

# INF-1400 Introduksjon til PyGame

21. januar, 2014

## Informasjon

Pygame er en “wrapper” rundt SDL-biblioteket som ble brukt i INF-1100. Det betyr at dere vil kjenne dere igjen i måten man bruker pygame på. Dokumentasjon til pygame finnes på [www.pygame.org](http://www.pygame.org) under *Documentation*. Øverst på siden finner dere en oversikt over alle klassene. Klikk på aktuell klasse for å se hvilke metoder denne har. Dere finner også brukereksempler på hvordan metodene brukes.

Her er et minimalistisk pygame-eksempel for å få dere i gang:

```
import pygame

class Canvas(object):
    # A drawing canvas.
    def __init__(self):
        # Set up the pygame window.
        pygame.init()
        pygame.display.set_mode((400, 300))
        pygame.display.set_caption("Pygame Demo")
        self.screen = pygame.display.get_surface()

    def handle_events(self):
        # Handle all events.
        for event in pygame.event.get():
            if event.type == pygame.QUIT:
                exit()
```

```

def update(self):
    # Change position of objects. Nothing here yet.
    pass

def draw_scene(self):
    # Draw a circle in the window.
    pygame.draw.circle(self.screen, pygame.Color("yellow"), (50, 50), 30, 2)
    pygame.display.flip()

def run(self):
    # The main loop.
    while 1:
        self.handle_events()
        self.update()
        self.draw_scene()

if __name__ == '__main__':
    cv = Canvas()
    cv.run()

```

I hovedløkka tegnes en gul sirkel med sentrum i (50,50), radius 30 pixler og strekbredde 2 pixler.

## 1 Tegning

- (a) Skriv et program der du tegner sirkler av forskjellig størrelse. Prøv å forandre alle parametre i draw-funksjonen.
- (b) Bruk pygame til å tegne OL-ringene.

## 2 Animasjon

I denne oppgaven tar du utgangspunkt i eksempelprogrammet over.

- (a) Legg til to variabler i initmetoden, `self.pos_x` og `self.pos_y` og gi begge startverdi 50.

- (b) Endre posisjons tuppelen i linjen som tegner sirkelen, til å bruke de to nye variablene.
- (c) Legg til kode i updatemetoden som endrer en eller begge posisjonsvariablene.
- (d) Endre `draw_scene` slik at den gamle sirkelen blir visket ut (bruk fillmetoden til `self.screen`, som er en instans av `pygame.Surface`-klassen).

### 3 Events

Fortsetter med koden fra forrige oppgave.

- (a) Legg til 4 nye variabler i `init`: `left_pressed`, `right_pressed`, `up_pressed`, `down_pressed`. Initialiser disse til `False`.
- (b) Legg til event-sjekker for om piltastene er trykket ned eller sluppet opp i `handle_events`. Test etter `pygame.KEYUP` og `pygame.KEYDOWN` som `event.type`. Deretter hent ut verdiene i `event.key` (piltastene er `pygame.K_UP`, `pygame.K_DOWN`, `pygame.K_RIGHT` og `pygame.K_LEFT`). Se <http://pygame.org/docs/ref/event.html> for mer dokumentasjon.
- (c) I piltast testene, gjør endring til `_pressed` variablene ut ifra hvilken piltast som er trykket ned eller sluppet opp.
- (d) I `update`, bruk `_pressed` variablene til å vite hvilken piltaster er trykket ned, og gjør tilsvarende endringer til `pos_x` og `pos_y`.