

L1.8 Entwickeln einer Wasserwaage -App

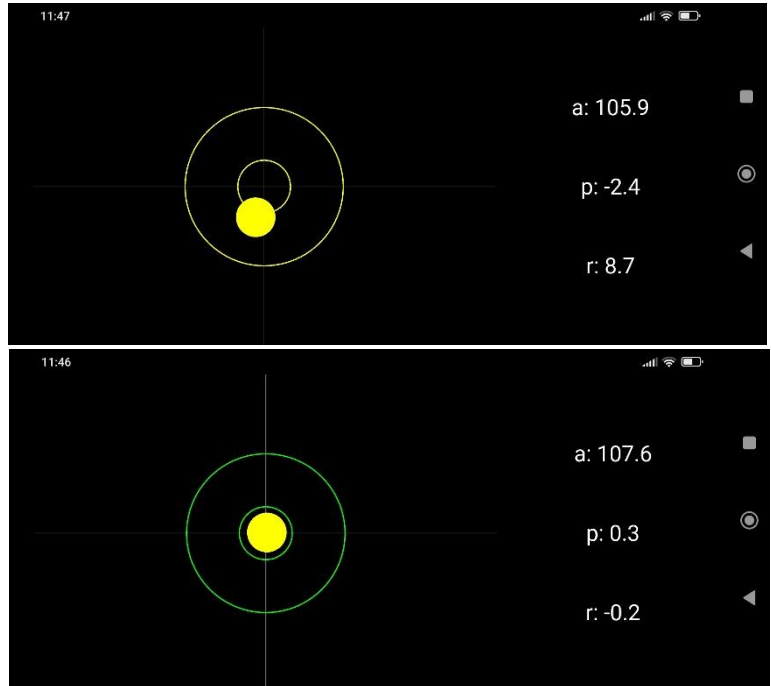
Was ist neu?

- Nutzen von Smartphone-Sensoren durch Einbinden des Package plyer. Hieraus nutzen wir das Modul SpatialOrientation für die Erkennung der Lage des Smartphones im Raum.
- ObjectProperty zum Austausch der Sensor-Werte zwischen Anwendungslogik und GUI (.kv-Datei).

1.8.1 Beginnen wir mit der Beschreibung der GUI in der .kv-Datei

Bitte beachten:

- Installieren Sie mit pip das für die Auswertung von Smartphone-Sensoren benötigte Package plyer.
- Imports werden in der .kv-Datei mit # eingeleitet. Das Objekt für den Zugriff auf den Lagesensor erhält unten stehend den Namen facade (Der komplexe Zugriff wird hinter der Fassade verborgen)
- Das selbst definierte Objekt für die Libelle der Wasserwaage enthält eine Zeichenfläche (canvas) auf die eine ausgefüllte Ellipse (Kreis) in gelb gezeichnet wird.
- Für alle Labels wird die Schriftgröße 70 vordefiniert.
- In der Wasserwaage wird die Fassade des Lagesensors als Object-Property angelegt. Diese enthält eine Zeichenfläche auf die das Fadenkreuz mit Linien und zwei Kreisen gezeichnet ist. Die Farbe der Kreise wird in der Anwendungslogik durch ein entsprechendes ListProperty festgelegt.
- Die Labels dienen der Ausgabe der aktuellen Werte für Roll-, Pitch- und Azimuth-Winkel. Diese werden vom Lagesensor des Smartphones ermittelt und die dazugehörigen NumericProperties immer wieder angepasst.



```

1 #:import facade plyer.spatialorientation
2
3 <Libelle>:
4     canvas:
5         Color:
6             rgb: [1,1,0]
7         Ellipse:
8             pos: self.pos
9             size: self.size
10
11 <Label>:
12     font_size: 70

