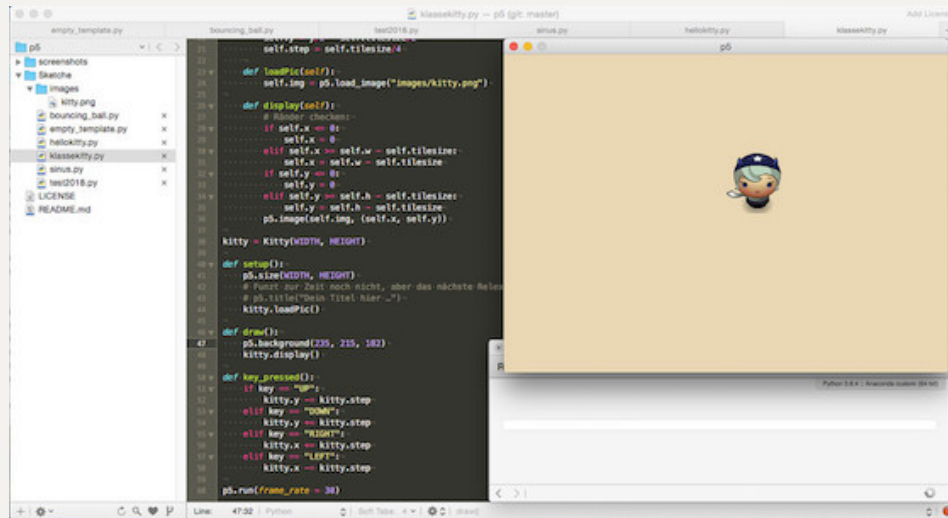


Tutorial: Spaß mit P5 (Python)

Das [gestern](#) in mein Wiki aufgenommene [P5 \(Python\)](#), der »echte« (C-) Python-Mode von [Processing](#), ist zwar noch beta und auch noch unvollständig implementiert, aber dennoch schon so mächtig, daß man damit viel anfangen kann. Daher mußte ich gestern abend noch ein wenig damit herumspielen und experimentieren:



Herausgekommen ist ein kleines Programm mit einer Spielfigur, die mit Hilfe der Pfeiltasten im Bildschirmfenster manövriert werden kann. Wem das bekannt vorkommt -- es ist eines der Programme, mit denen ich seinerzeit [PyGame](#) (und auch [Processing.py](#)) [erkundet habe](#). P5 scheint das Zeug dazu zu haben, unter anderem auch so etwas wie der legitime Nachfolger von PyGame zu werden.

Also, was habe ich angestellt? Im Gegensatz zur [Dokumentation](#), die

```
1 from p5 import *
```

empfiehlt, habe ich ein einfaches

```
1 import p5
```

Statement verwendet. Ich mag nämlich die mit den *Wildcards* verbundene Namensraumverschmutzung nicht und das Präfix `p5.` ist ja auch schnell getippt. Und die Spielfigur habe ich von vornherein in eine Klasse gekapselt, Der gesamte Sketch sieht daher so aus:

```
1 import p5
```

```

2  import os
3
4  # Hier wird der Pfad zum Verzeichnis des ».py«-
  Files gesetzt
5  # Erspart einem das Herumgehample in TextMate mit
  dem os.getcwd()
6  # und os.path.join()
7  file_path =
  os.path.dirname(os.path.abspath(__file__))
8  os.chdir(file_path)
9
10 WIDTH = 630
11 HEIGHT = 450
12
13 class Kitty():
14
15     def __init__(self, x, y):
16         self.w = x
17         self.h = y
18         self.tilesizesize = 90 # Bildgröße: 90x90
19
20     Pixel
21
22         self.x = x/2 - self.tilesizesize/2
23         self.y = y/2 - self.tilesizesize/2
24         self.step = self.tilesizesize/4
25
26     def loadPic(self):
27         self.img =
28         p5.load_image("images/kitty.png")
29
30     def display(self):
31         # Ränder checken:
32         if self.x <= 0:
33             self.x = 0
34         elif self.x >= self.w - self.tilesizesize:
35             self.x = self.w - self.tilesizesize
36         if self.y <= 0:
37             self.y = 0
38         elif self.y >= self.h - self.tilesizesize:
39             self.y = self.h - self.tilesizesize
40         p5.image(self.img, (self.x, self.y))
41
42 kitty = Kitty(WIDTH, HEIGHT)
43
44 def setup():
45     p5.size(WIDTH, HEIGHT)
46     # Funzt zur Zeit noch nicht, aber das nächste
47     Release

```

```

43     # soll den Titel wieder anzeigen.
44     # p5.title("Dein Titel hier ...")
45     kitty.loadPic()
46
47     def draw():
48         p5.background(235, 215, 182)
49         kitty.display()
50
51     def key_pressed():
52         if key == "UP":
53             kitty.y -= kitty.step
54         elif key == "DOWN":
55             kitty.y += kitty.step
56         elif key == "RIGHT":
57             kitty.x += kitty.step
58         elif key == "LEFT":
59             kitty.x -= kitty.step
60
61     p5.run(frame_rate = 30)

```

Einiges bedarf vielleicht der Erläuterung: `p5.title()` zeigt momentan nur den ersten Buchstaben des Titels an, daher habe ich dieses Kommando auskommentiert. Der Bug ist aber [bekannt](#) und soll mit dem nächsten Release behoben werden. Außerdem sind einige P5-Konstanten und -Variablen global (wie zum Beispiel `key`), sie dürfen daher **nicht** mit dem Präfix `p5.` versehen werden. Die [Reference](#) gibt darüber in der Regel Auskunft.

Und *last but not least* ist die Namenskonvention *pythonisch* -- sie folgt der [PEP 8](#), das heißt statt des in Processing (Java) üblichen *CamelCase* gibt es den Unterstrich, also `frame_rate` statt `frameRate`.

Ich nutzte ja den freien Editor [TextMate 2](#) und ich kann direkt aus diesem Editor heraus die Python-Sketches starten. Das ist fast genau so komfortabel wie die Processing-IDE (PDE), die ich daher kaum vermisse. Aber auch andere freie Editoren wie zum Beispiel [Atom](#) und [Visual Studio Code](#) erlauben dies, daher könnt Ihr den Editor nehmen, mit dem Ihr verachsen seid.



Das Bild der *Kitty* alias »*Horn Girl*« stammt wieder aus dem von *Daniel Cook* (*Danc*) in seinem Blog [Lost Garden](#) unter einer [freien Lizenz](#) (CC BY 3.0 US) zu Verfügung gestellten Tileset [Planet Cute](#). Ich habe es nur noch ein wenig auf die Größe von 90x90 Pixeln zurechtgeschnitten.

Fazit: P5 (Python) macht Spaß. Daher wird dies sicher nicht der letzte Beitrag dazu sein, den Ihr in diesem Blog Kritzelheft lesen werdet. Ich habe dafür auf GitHub schon einmal ein [Repositoryum](#) angelegt. *Still digging!*