 © 2016 Neumann Softwaretechnik Consulting

Pygame Basics

1 Vorbereitung

Für die Arbeit mit Pygame werden wird die Umgebung „PortablePython Version 2.7.x“ verwenden.

1.1 Download der PortablePython-Umgebung

Die Installationsdatei kann hier heruntergeladen werden:

http://portablepython.com/wiki/PortablePython2.7.6.1/

>>> http://ftp.belnet.be/PortablePython/v2.7/PortablePython_2.7.6.1.exe

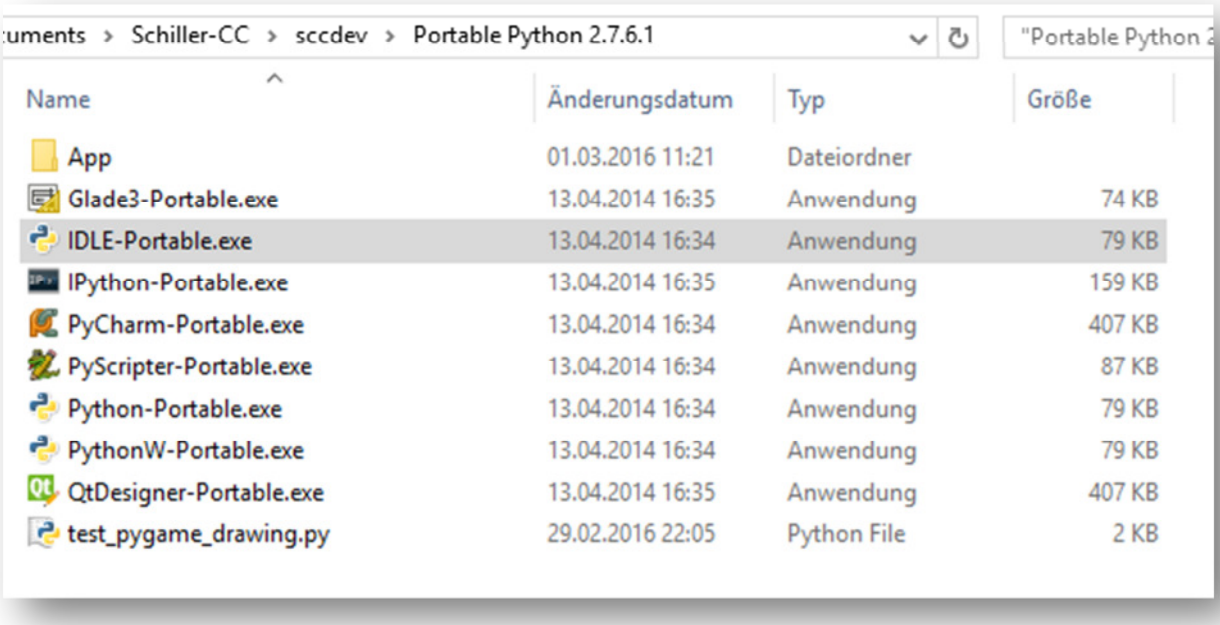
1.2 Installation der Umgebung:

Die Datei PortablePython_2.7.6.1.exe ausführen und den Inhalt in ein Verzeichnis auf der lokalen Platte installieren.

Hinweis: Nach der Installation kann das Verzeichnis auf einen USB-Stick kopiert oder verschoben werden. Die Python-Umgebung sollte dann aus dem neuen Verzeichnis heraus weiterhin startbar sein.

1.3 Ergebnis der Installation:

Im Installationsverzeichnis sollten die folgenden Dateien enthalten sein:

