

Tegn med kode - del 3: Animasjon

LØSNINGSFORSL
AG

6.-9. trinn · 45 min · SkapLab – Python / pygame

Hva du skal lære

Lage enkel animasjon ved å tegne om igjen for hvert bilde (frame). Forstå begrepene FPS (bilder per sekund) og spill-løkken.

Slik fungerer animasjon

- Hvert bilde i animasjonen kalles en frame
- Du sletter alt (fill) og tegner på nytt for hvert frame
- FPS styrer hvor raskt animasjonen kjører (30-60 er normalt)
- `pygame.time.Clock()` holder FPS-takten stabil

Kodeeksempel - bouncing ball

```
import pygame
pygame.init()
skjerm = pygame.display.set_mode((800, 600))
klokke = pygame.time.Clock()

x, y = 400, 300
fart_x = 4
fart_y = 3

kjører = True
while kjører:
    for h in pygame.event.get():
        if h.type == pygame.QUIT: kjører = False

    # Flytt ballen
    x += fart_x
    y += fart_y

    # Spretts mot kantene
    if x < 20 or x > 780: fart_x = -fart_x
    if y < 20 or y > 580: fart_y = -fart_y

    # Tegn
    skjerm.fill((20, 20, 40))
    pygame.draw.circle(skjerm, (0, 200, 255), (int(x), int(y)), 20)

    pygame.display.flip()
    klokke.tick(60)
```

```
pygame.quit()
```

Oppgaver

1. Kjør koden. Beveger ballen seg og spretter den mot kantene?
2. Øk farten: endre `fart_x` og `fart_y` til 7. Hva skjer?
3. Endre fargen på ballen hvert sekund (hint: bruk en teller).
4. Legg til en andre ball med annen farge og fart.
5. Lag et rektangel som beveger seg frem og tilbake langs bunnen.
6. Utfordring: ballen etterlater seg et spor av sirkler som fader bort.

Løsningsforslag - fargeskiftende ball

```
import pygame, math
pygame.init()
skjerm = pygame.display.set_mode((800, 600))
klokke = pygame.time.Clock()
tid = 0

x, y, fart_x, fart_y = 400, 300, 4, 3

kjører = True
while kjører:
    for h in pygame.event.get():
        if h.type == pygame.QUIT: kjører = False

    x += fart_x
    y += fart_y
    if x < 20 or x > 780: fart_x = -fart_x
    if y < 20 or y > 580: fart_y = -fart_y

    # Fargen varierer med tid (sinus gir glatt overgang)
    r = int(127 + 127 * math.sin(tid * 0.05))
    g = int(127 + 127 * math.sin(tid * 0.05 + 2))
    b = int(127 + 127 * math.sin(tid * 0.05 + 4))

    skjerm.fill((20, 20, 40))
    pygame.draw.circle(skjerm, (r, g, b), (int(x), int(y)), 20)
    pygame.display.flip()
    klokke.tick(60)
    tid += 1

pygame.quit()
```

math.sin() svinger mellom -1 og 1. Ganget med 127 og lagt til 127 gir det en fargesverging mellom 0 og 254 – perfekt for RGB-verdier.