

Random Walks

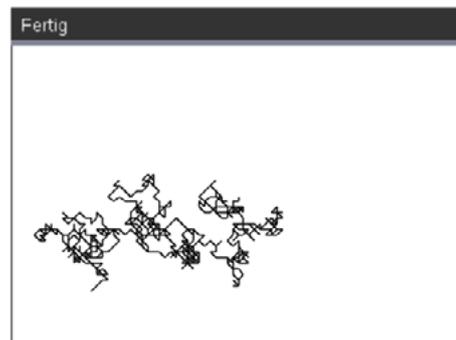
rwal.py, rwalk1.py

Wir beobachten den Weg eines Betrunkenen. Er ist total besoffen und steht in der Mitte eines Platzes. Er will weiter gehen und stolpert jeden Schritt in eine zufällige Richtung. Der Weg, den er zurücklegt, nennt man einen *random walk* (deutsch: Irrfahrt, Zufallsweg). Mit dem folgenden Programm soll ein derartiger Weg am Schirm des TI-Nspire™ CX II-T CAS simuliert werden.

```

1.4 1.5 1.6 *PyKurz RAD
rwalk.py 1/13
import tiplotlib as plt
from random import *
from ti_system import *
plt.cls()
x,y=0,0; h=.25;k=0
while k!="esc":
    k=get_key()
    x1,y1=x,y
    x+=h*randint(-1,1)
    y+=h*randint(-1,1)
    plt.line(x1,y1,x,y,"")

```



Mit der `[esc]`-Taste kannst du den Weg nach Hause (oder zum nächsten Gasthaus) abbrechen. Beschreibe, in welche Richtungen sich der Trunkenbold in dem Programm bewegen kann.

Was verändert sich in seinen Torkelbewegungen, wenn der Code so verändert wird:

```

1.5 1.6 1.7 *PyKurz RAD
rwalk1.py 11/13
import tiplotlib as plt
from random import *
from ti_system import *
plt.cls()
x,y=0,0; h=.25;k=0
while k!="esc":
    k=get_key()
    x1,y1=x,y; r=random()
    if r<0.5: x+=h*choice([-1,1])
    else: y+=h*choice([-1,1])
    plt.line(x1,y1,x,y,"")

```

Im zweidimensionalen Fall gilt, dass ein Irrweg schlussendlich immer wieder zum Ausgangspunkt zurückkehrt.

Ein Betrunkener, der sein Haus verlässt, kehrt früher oder später wieder in sein Haus zurück.

Hingegen ist es in der dritten Dimension nicht sicher, dass ein Irrweg wieder zum Ausgangspunkt zurückführt. Dazu gibt es auch ein passendes Sprichwort:

Ein Mensch kann betrunken werden, ein Vogel nicht.

Hinweis: siehe auch die Beispiele 2.2.5 und 2.2.6 zu den Zufallszahlen.