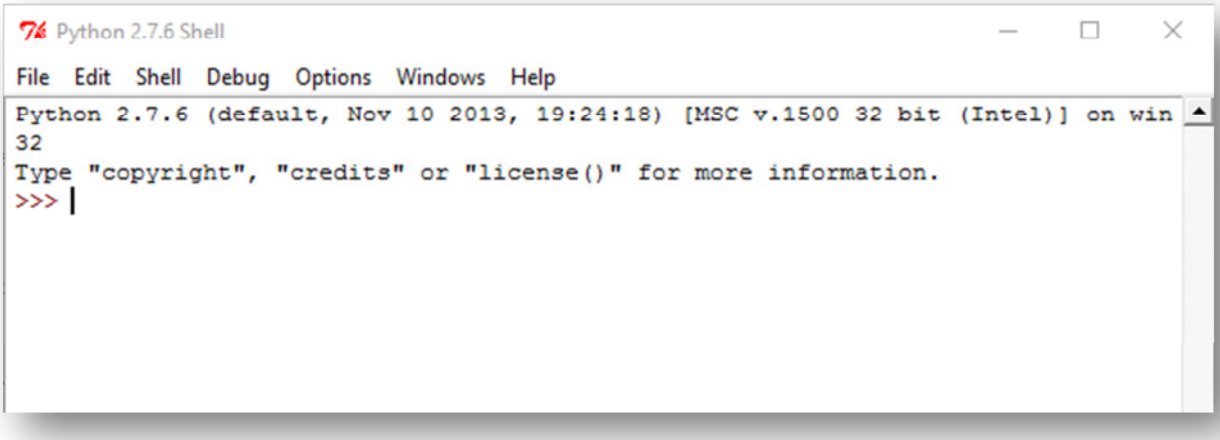


1.4 Test der Installation:

Starte das Programm IDLE-Portable.exe durch einen Doppelklick.

Daraufhin wird ein Fenster mit einer Python Shell in der Version 2.7.6 geöffnet.

Hinweis: Wenn das Programm von einem USB-Stick gestartet wird, kann das Öffnen des Fensters einige Sekunden dauern!

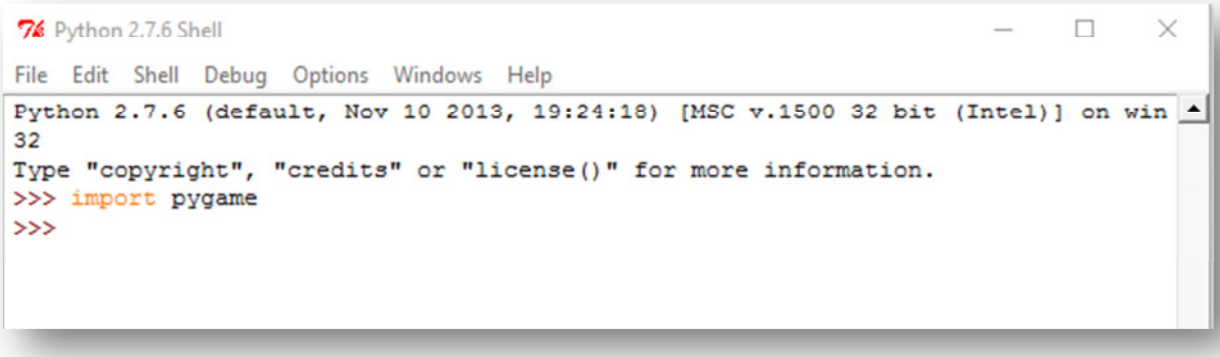


Um zu testen, ob die pygame-Umgebung funktioniert gibst Du folgenden Befehl in der Befehlszeile der Python-Shell ein:

```
import pygame
```

Wenn keine weiteren Meldungen kommen und nach kurzer Zeit einfach nur ein neuer Befehlsprompt („>>>“) erscheint, ist alles in Ordnung und wir können weiter machen.

So sollte es aussehen:



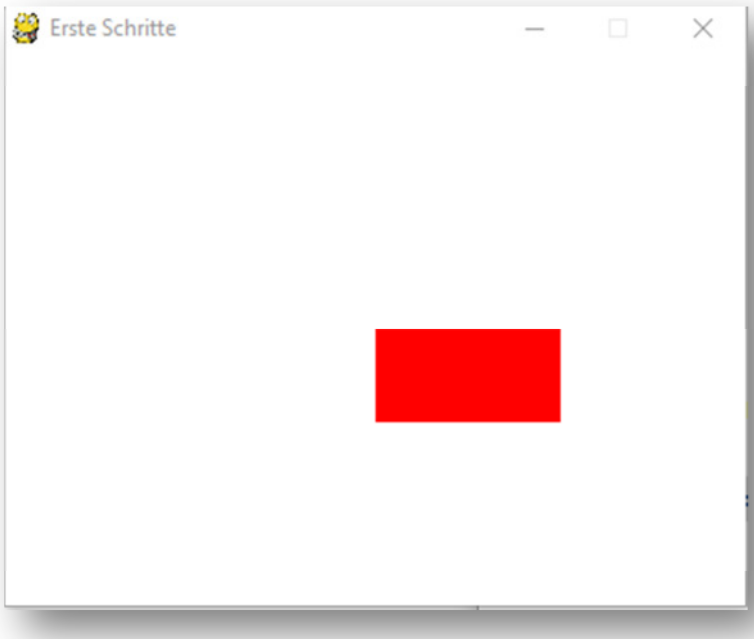
Falls andere Meldungen kommen, muss man sehen, was die Fehlerursache sein kann.

