



# Asteroidespillet, Del 4



## Introduksjon

I denne oppgaven skal vi endelig gjøre romskipspillet vårt til et spill. Denne gangen skal vi nemlig få inn poengsystemet og kollisjon med asteroidene!

Vi starter med koden fra [del 3](#). Pass på at alle bildene fremdeles finnes sammen med koden.

## 1. Lag spillskjelettet

Koden skal nå se slik ut:

```
import pygame
import random

# Variable for vindusstørrelsen, slik at vi slipper å endre mange steder hvis vi vil bytte størrelse
display_width = 600
display_height = 800

# Start opp pygame slik at det kan brukes. Viktig!
pygame.init()

class Spaceship(pygame.sprite.Sprite):
    def __init__(self):
        pygame.sprite.Sprite.__init__(self)

        self.image = pygame.image.load("spaceship.png")

        self.rect = self.image.get_rect()

        self.rect.centerx = 300
        self.rect.centery = 700

    def updatePosition(self, x_speed):
        self.rect.centerx += x_speed

        if self.rect.right > display_width:
            self.rect.right = display_width

        if self.rect.left < 0:
            self.rect.left = 0

class Asteroid(pygame.sprite.Sprite):
    def __init__(self):
        pygame.sprite.Sprite.__init__(self)

        self.image = pygame.image.load("asteroid.png")

        self.rect = self.image.get_rect()

        self.rect.centerx = random.randrange(0, display_width)
        self.rect.centery = -200

    def updatePosition(self, y_speed):
        self.rect.centery += y_speed

# Start pygame-klokka. Denne holder rede på tiden i spillet vårt
clock = pygame.time.Clock()
```

```

# Sett opp spillvinduet
gameDisplay = pygame.display.set_mode((display_width,display_height))
pygame.display.set_caption('Asteroid Run')

# Les inn bakgrunnsbildet
bgImg = pygame.image.load("background.png")

# Tegn bakgrunnsbildet
def drawBackground():
    gameDisplay.blit(bgImg, (0, 0))

ship = Spaceship()

# Lag en liste av "sprites" og legg til romskipet vårt i listen.
sprites_list = pygame.sprite.Group()
sprites_list.add(ship)

asteroid = Asteroid()
sprites_list.add(asteroid)

x_change = 0
y_speed = 5

# Start "hovedløkka" i spillet
finished = False
while not finished:
    # Sørg for at denne løkken går 60 ganger i sekundet
    clock.tick(60)

    # Sjekk hendelser
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            finished = True
        if event.type == pygame.KEYDOWN:
            if event.key == pygame.K_LEFT:
                x_change = -5
            if event.key == pygame.K_RIGHT:
                x_change = 5
        if event.type == pygame.KEYUP:
            if event.key == pygame.K_LEFT or event.key == pygame.K_RIGHT:
                x_change = 0

    ship.updatePosition(x_change)
    asteroid.updatePosition(y_speed)

    drawBackground()
    sprites_list.draw(gameDisplay)

    # Oppdater vinduet med all grafikken som skal tegnes
    pygame.display.update()

    if asteroid.rect.top > display_height:
        sprites_list.remove(asteroid)

        asteroid = Asteroid()
        sprites_list.add(asteroid)

        # Gjør neste asteroide litt raskere!
        y_speed += 1

# Når vi hopper ut av løkka, avslutter vi spillet.
pygame.quit()

```

I denne versjonen har vi lagt inn en linje som gjør asteroidene raskere for hver gang. I *if*-testen på om vi skal lage ny asteroide har vi nå følgende kode:

```
# Gjør neste asteroide litt raskere!  
y_speed += 1
```

## 2. Skriv tekst på skjermen

Før vi kan gi poeng til spilleren må vi lære hvordan vi skriver tekst på skjermen i pygame. Det er dessverre ikke like enkelt som å skrive ut med `print`.

I pygame må vi nemlig *tegne* teksten som grafikk. For å få til dette, trenger vi en skrifttype. Dette er en fil som du må laste ned, så høyreklikk på linken under og velg "Lagre lenken som...". Lagre filen på samme sted som du har koden din.

