

Mønt samler

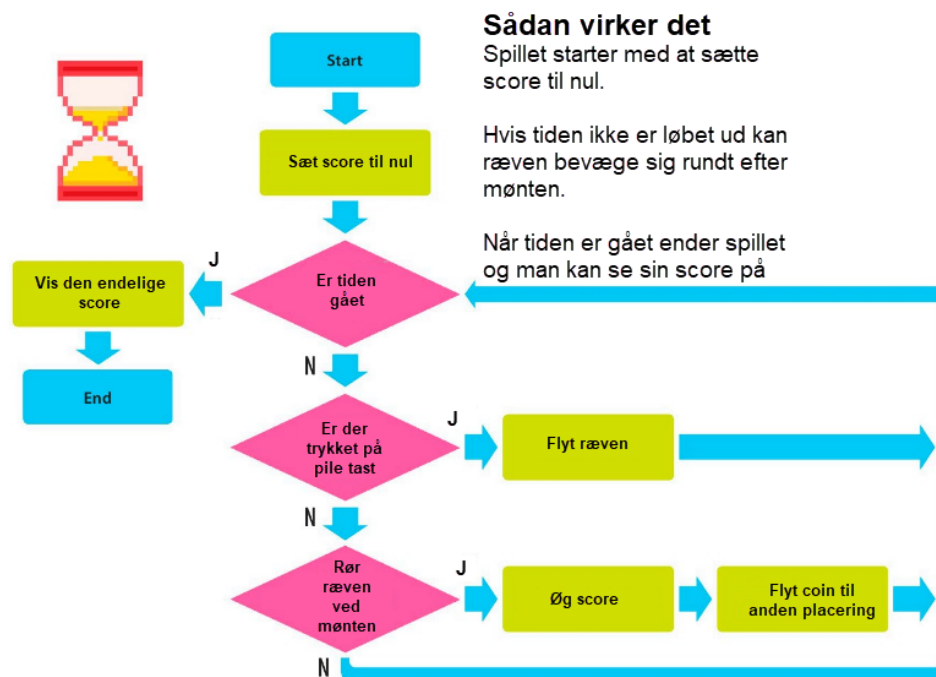


Intro

I dette lille spil skal ræven samle mønter på tid.

For hver mønt der samle får man point og det gælder om få flest mulige point inden tiden er gået.

Program flow



Assets

Til spillet skal der bruges nogle assets.

[coin.png](#)

[fox.png](#)

filerne skal placeres i undermapper til din .py kode fil. Mappen skal hedde "images" således:

```
mu_code folder
    coin_collector.py
    images
        coin.png
        fox.png
```

Koden

Først defineres spil vinduet

```
1 WIDTH = 400
2 HEIGHT = 400
3 TITLE = "Coin Collector"
```

Tilføj et par variabler.

Når spillet starter har man 0 point og man er ikke game over 😊

```
5 score = 0
6 game_over = False
```

Så loader vi spillets to Actor's.

En actor er en sprite som man tildele noget grafik. Her vores fox og coin billeder.

en actor har indbygget metoder vi kan bruge i spillet.

Den første metode er .pos hvor vi sætter en start position for hver actor.

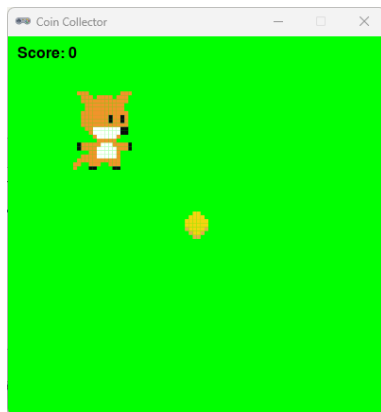
```
8 fox = Actor("fox")
9 fox.pos = 100, 100
10
11 coin = Actor("coin")
12 coin.pos = 200, 200
```

Vi skal have noget på skærmen. Derfor tilføjer vi draw funktionen.
I funktionen sætte vi baggrundsfarve og kalder actor metoden draw.

Vi viser også en statisk tekst (noget som ikke ændre sig) + værdien som er i variabelen score.
placering er 10,10 fra top venstre hjørne af spil vinduet.

```
14 def draw():
15     screen.fill("green")
16     fox.draw()
17     coin.draw()
18     screen.draw.text("Score: " + str(score), color="black", topleft=(10
```

Det er nu tid til at afprøve spillet første gang.
Det skulle gerne se sådan her ud



Hvis ikke det gør. Så gå koden efter og se om der er nogle fejl indtastninger.