 © 2016 Neumann Softwaretechnik Consulting

2 Die ersten Schritte

Als erste Übung wollen wir über die Python-Shell-Umgebung ein Pygame-Fenster erstellen und darin ein rotes Rechteck darstellen.

Das ist unser Ziel:



Im Folgenden werden wir einige Befehle über die Python-Shell eingeben.

Dazu tippt Ihr die angegebenen Befehle jeweils hinter dem Prompt („>>>“) ein. Nach jeder Zeile drückst Du die ENTER-Taste. Wenn keine Fehler auftreten, sollte der Computer unter der Eingabe einen neuen Prompt anzeigen. Dann kannst Du den nächsten Befehl eingeben.

Zunächst müssen wir ein paar Bibliotheken importieren:

```
import pygame, sys
from pygame.locals import *
```

Nach der Eingabe dieser Befehle passiert noch nicht viel auf dem Bildschirm:

 © 2016 Neumann Softwaretechnik Consulting

```
Python 2.7.6 (default, Nov 10 2013, 19:24:18) [MSC v.1500 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> import pygame
>>> import pygame, sys
>>> from pygame.locals import *
>>>
```

Danach initialisieren wird die pygame-Umgebung:

```
pygame.init()
```

Jetzt gibt die Shell auch nur ein Datenarray zurück, das uns momentan nicht weiter interessiert:

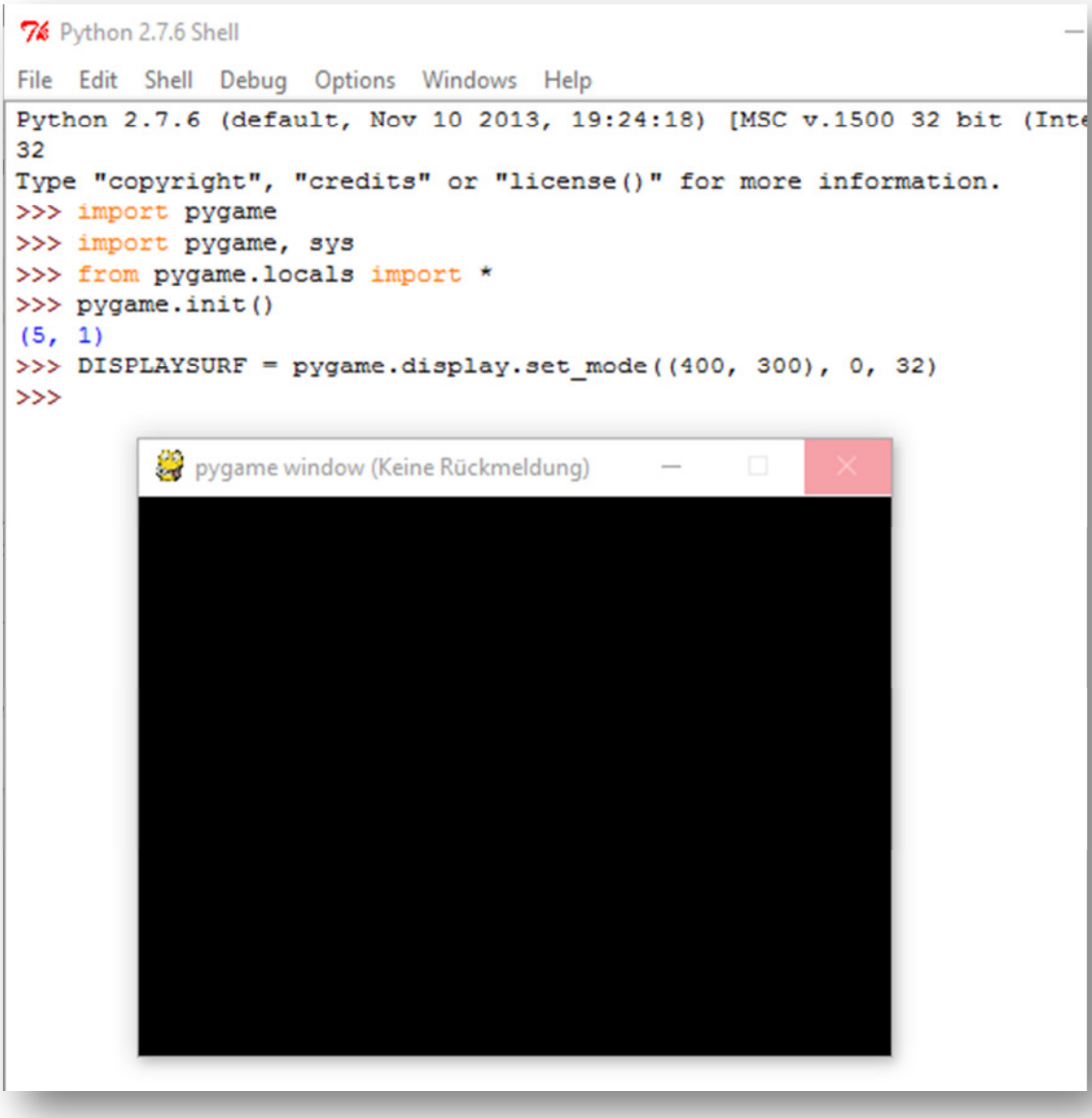
```
Python 2.7.6 (default, Nov 10 2013,
Type "copyright", "credits" or "lice
>>> import pygame
>>> import pygame, sys
>>> from pygame.locals import *
>>> pygame.init()
(5, 1)
>>>
```