```

J2	BPE 8: Entwicklung von mobilen Applikationen Wasserwaage-App	Informatik
----	--	------------

```

13
14 <Wasserwaage>:
15     facade: facade
16     canvas:
17         Line:
18             width: 1
19             points: 0, root.height/2, root.width*2/3, root.height/2
20         Line:
21             width: 1
22             points: root.width*1/3, 0, root.width*1/3, root.height
23         Color:
24             rgb: root.circle_color
25         Line:
26             width: 2
27             ellipse: root.width/3-self.height/12, self.height/2-
                self.height/12, self.height/6, self.height/6
28         Line:
29             width: 2
30             ellipse: root.width/3-self.height/4, self.height/2-
                self.height/4, self.height/2, self.height/2
31         Label:
32             center_x: root.width *5/6
33             center_y: root.height*3/4
34             text: f"a: {root.azimuth}"
35         Label:
36             center_x: root.width *5/6
37             center_y: root.height*2/4
38             text: f"p: {root.pitch}"
39         Label:
40             center_x: root.width *5/6
41             center_y: root.height*1/4
42             text: f"r: {root.roll}"
43         Libelle:
44             size: root.height/8, root.height/8
45             center: root.x_libelle,root.y_libelle

```

1.8.2 Anwendungslogik zur Wasserwaage (main.py)

- Hinterlegen Sie die Winkel für azimuth, pitch und roll in den NumericProperties im Gradmaß.
- Im Konstruktor der Wasserwaage wird die App im Querformat gestartet. Der Lagesensor wird über die Fassade des Sensors (vom Typ ObjectProperty) gestartet. Die Clock ruft die Methode getOrientation() 20 Mal in der Sekunde auf.
- In der Methode getOrientation() werden die Lagesensor-Werte erfasst und ins Gradmaß umgerechnet. Diese umgerechneten Werte sollen in entsprechenden NumericProperties hinterlegt werden. Aus pitch- und roll-Werten ergibt sich die Position der "Libelle" auf der APP-Oberfläche (x_libelle, y_libelle)
Ändern Sie die Farbe der gezeichneten Kreise über das ListProperty circle_color. Zeichnen Sie diese in gelb, blau oder grün je nach vom Sensor gelieferter Position.

```

1 import math
2 from kivy.app import App
3 from kivy.uix.widget import Widget
4 from kivy.properties import NumericProperty,
   ObjectProperty, ListProperty
5 from plyer import orientation
6 from kivy.clock import Clock
7
8 class Libelle(Widget):
9     pass
10
11 class Wasserwaage(Widget):
12     azimuth = NumericProperty(0)
13     pitch = NumericProperty(0)
14     roll = NumericProperty(0)
15     x_libelle = NumericProperty(0)
16     y_libelle = NumericProperty(0)
17     facade = ObjectProperty()
18     circle_color = ListProperty([1, 1, 0]) # Anfangswert gelb
19
20     def __init__(self, **kwargs):
21         super(Wasserwaage, self).__init__(**kwargs)
22         orientation.set_landscape()
23         self.facade.enable_listener()
24         Clock.schedule_interval(self.get_orientation, 1/20)
25
26     def get_orientation(self, dt):
27         if self.facade.orientation != (None, None, None):
28             a, p, r = self.facade.orientation
29             self.azimuth = round(math.degrees(a), 1) # azimuth in Grad
30                                                         # pitch in Grad
31                                                         # roll in Grad
32         # insert code here

```

J2	BPE 8: Entwicklung von mobilen Applikationen Wasserwaage-App	Informatik
----	--	------------

```
33
34 class WasserwaageApp(App):
35     def build(self):
36         self.title = 'Wasserwaage'
37         return Wasserwaage()
38
39 if __name__ == '__main__':
40     WasserwaageApp().run()
```

- Ergänzen Sie die Methode `getOrientation()` entsprechend oben stehender Beschreibung.
- Ergänzen Sie in der Datei `buildozer.spec` unter Requirements das Package `plyer`

```
1 # (List) Application requirements
2 # comma seperated e.g. requirements = sqlite3,kivy
3 requirements = kivy,plyer
```