Vi skal lidt vider og få lidt bevægelse ind i spillet.
Vi skal bruge disse tre funktioner. Vi starter med at give dem en "pass" instruktion.
Det betyder bare at vi helt bevisst har valgt ikke at lave noget "rigtig" kode
Husk at slette "pass" så snart vi begynder at tilføje kode til funktionen!

```
20 def place_coin():
21     pass
22
23 def time_up():
24     pass
25
26 def update():
27     pass
```

Husk at slette "pass" så snart vi begynder at tilføje kode til funktionen!

For vi skal have programmet til selv at bestemme hvor mønten dukker op.
der for skal vi bruge en metode der hedder randint.
Den skal importeres fra et bibliotek der hedder random.

Helt først i programmet tilføjes denne linje.

```
1 from random import randint
2
```

Ændre nu funktionen place_coin() så den ser sådan her ud.

```
22 def place_coin():
23     coin.x = randint(20, (WIDTH-20))
24     coin.y = randint(20, (HEIGHT-20))
25
```

Kan du regne ud hvad koden gør?

Tilføj denne linje sidst i programmet

```
29 def update():
30     pass
31
32 place_coin()
```

Prøv nu at starter og stoppe spillet et par gange. Hold øje med hvor mønten er på skærmen.

Vi skal have spillet til at stoppe.

Ændre time_up() funktionen så den ser sådan her ud.

```
26 def time_up():
27     global game_over
28     game_over = True
```

Denne linje kode placeres lige før linjen med place_coin()

```
33 clock.schedule(time_up, 7.0)
34 place_coin()
```

Den gør at når spillet starter, så startes en timer på 7 sekunder.

når tiden er gået går programmet til funktionen time_up.

Den funktion har vi lige lavet og den sætter vores game_over variable til True.

Nu skal vi tilføje koden som får spillet til at stoppe.

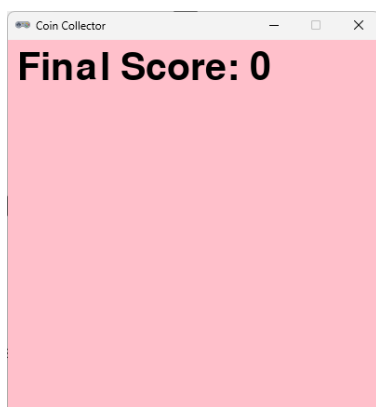
Ændre draw() funktionen så den får if betingelsen med.

```
16 def draw():
17     screen.fill("green")
18     fox.draw()
19     coin.draw()
20     screen.draw.text("Score: " + str(score), color="black", topleft=(10,10))
21
22     if game_over == True:
23         screen.fill("pink")
24         screen.draw.text("Final Score: " + str(score), color="black", topleft=(10,10), fontsize=60)
```

Når skærmen gentegner vil den hvis game_over variabelen er True sætte baggrundsfarven til pink og ændre score teksten.

Afprøv spillet ved at starte det og vente 7 sekunder.

så skal skærmen gerne se sådan her ud



Hvis ikke det gør. Så gå koden efter og se om der er nogle fejl indtastninger.

Vi skal have ræven til at flytte sig.

Ændre update() funktionen så den ser sådan her ud

```
34 def update():
35     global score
36     if keyboard.left:
37         fox.x = fox.x -2
38
39     if keyboard.right:
40         fox.x = fox.x +2
41
42     if keyboard.up:
43         fox.y = fox.y -2
44
45     if keyboard.down:
46         fox.y = fox.y +2
47
```

Afprøv om du kan bevæge ræven med pile tasterne.

hvis du kan så kan du gå videre

Tilpas update() funktionen med denne kode

```
34 def update():
35     global score
36     if keyboard.left:
37         fox.x = fox.x -2
38
39     if keyboard.right:
40         fox.x = fox.x +2
41
42     if keyboard.up:
43         fox.y = fox.y -2
44
45     if keyboard.down:
46         fox.y = fox.y +2
47
48     coin_collected = fox.colliderect(coin)
49
50     if coin_collected == True:
51         score = score + 10
52         place_coin()
53
```

Vi introducer her en ny variable coin_collected og vi sætter den til true hvis ræven rammer mønten.

Når coin_collected variabelen er True så øges score med 10 og vi flytter mønten.

Vores spil er nu færdig, men har man lyst kan man arbejde videre.

Forslag til udvidelse

Øg hvor lang tid man har til spillet

Ændre hastigheden på ræven

Ændre Actor billedet når ræven går til venstre/højre

Vis hvor lang tid der er tilbage i spillet. (hint. update() kaldes 60 gange i sekundet)

Lav bonus tid actors som dukker op på tilfældig tidspunkt og forsvinder igen

Flyt mønten hvis den ikke er taget efter noget tid