

Rock Paper Scissors - Micropython

Dette spillet bruker to microbits med samme kode. Hvis man trykker a-knappen velger spilleren tilfeldig enten stein, saks eller papir, og den får også valget til den andre spilleren. Etter det blir det regnet ut hvem som vant, og poengsummene blir vist rett etterpå. Man kan også tilbakestille alle poengsummene med b.

Hvilke funksjoner blir brukt i koden?

Funksjoner er prosesser som man kan programmere selv med hjelp av def (define). Disse blir ofte brukt hvis denne prosessen trengs mer enn en gang i koden, sånn at man ikke gjentar noe og sånn at koden er enklere å se på. Dette programmet bruker tre forskjellige funksjoner; CreateAndSendRandomChoice(), Arbitrate() og DisplayScore().

CreateAndSendRandomChoice():

Som navnet forklarer, gjør denne funksjonen at det blir det valgt enten stein, saks eller papir. Variabelen "randomRPS" blir brukt for å lagre dette valget. Det blir brukt random.choice(RPS_LIST), som velger noe tilfeldig fra listen RPS_LIST, som er en liste av alle tre alternativene. Etterpå blir valget sendt til den andre spilleren.

```
30 def CreateAndSendRandomChoice():
31     randomRPS = random.choice(RPS_LIST)
32     radio.send(randomRPS)
33     return randomRPS
```

Arbitrate():

Denne funksjonen regner ut hvem av spillerne som vant. Her blir valgene til begge spillerne sammenlignet og spilleren er enten glad, trist eller forvirret. Dette kommer selvfølgelig an på om den vant eller ikke. Først blir det sjekket om de to valgene er like. Hvis de er det blir spillerne forvirret og det er uavgjort. Da slutter funksjonen. Hvis dette ikke stemmer sjekker spilleren om den har vunnet, så hvis den har valgt stein og den andre saks, hvis den har valgt saks og den andre papir, eller hvis den valgte papir og den andre stein. Da blir 'wins' en større, og spilleren blir glad. Dette ender også funksjonen. Hvis det fortsatt ikke er det finnes det bare en løsning, og det er at spilleren har tapt. Så blir 'losses' en større og spilleren blir trist.

```

35 def Arbitrate(sent, received):
36     global wins
37     global losses
38     if sent == received:
39         display.show(Image.CONFUSED)
40         return
41     if (sent == ROCK and received == SCISSORS) or (sent == SCISSORS and received == PAPER) or (sent == PAPER and received == ROCK):
42         wins += 1
43         display.show(Image.HAPPY)
44         return
45     losses += 1
46     display.show(Image.SAD)

```

DisplayScore():

Denne funksjonen er den enkleste av alle, fordi den bare viser poengsummen av spillerne. Fordi variablene “wins” og “losses” er en integer (et nummer) kan man ikke vise de på skjermen. Derfor må man konvertere de til “strings” med `str()` sånn at man kan se dem, ellers kommer en *TypeError*.

```

48 def DisplayScore():
49     display.scroll(str(wins) + "-" + str(losses))

```

Hva gjør resten av koden?

Fordi vi har mange funksjoner er den egentlige koden ganske kort. Med a-knappen gjør man “sentRPS” om til “CreateAndSendRandomChoice()”. Det vil si at “sentRPS” blir det tilfeldige valget som blir randomisert i funksjonen.

```

53 if button_a.was_pressed():
54     sentRPS = CreateAndSendRandomChoice()

```

Valget til den andre spilleren er lagret under `receivedRPS`. Dette må altså være `radio.receive()`, fordi det er signalet den får fra motstanderen.

Hvis den får et signal (valget til den andre spilleren) så sjekker den først om den har sendt sitt valg allerede. Hvis ikke gjennomfører den `CreateAndSendRandomChoice()`. Etter det viser den sitt eget valg og det fra den andre, og `Arbitrate()` blir gjennomført. Til slutt viser den den nåtidige poengsummen til begge spillerne og `sentRPS` blir tømt.

```
55 receivedRPS = radio.receive()
56 if receivedRPS:
57     if sentRPS == "":
58         sentRPS = CreateAndSendRandomChoice()
59         display.show(RPS_DICTIONARY[sentRPS])
60         sleep(1500)
61         display.show(RPS_DICTIONARY[receivedRPS])
62         sleep(1500)
63         Arbitrate(sentRPS, receivedRPS)
64         sleep(1000)
65         DisplayScore()
66         sentRPS = ""
```

Hvis man trykker b-knappen blir poengsummene slettet og vist.

```
67 if button_b.was_pressed():
68     wins = 0
69     losses = 0
70     DisplayScore()
```