Im nächsten Schritt erstellen wir eine sogenannte „Surface“ (deutsch: Oberfläche). Das ist sozusagen die Leinwand auf der sich alle weiteren Zeichenschritte abspielen werden.
Die Parameter der dem Befehl geben die Größe des Windows („(400,300)“) und die Position („0,32“) relativ zum Shell-Fenster an:

```
DISPLAYSURF = pygame.display.set_mode((400, 300), 0, 32)
```

Jetzt wird das Leinwandfenster geöffnet:

 © 2016 Neumann Softwaretechnik Consulting

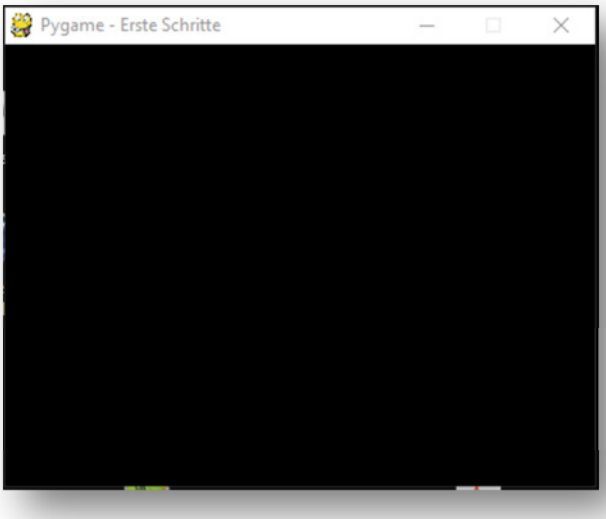


In der Kopfzeile des Windows soll der Text „Pygame – Erste Schritte“ angezeigt werden:

```
pygame.display.set_caption('Pygame – Erste Schritte')
```

Die neue Überschrift:

 © 2016 Neumann Softwaretechnik Consulting




```
Python 2.7.6 (default, Nov 10 2013, 19:24:18) [MSC v.1500 32 bi
Type "copyright", "credits" or "license()" for more information
>>> import pygame
>>> import pygame, sys
>>> from pygame.locals import *
>>> pygame.init()
(5, 1)
>>> DISPLAYSURF = pygame.display.set_mode((400, 300), 0, 32)
>>> pygame.display.set_caption('Pygame - Erste Schritte')
>>>
```

Jetzt bringen wir ein wenig Farbe ins Spiel. Damit wir aber nicht immer die drei RBG-Werte eingeben müssen, definieren wir uns einfach zwei Variablen für „WHITE“ und „RED“:

```
WHITE = (255, 255, 255)

RED   = (255,   0,   0)
```

```
Python 2.7.6 (default, Nov 10 2013, 19:24:18) [MSC v.1500 32 b
Type "copyright", "credits" or "license()" for more informatio
>>> import pygame
>>> import pygame, sys
>>> from pygame.locals import *
>>> pygame.init()
(5, 1)
>>> DISPLAYSURF = pygame.display.set_mode((400, 300), 0, 32)
>>> pygame.display.set_caption('Pygame - Erste Schritte')
>>> WHITE = (255, 255, 255)
>>> RED   = (255,   0,   0)
>>>
```

Genug mit den Vorbereitungen!

