Hide and Seek - Micropython

Dette spillet er en kombinasjon av *Hide and Seek* og *Hot and Cold*. Det finnes en spiller som leter etter alle andre microbits og alle deltakerne må gjemme microbiten sin. Microbiten til søkeren kommer til å ha sterkere lys på skjermen jo mer den nærmer seg en av deltakerne. Når en deltaker har blitt funnet kan søkeren trykke på a-knappen på microbiten som de fant. Hvis alle deltakerne blir funnet vinner søkeren og spillet begynner på nytt.

Deltakere

Registrering

Før spillet begynner må deltakerne registrere seg hos søkeren. Sånn at signalet er forskjellig for hver deltaker, sender deltakerne unike IDer til søkeren. Fordi disse IDene er individuell for hver microbit må man ikke bruke en forskjellig kode for hver deltaker.

```
9  idHider = str(hash(str(machine.unique_id()))) + str(random.randint(1000, 9999))
```

```
32          if not ready:
33              display.show("J")
34              if button_a.was_pressed():
35                  radio.send("S,"+idHider)
36                  ready = True
37          else:
38              display.show("R")
39              if received == "START":
40                  gameStart = True
```

Deltakerne må også sende en status. Dette er viktig hos søkeren sånn at den vet hvilken melding den skal bruke hvor, men dette blir forklart senere i koden. De sender altså “S,” og IDen sin. Når disse er sendt er microbitene klare for spillet hvor de venter på et startsignal fra søkeren.

Når spillet løper

Hvis spilleren ikke er funnet sender den “H,” og IDen sin. Statusen “H” står for *hot*, og dette blir brukt hos søkeren for å differensiere disse signalene fra de som blir sendt senere.

```
17          if not found:
18              display.show("H")
19              radio.send("H,"+idHider)
20              sleep(500)
```

Når søkeren finner deltakeren trykker de på a-knappen av deltakeren. Dette sender “S,” og IDen til søkeren som hos registreringen. Som sagt er dette for å få søkeren til å vite forskjellen mellom *hot* signalene og *found* signalene, i mens den fortsatt får beskjed fra hvilken deltaker signalet kommer fra.

```
21          if button_a.was_pressed():
22              sleep(1000)
23              radio.send("S,"+idHider)
24              found = True
25              display.scroll("FOUND")
```

Søkeren

Registrering

Som sagt finnes det en slags registreringsperiode før selve spillet har begynt. I denne perioden kan deltakerne sende et signal til søkeren for å bli del av spillet. Disse meldingene blir lagret i en liste eller en dictionary som heter *‘hidersFound’* hos søkeren. Den legger de bare til hvis status-meldingen er “S”, sånn at de ikke blir registrert når spillet løper og når statusmeldingen er noe annet.

```
31          if received and statusMsg == "S":
32              hidersFound[idMsg] = False
```

Hvis man trykker a-knappen, og minst en deltaker er registrert, er registreringsperioden over og spillet begynner.

```
33          if button_a.was_pressed() and len(hidersFound) >= 1:
34              gameStart = True
35              radio.send("START")
```

Hvordan finner søkeren deltakerne?

Som sagt sender deltakerne disse “hot” signalene til søkeren. Signalstyrken av disse meldingene blir regnet om til en lysstyrke mellom 0 og 9. Dvs. at jo nermere søkeren er til deltakeren, desto lysere blir skjermen.

Først blir microbiten fortalt å vise bildet *‘imageBright’.* Dette er bildet som viser lysstyrken som tilsvarer hvor langt vekk deltakeren er.

```
37          display.show(imageBright)
```

På begynnelsen er bildet tomt, dvs. lysstyrken er 0.

```
8  imageBright = Image("00000:00000:00000:00000:00000")
```

Etter det blir det sjekket om søkeren har fått et signal fra andre microbits. Hvis den har det, spør den etter statusmeldingen. Hvis den er “H” tar den signalstyrken av ID-meldingen (rssi) og legger den til listen *‘hidersStrength’,* som holder styr på signalstyrkene til alle de forskjellige IDene (deltakerne). Dette blir brukt for å bare bruke det sterkeste signalet fra deltakerne, dvs. at bare den deltakeren som er nærmest søkeren kommer til å ha en effekt på lysstyrken.

```
40              hidersStrength[idMsg] = rssi
41              rssiMax = max(hidersStrength.values())
```

Nå som søkeren har valgt den meldingen som har høyest rssi regner den ut hvilken lysstyrke denne signalstyrken tilsvarer. Hvis signalstyrken er større enn -44 blir lysstyrken *(‘brightness’)* automatisk 9, som er det lyseste. Hvis den er mindre enn det minste tallet som man kan velge selv er lysstyrken 0.

```
42              brightness = 9-(((-1)*rssiMax)-44)%brightnessSteps
43              if (rssiMax > -44):
44                  brightness = 9
45              if (rssiMax < (-44 - brightnessSteps*9)):
46                  brightness = 0
```

Denne lysstyrken blir lagt til en liste som heter *‘rssiHistory’*, og den inneholder max fem lysstyrker om gangen. Hvis listen er lengre enn fem blir den første verdien i listen fjernet.

```
47              rssiHistory.append(brightness)
48              if (len(rssiHistory) > historyLength):
49                  rssiHistory.pop(0)
```

Etter det blir gjennomsnittet av de fem verdiene i *‘rssiHistory’* regnet ut og *‘imageBright’* blir fylt med denne verdien.

```
50              rssiAverage = sum(rssiHistory) / len(rssiHistory)
51              imageBright.fill(int(rssiAverage))
```

Så blir denne prosessen gjentatt neste gang den får et signal med statusen “H” som gjør at lysstyrken endrer seg hele tiden.

Det blir også sjekket for signaler uten statusen “H”. Hvis søkeren får et sånt telles deltakeren som sendte signalet som funnet.

```
52              if idMsg in hidersFound and not statusMsg == "H":
53                  hidersFound[idMsg] = True
```

Hvis alle spillerne er funnet har søkeren vunnet. *‘HidersNotFound’* er en liste av deltakerne som ikke har blitt funnet enda, og hvis denne er tom blir alle listene tømt og registreringsperioden blir satt i gang igjen.

```
54          hidersNotFound = dict(filter(lambda elem:elem[1] == False, hidersFound.items()))
55          if len(hidersFound) >= 1 and len(hidersNotFound) == 0:
56              display.scroll("WIN!")
57              sleep(2000)
58              gameStart = False
59              radio.send("END")
60              hidersFound.clear()
61              hidersStrength.clear()
62              hidersNotFound.clear()
63              rssiHistory.clear()
```