<https://raw.githubusercontent.com/kkringerike/asteroidrun/master/elektra.otf>

Så lager vi oss en funksjon, legg den rett etter `drawBackground`-funksjonen:

```
def drawText(text, x, y, size, color):  
    font = pygame.font.Font('elektra.otf', size)  
    textImage = font.render(text, True, color)  
    textRect = textImage.get_rect()  
    textRect.center = (x, y)  
    gameDisplay.blit(textImage, textRect)
```

Denne funksjonen tar hele 5 argumenter! Det første argumentet er selve teksten vi skal skrive, de neste to er posisjonen på skjermen. Så kommer tekststørrelsen, og til slutt kommer fargen. For å sette en farge i pygame, bruker du en *liste* av tre tall. Disse tallene angir hvor mye rødt, grønt og blått det skal være i fargen din. Tallene går fra 0 til 255, så du kan mikse og blande alle mulige farger med disse tre tallene.

Legg inn disse to fargene et sted utenfor hovedløkka:

```
white = (255, 255, 255) # Alle fargene på fullt blir hvitt.  
red = (255, 0, 0) # Rødt på fullt, grønt og blått på 0, gir en knall rødfarge.
```

Legg til den neste linjen på samme sted som du tegner all grafikken, inne i hovedløkka. Pass på at den kommer etter `drawBackground()`.

```
drawText("Hello World", display_width/2, 15, 20, white)
```

☐ Kjør programmet og se hva som skjer. Du skal nå få opp spillet med en hvit tekst i toppen av vinduet.

## Tips

Prøv deg frem med forskjellige farger. Hva blir f.eks (0, 255, 0) eller (128, 128, 128)?

## 3. Gi poeng

Nå skal vi lage poengsystemet. Vi gir 100 poeng for hver asteroide vi klarer å unngå. Altså, for hver gang vi lager en ny asteroide skal vi øke poengsummen med 100.

Lag først en variabel for poengene. Vi setter den utenfor hovedløkka:

```
score = 0
```

Så øker vi poengene med 100 for hver nye asteroide. Legg til i if-testen hvor vi lager nye asteroider:

```
score += 100
```

(Husk at `+=` betyr "det jeg hadde + et tall")

Så skal vi skrive ut poengene for spilleren. Endre "Hello World"-linja til å istedet si:

```
drawText("Score: " + str(score), display_width/2, 15, 20, white)
```

☐ Kjør programmet og se hva som skjer. Du skal nå få 100 poeng hver gang en asteroide forsvinner ut av bildet

## 4. Kræsj!

Foreløpig er det jo ikke mulig å feile i dette spillet. Asteroidene flyr bare forbi romskipet. For å gjøre det til et spill må vi sjekke om romskipet kræsjer i asteroidene.

Heldigvis har pygame laget en fin funksjon i Sprite-klassen som kan sjekke om en sprite kolliderer med en annen sprite. Denne funksjonen heter `collide_mask` og den tar to sprites som argumenter, og returnerer `True` dersom disse to spriteene er borti hverandre, og `False` hvis de ikke gjør det.

Vi kan derfor lage oss en *if*-test som vi legger i hovedløkka, legg den rett før vi tester om asteroiden har gått ut av vinduet:

```
if pygame.sprite.collide_mask(spaceship, asteroid):
    drawText("You Crashed!", display_width/2, 300, 50, red)

    # Oppdater vinduet
    pygame.display.update()

    # Vent to sekunder
    time.sleep(2)

    # Lag ny asteroide med standard fart
    sprites_list.remove(asteroid)
    aster_speed = 7
    asteroid = Asteroid()
    sprites_list.add(asteroid)

    # Lag nytt romskip og nullstill score
    sprites_list.remove(ship)
    score = 0
    ship = Spaceship()
    sprites_list.add(ship)
```

Her bruker vi først tekstfunksjonen vår til å skrive ut tekst på skjermen. Vi må kalle `pygame.display.update()` for at denne teksten skal synes.

Så venter vi i 2 sekunder. Her bruker vi *time*-biblioteket, så vi må legge inn en ny `import` øverst i koden vår:

```
import time
```

Deretter fjerner vi asteroiden og lager en ny med standard fart, før vi til slutt lager et nytt romskip og nullstiller scoren.

- ☐ Kjør programmet og se hva som skjer. Du skal nå ha et ferdig spill som går ut på å unngå å kræsje i asteroider! Se hvor mange poeng du klarer å få!

## Utfordringer

- ☐ Kan du lage et High-Score-system? Når du kræsjer kan det stå f.eks "High Score: (beste poengsum)", og "New High Score: (poeng)" hvis du slo den beste poengsummen.