Micro:bit

Python: Tilfeldig

Skrevet av: Oversatt fra microbit-micropython.readthedocs.io (https://microbit-micropython.readthedocs.io/en/latest/tutorials/random.html)

Oversatt av: Øistein Søvik og Susanne Rynning Seip

Kurs: Microbit
Tema: Elektronikk, Tekstbasert
Fag: Programmering, Matematikk
Klassetrinn: 5.-7. klasse, 8.-10. klasse, Videregående skole

Denne oppgaven er en del av oppgavesamlingen *Programmering i micro-python* og bygger videre på Python: Knapper (../python_buttons/python_buttons_nb.html).

Vi anbefaler at du laster ned og skriver koden din i mu editor (https://codewith.mu/) når du jobber med disse oppgavene. Instruksjoner for hvordan man laster ned Mu finner du på nettsiden via linken.

Når Mu er installert kan du koble micro:biten din til datamaskinen via en USB-kabel. Skriv koden din i editor-vinduet og trykk på “Flash”-knappen for å laste koden over på micro:biten. Hvis det ikke fungerer, sørg for at micro:biten har dukket opp som en USB-enhet på datamaskinen din.

Introduksjon

Er du lei av at enheten din gjør det samme hver gang; kanskje du ønsker å gjøre utfallet mer spennende? I denne oppgaven skal vi lære hvordan vi kan få enheten til å oppføre seg tilsynelatende tilfeldig.

MicroPython har en `random` modul som gjør det enkelt å inkludere tilfeldigheter og litt kaos inn i koden din. For eksempel, her er hvordan en kan scrolle et tilfeldig navn over displayet:

```
from microbit import *
import random

navn = ["Mari", "Yolanda", "Jakob", "Sofie", "Kushal", "Mei Xiu", "William" ]

display.scroll(random.choice(names))
```

Listen (`navn`) inneholder syv navn som er definert som tekststrenger. Den siste linjen består av nøstede (Tenk på lagene på en "løk" som vi nevnte tidligere) funksjoner: metoden `random.choice` tar inn listen `navn` som et argument og returnerer et tilfeldig element. Dette elementer er argumentet til `display.scroll` .

Prøv det ut selv

- [ ] Endre på listen og inkluder dine egne navn.

- [ ] Hva kan bruksområdet til en slik liste være? Tenk ut to tilfeller hva du kan bruke en slik liste til.

- [ ] Inkluder samme navnet flere ganger i listen. Hva skjer?

Tilfeldige tall

Tilfeldige tall er veldig nyttige og er vanlige i spill. Vet du om andre plasser vi bruker terninger?

MicroPython kommer utrustet med en rekke nyttige metoder for å lage tilfeldige tall. Her er hvordan en kan lage en enkel terning:

```
from microbit import *
import random

display.show(str(random.randint(1, 6)))
```

Hver gang enheten er startet på nytt så viser displayet et tall mellom 1 og 6. Du begynner å bli kjent med grepet *nøsting*, så det er viktig å huske på at `random.randint` returnerer et heltall mellom to argumenter, inkludert endepunktene (argumentet heter `ranint` som er en forkortelse for *random integer*, siden *integer* betyr heltall). Merk at siden `display.show` forventer tekst så må vi bruke `str` funksjonen til å gjøre om tallverdien til en tekst (vi gjør om for eksempel `6` til `"6"`).

Dersom du vet at du alltid ønsker et tall mellom `0` og `N` så kan vi bruke `random.randrange` metoden. Den tar inn ett tall og returner et tilfeldig tall opp til, men ikke inkludert verdien til argumentet `N` . (dette er forskjellig fra `random.randint`)

Noen ganger trenger du desimaltall. Programmerere kaller gjerne disse for *flyttall* og det er mulig å generere slike tall ved å bruke `random.random` metoden. Dette returnerer en verdi mellom `0.0` og `1.0` inklusive endepunktene. Dersom du trenger større tilfeldige desimaltall kan du legge sammen resultatene fra `random.randrange` og `random.random` slik som dette

```
from microbit import *
import random

answer = random.randrange(100) + random.random()
display.scroll(str(answer))
```

Prøv det ut selv

- [ ] Skriv ned det største tallet programmet ovenfor kan lage.

- [ ] Skriv ned det minste tallet programmet ovenfor kan lage.

- [ ] Skriv om koden for terningen ovenfor slik at den bruker `randrange` i stedet for `randint`. Merk at du må gjøre noen endringer slik at den gir ut verdier fra 1 til 6 og ikke 0 til 5.

Hint

Kaostilstander

Når datamaskinen din lager tilfeldige tall er de ikke helt tilfeldig. De gir bare ut tilsynelatende tilfeldige resultater gitt en *starttilstand* (Kalles gjerne for *seed* på engelsk). Tilstanden er ofte laget fra omtrentlig tilfeldige tall, slik som temperaturen i datamaskinen din, musebevegelsene de siste minuttene eller klokkeslettet. Allle disse metodene kan kombineres.

Noen ganger ønsker du å ha tilfeldig oppførsel som gjentar seg, en kilde av tilfeldighet som er reproduserbar. Det er som å si at du trenger de samme fem tilfeldige verdiene hver gang du kaster en terning.

Dette er heldigvis enkelt siden vi kan sette starttilstanden direkte i MicroPython. Gitt en fast starttilstand så vil den generere de samme tilfeldige tallene hver gang. Starttilstanden er satt med `random.seed` og ett positivt heltall. Denne versjonen av terning programmet produserer alltid de samme resultatene

```
from microbit import *
import random

random.seed(1337)
while True:
    if button_a.was_pressed():
        display.show(str(random.randint(1, 6)))
```

Litt juks

- [ ] Skriv om koden ovenfor slik at den alltid gir ut seks (6) dersom en holder inne knappen `A` *og* `B`.

- [ ] Skriv om koden ovenfor slik at hver gang du gjør en *bevegelse* viser den seks (6) ellers skal den virke som før.

Python kan kjenne igjen følgende bevegelser `up`, `down`, `left`, `right`, `face up`, `face down`, `freefall`, `3g`, `6g`, `8g`, `shake` ved å bruke funksjonen `accelerometer.is_gesture("bevegelse")`.

Hint

Neste oppgave i samlingen er Python: Rotasjon og fall (../python_gestures/python_gestures_nb.html). Klikk videre for å fortsette gjennom samlingen.