

Unit 6 : micro:bit met Python

Toepassing: Dobbelen

In deze toepassing schrijf je een programma waarmee je data verzamelt met behulp van de micro:bit. Daarna voer je het programma uit in een gedeeld scherm, zodat je de grafiek ziet “groeien” tijdens het verzamelen van de data.

Doelen :

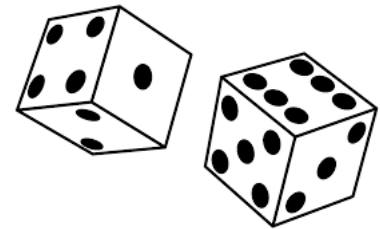
- Het schrijven van een programma;
- Het maken van een dynamische grafiek met de ‘Data & Statistics’-pagina.

1. In deze toepassing gebruik je de technieken uit de drie lessen van deze unit:

Data verzamelen met de micro:bit waarbij je ‘gestures’ of knoppen gebruikt, en het gebruik van lijsten om de data op te slaan en uit te wisselen met TI-Nspire.

Daarna maak je een ‘TI-Nspire’-pagina waarin:

- aan de ene kant het Python-programma word uitgevoerd;
- aan de andere kant de grafiek (puntenplot of histogram) wordt getoond terwijl het programma loopt.



2. Begin je programma met het importeren van de ‘micro:bit’-module en ook de random-module.

Definieer daarna een lege lijst (“ogen”) en sla deze direct op als ‘TI-Nspire’-lijst zodat ook deze lijst leeg is als het programma wordt gestart.

Laat het programma dan de instructie voor de gebruiker tonen. In dit geval gebruiken we de optie om de micro:bit te schudden voor het werpen met de dobbelstenen.

```

1.1 | *Dobbelen2 | RAD | 7/7
*dobbelen.py
from microbit import *
from random import *
ogen = []
store_list("ogen",ogen)
print("Schud de micro:bit om te dobbelen")
    
```

3. Het programma moet elke keer als er “geworpen” wordt het volgende doen:

- Kies twee willekeurige getallen (van 1 tot 6).
- Tel deze op.
- Voeg de uitkomst toe aan de lijst “ogen”.
- Toon het aantal worpen, de twee getallen en hun som op het scherm.
- Laat de twee getallen ook op het display zien.
- Pas de ‘TI-Nspire’-lijst aan.

```

1.1 | *Dobbelen2 | RAD | 11/11
*dobbelen.py
from microbit import *
from random import *
ogen = []
store_list("ogen",ogen)
print("Schud de micro:bit om te dobbelen")

while get_key() != "esc":
    **
    **
    **
    
```

Tip: Het aantal worpen is gelijk aan de lengte van de lijst “ogen”:
len(ogen)

4. Begin met het werpen van de dobbelsteen.

```

1.1 *Dobbelen2 RAD 11/11
*dobbelen.py
from microbit import *
from random import *
ogen = []
store_list("ogen",ogen)
print("Schud de micro:bit om te dobbelen")

while get_key() != "esc":
    if accelerometer.was_gesture("shake"):
        d1 = randint(1,6)
        d2 = randint(1,6)
    
```

5. Tel de uitkomsten op en voeg deze som toe aan de lijst "ogen".

```

1.1 *Dobbelen2 RAD 13/14
*dobbelen.py
ogen = []
store_list("ogen",ogen)
print("Schud de micro:bit om te dobbelen")

while get_key() != "esc":
    if accelerometer.was_gesture("shake"):
        d1 = randint(1,6)
        d2 = randint(1,6)
        som = d1 + d2
        ogen.append(som)
    
```

6. Laat de twee uitkomsten zien op het 'micro:bit'-display.

display.clear() wordt gebruikt om de vorige uitkomst van het display te wissen, zodat het duidelijk is dat er weer een nieuwe waarde wordt getoond.

```

1.1 1.2 *Dobbelen2 RAD 18/20
dobbelen.py
d1 = randint(1,6)
d2 = randint(1,6)
som = d1 + d2
ogen.append(som)
display.clear()
display.show(d1)
sleep(250)
ogen.append(som)
display.clear()
display.show(d2)
sleep(250)
    
```

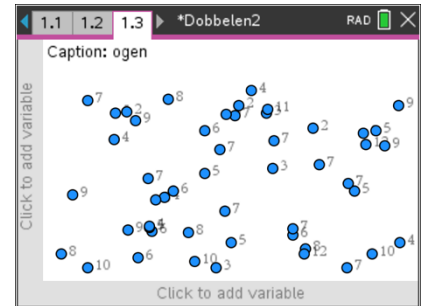
7. Voeg regels toe om de informatie op het 'TI-Nspire'-scherm te tonen en om de 'TI-Nspire'-lijst te updaten.

```

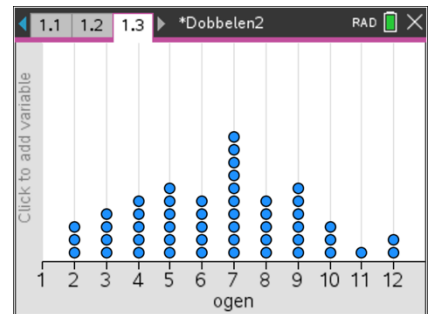
1.1 1.2 *Dobbelen2 RAD 15/22
dobbelen.py
ogen.append(som)
display.clear()
display.show(d1)
sleep(250)
display.clear()
display.show(d2)
sleep(250)
w = len(ogen)
print("worp",w," : ",d1,"+",d2,"=",som)
store_list("ogen",ogen)
    
```

8. Als het Python-programma goed werkt, ga je verder met de link naar de 'TI-Nspire'-plotmogelijkheden. Voer eerst het programma uit en werp ongeveer 50 keer met de dobbelstenen. Druk op **[esc]** om het programma te stoppen.

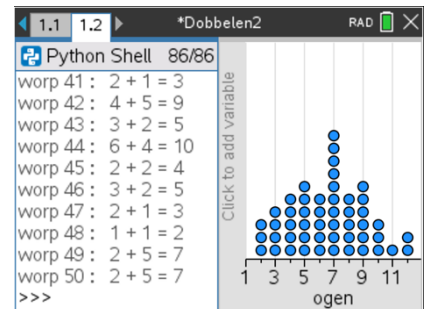
Zorg dat je in de Python-shell zit (de pagina met de >>> - cursor) en druk dan op **[ctrl]-[doc]** of **[ctrl]-[i]** om een pagina toe te voegen en kies voor **Data & Statistics**. Dan zie je een scherm zoals hiernaast.



9. Klik op "Click to add variable" onderin het scherm en selecteer "ogen". Je ziet dan een puntenplot van de data verschijnen.



10. Ga een pagina terug naar de Python-shell met **[ctrl]-[pijl naar links]** en druk op **[ctrl]-[4]** om de twee pagina's samen te voegen tot één gedeelde pagina. Links de Python-shell en rechts de 'Data & Statistics'-toepassing.



11. De Shell is 're-initialized'. Dit betekent dat de data zijn gewist. Ga weer naar de Python-editor en start het programma met **[ctrl]-[R]**. Je komt dan terug in het gedeelde scherm en als je de micro:bit heen en weer schudt dan zie je de grafiek ontstaan.

Je kunt ook het programma starten vanuit de Python-shell (het linkerdeel van het scherm) om opnieuw te starten.