Der Hintergrund der Leinwand wird weiß gefüllt („fill“) und danach wird ein rotes Rechteck auf die Fläche gezeichnet (“draw”):

```
DISPLAYSURF.fill(WHITE)

pygame.draw.rect(DISPLAYSURF, RED, (200, 150, 100, 50))
```

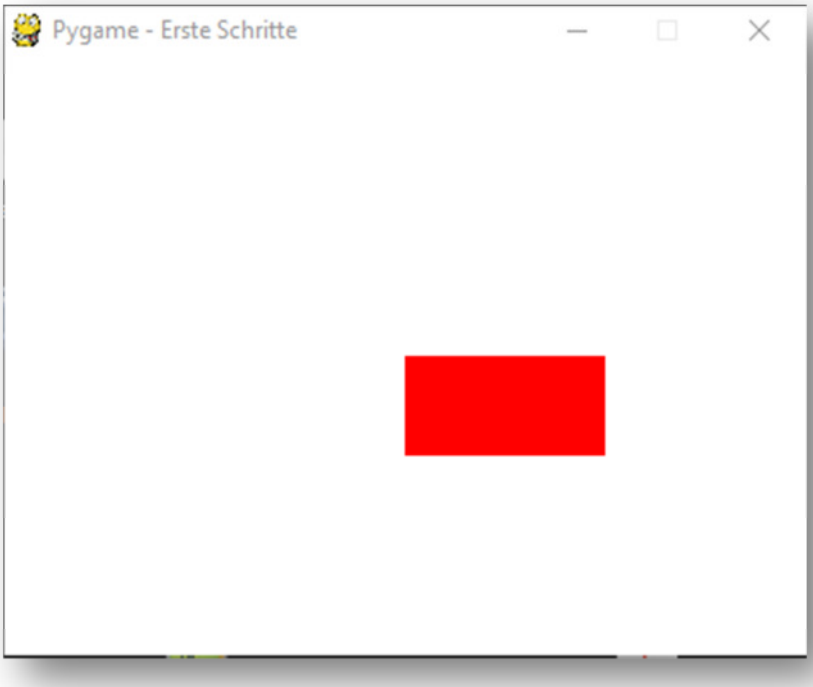
Aber … wie man sieht, sieht man noch nichts:

```
Python 2.7.6 (default, Nov 10 2013, 19:24:18) [MSC v.1500 32 bi
Type "copyright", "credits" or "license()" for more information
>>> import pygame
>>> import pygame, sys
>>> from pygame.locals import *
>>> pygame.init()
(5, 1)
>>> DISPLAYSURF = pygame.display.set_mode((400, 300), 0, 32)
>>> pygame.display.set_caption('Pygame - Erste Schritte')
>>> WHITE = (255, 255, 255)
>>> RED   = (255,   0,   0)
>>> DISPLAYSURF.fill(WHITE)
<rect(0, 0, 400, 300)>
>>> pygame.draw.rect(DISPLAYSURF, RED, (200, 150, 100, 50))
<rect(200, 150, 100, 50)>
>>>
```

… jetzt kommt der Befehl mit dem „ABRAKADABRA“:

```
pygame.display.update()
```

We proudly present:



Das Fenster kann man mit folgendem Befehl wieder schließen:

```
pygame.quit()
```

3 Das komplette Programm als Datei erste_Schritte.py

Um die Befehle nicht immer einzeln eingeben zu müssen, kannst Du die Befehle auch in eine Textdatei mit dem Namen „erste_Schritte.py“ schreiben:

```
import pygame, sys
from pygame.locals import *

pygame.init()

# set up the window
DISPLAYSURF = pygame.display.set_mode((400, 300), 0, 32)
pygame.display.set_caption('Erste Schritte')

# set up the colors
WHITE = (255, 255, 255)
RED   = (255,   0,   0)

# draw on the surface object
DISPLAYSURF.fill(WHITE)
pygame.draw.rect(DISPLAYSURF, RED, (200, 150, 100, 50))

# run the game loop
while True:
```

```
for event in pygame.event.get():
    if event.type == QUIT:
        pygame.quit()
        sys.exit()
pygame.display.update()
```

Damit man das Fenster auch mit der Maus wieder schließen kann, ist der Befehl „sys.exit()“ noch dazu gekommen. Das Ganze ist in eine Endlosschleife („while True:“) eingebettet, damit die Animation so lange läuft, bis man das Window schließt. Ist hier noch nicht so spannend, weil wir noch keine bewegten Elemente haben … aber das kann ja schnell kommen ;-)

4 Aufgaben

1) Wie wärs, wenn wir dem Rechteck mal die Farbe BLUE geben würden?
   (RGB-Code für Blau ist: (0, 0, 255))

2) Mit dem Befehl
   `pygame.draw.circle(DISPLAYSURF, GREEN, (300, 50), 20, 0)`
   kann man einen Kreis zeichnen.
   Wie wäre es mit einem grünen Kreis?

3) Jetzt noch mit Bewegung …
   … Was müsste man machen, um den Kreis von links nach rechts „wandern“ zu lassen?