.connect(self.draw_cart_pendulum)
117		133	
		134	self.slider_l.slider.valueChanged.connect(self.change_pendulum_length)
		135	self.slider_l.slider.valueChanged.connect(self.draw_cart_pendulum)
		136	
		137	# The values of the instance attributes self.x, self.phi und self.l should
		138	# be displayed by the sliders from the beginning
		139	self.iv_slider_x.slider.setValue(int(self.x * 1000))
		140	self.iv_slider_phi.slider.setValue(int(self.phi * 180 / np.pi))
		141	self.slider_l.slider.setValue(int((self.l - 0.3) * 100))
		142	
118	self.draw_cart_pendulum()	143	self.draw_cart_pendulum()
119		144	
120	# end of self.__init__(...)	145	# end of self.__init__(...)
:		:	
139	yvalues_cart = np.array([-dy, dy, dy, -dy, -dy])	164	yvalues_cart = np.array([-dy, dy, dy, -dy, -dy])
140		165	
141	# position of the suspension (joint) and the load ("tip")	166	# position of the suspension (joint) and the load ("tip")
142	l = 1	167	l = self.l
143	x_joint = self.x	168	x_joint = self.x
144	y_joint = 0	169	y_joint = 0
145	x_tip = x_joint + l * np.sin(self.phi)	170	x_tip = x_joint + l * np.sin(self.phi)
:		:	
147	self.scene.plot(xvalues_cart, yvalues_cart)	172	self.scene.plot(xvalues_cart, yvalues_cart)
148	self.scene.plot([x_joint, x_tip], [y_joint, y_tip], symbol="o")	173	self.scene.plot([x_joint, x_tip], [y_joint, y_tip], symbol="o")
149		174	
		175	def toggle_animation(self):
		176	"""
		177	Switch between playback and stop
		178	"""
		179	
		180	if self.is_playing:
		181	self.stop_animation()
		182	self.is_playing = False
		183	
		184	else:
		185	self.start_animation()
		186	self.is_playing = True
		187	
150	def start_animation(self):	188	def start_animation(self):
151	"""	189	"""
152	Starting the simulation and animation. Thereby a timer is simply	190	Starting the simulation and animation. Thereby a timer is simply
:		:	
162	dt = int(float(self.parameter_mask.dt.getValue()) * 1000)	200	dt = int(float(self.parameter_mask.dt.getValue()) * 1000)
163	self.timer.start(dt)	201	self.timer.start(dt)
164		202	
		203	self.actn_toggle_anim.setText("Stop")
		204	self.actn_toggle_anim.setIcon(QtGui.QIcon("../data/stop.png"))
		205	
165	def stop_animation(self):	206	def stop_animation(self):
166	"""	207	"""

167	Stops the animation and resets the system to the start values.	208	Stops the animation and resets the system to the start values.
:		:	
172	# delete the timer attribute	213	# delete the timer attribute
173	del self.timer	214	del self.timer
174		215	
		216	self.actn_toggle_anim.setText("Play")
		217	self.actn_toggle_anim.setIcon(QtGui.QIcon("../data/play.png"))
		218	
175	def init_solver(self):	219	def init_solver(self):
176	"""	220	"""
177	Create solver and pass parameters (step size and masses)	221	Create solver and pass parameters (step size and masses)
:		:	
204	# update scene	248	# update scene
205	self.draw_cart_pendulum()	249	self.draw_cart_pendulum()
206		250	
		251	def change_pendulum_length(self, l):
		252	"""
		253	process slider values for pendulum length
		254	"""
		255	self.l = l / 100.0 + 0.3 # absolut value and scaling
		256	
		257	# overwrite the global variable `l` in the `cart_pendulum_model` module
		258	cart_pendulum_model.l = self.l
		259	
207	# end of class Gui	260	# end of class Gui
208		261	
209		262	

Gemeinsame Unterverzeichnisse: ../course12-gui-part2/exercise/code/__pycache__ und ../course12-gui-part2/exercise/solution/__pycache__.