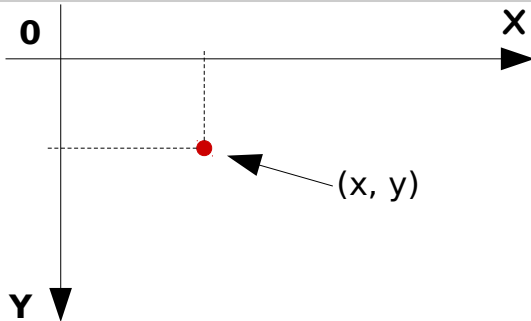


## Setup

- `init()` → initialiserer pygame modulet
- `display.set_mode` → sæt skærmstørrelse
- `display.flip` → opdaterer skærmen

```
import pygame, sys, random
pygame.init()
X = 900
Y = 600
# scr - skærm Surface
scr = pygame.display.set_mode([X, Y])
# flip opdaterer skærmen
pygame.display.flip()
```



## Farver

RGB (rød, grøn, blå)

0 = ingen farve, 255 = fuld farve

```
RED = (255, 0, 0)
GREEN = (0, 255, 0)
BLUE = (0, 0, 255)
WHITE = (255, 255, 255)
BLACK = (0, 0, 0)
YELLOW = (255, 255, 0)
```



Coding Pirates

# Pygame

Basis

## Surface og billeder

Surface kan oversættes til 'overflade'  
Et objekt der repræsenterer ethvert  
slags 'billede'/'overflade' i pygame

```
# scr er et Surface objekt
```

```
# fyld med en farve
scr.fill(YELLOW)
```

```
# størrelses operationer
print('Bredde', scr.get_width())
print('Højde', scr.get_height())
print('Størrelse', scr.get_size())
```

```
# load et billede som Surface objekt
img = pygame.image.load('pirate.gif')
```

```
# optimer/konverter til pixel format
# der kan blittes hurtigt
# convert_alpha bevarer alpha-bits
# (transparency-bits)
img = img.convert_alpha()
```

```
# blit (BLock Image Transfer) et
# Surface ind på et andet Surface
# her ind på skærmen
scr.blit(img, [100, 200])
```

```
pygame.display.flip()
```

## Tegne operationer

Pygame.draw modulet indeholder  
operationer til at tegne på et  
Surface

```
# cirkler i forskellige farver og
# størrelser
# [100, 100] = centrum i x, y
# 50 = cirkel radius
pygame.draw.circle(scr, BLACK,
[100, 100], 50)
```

```
pygame.draw.circle(scr, BLUE,
[80, 80], 10)
pygame.draw.circle(scr, BLUE,
[120, 80], 10)
pygame.draw.circle(scr, RED,
[100, 110], 10)
```

```
# rektangel (firkant)
# [75, 130, 50, 5] =
# [x, y, bredde, højde]
pygame.draw.rect(scr, WHITE,
[75, 130, 50, 5])
```

```
# linie
# [50, 150] = start x, y
# [150, 150] = stop x, y
# 2 = linie tykkelse
pygame.draw.line(scr, RED,
[50, 150], [150, 150], 2)
```

```
pygame.display.flip()
```



## Gratis open source programmer/links

- GIMP billedbehandling: <http://gimp.org>
- Audacity lyd: <http://audacityteam.org>
- Gratis lydklip: <http://freesound.org>
- Gratis grafik: <http://openclipart.org>



Coding Pirates

# Pygame

Events

## Events

Operativ systemet sender 'events' til programmet i en event kø  
Event køen skal løbes igennem og tømmes i game loop'et  
Forskellige slags events: QUIT, KEYDOWN/UP, MOUSEMOTION,  
MOUSEBUTTONDOWN/UP, JOY... (joystick)

```
# game loop - kør for "evigt", indtil quit
while True:
    pygame.time.wait(20)
    # event handling → løb alle events i event køen igennem (tømmer køen)
    for event in pygame.event.get():
        # quit event, lukker programmet
        if event.type == pygame.QUIT:
            pygame.quit()
            sys.exit()
        # mus / mouse events → tegn cirkler hvor der er klikket (event.pos)
        elif event.type == pygame.MOUSEBUTTONDOWN:
            if event.button == 1: # 1 = venstre museknap → tegn blå cirkel
                pygame.draw.circle(scr, BLUE, event.pos, 3)
            elif event.button == 3: # 3 = højre museknap → tegn hvid cirkel
                pygame.draw.circle(scr, WHITE, event.pos, 3)
        # tastatur / key events
        elif event.type == pygame.KEYDOWN:
            if event.key == pygame.K_SPACE:
                # blit en tekst på muse positionen
                scr.blit(hallo_text, pygame.mouse.get_pos())
        # der findes også Joystick events
    # husk flip
    pygame.display.flip()
```

## Tekst og fonte

Skrivning af tekst sker ved at man anvender et font-objekt til at generere et Surface objekt, som så blittes ind på skærmen

```
# et font objekt for Arial fonten,
# skriftstørrelse 20
fnt = pygame.font.SysFont('arial', 20)

# render (lav) et Surface objekt
# med teksten 'hallo' i blå
hallo = fnt.render('hallo', True, BLUE)

# blit ind på skærmen
scr.blit(hallo, [150, 200])
```

## Musik og lyde

Musik til baggrunds musik skal være en MP3 fil eller en WAV fil

```
pygame.mixer.music.load('lyd.mp3')
# hvor mange gange musikken
# skal spilles -1 = forevigt
pygame.mixer.music.play(-1)
```

Lyde til events (fx et skud eller en kollision). Load som Sound objekt og afspil senere. Skal være en WAV fil

```
hit = pygame.mixer.Sound('hit.wav')
# afspil 1 gang - fx ved kollision
hit.play()
```





Coding Pirates

# Pygame

Game Loop

## Game Loop

De fleste spil bygger på samme skabelon med et game loop

```
# initialiser pygame
# definer farve konstanter, mm.
# load billeder, fonte og musik
pygame.init()
..
# game loop - kørs for 'evigt'
while True:
    # 20 ms wait
    pygame.time.wait(20)

    # 1. håndter events
    for event in pygame.event.get():
        ..
    # 2. beregn nye objekt positioner

    # 3. detekter kollisioner

    # 4. kald funktioner der tegner
    # objekter - gerne flere funktioner
    # til at tegne forskellige objekter:
    # - baggrunds objekter
    # - fjender
    # - spilleren

    # opdater skærmen med flip
    pygame.display.flip()
```

## Objekter og bevægelse

Brug et dictionary til hvert object.

```
ball = { 'x': 100, 'y': 200,
         'dx': 2, 'dy': 3 }

# game loop
while True:
    pygame.time.wait(20)

    # 2. beregn nye objekt positioner
    ball['x'] += ball['dx']
    ball['y'] += ball['dy']

    # 3. detekter kollisioner
    # med siderne
    if ball['x'] < 0 or ball['x'] > X:
        # vend 'x' retning
        ball['dx'] = -ball['dx']
    if ball['y'] < 0 or ball['y'] > Y:
        # vend 'y' retning
        ball['dy'] = -ball['dy']

    # 4. tegn - baggrund og bold
    scr.fill(YELLOW)
    pygame.draw.circle(scr, BLUE,
                      [ball['x'], ball['y']], 10)

    pygame.display.flip()
```

## Rect objekt

Pygame.Rect objekter kan anvendes til smarte kollisions operationer på firkantede objekter

```
# load et billede som Surface objekt
img = pygame.image.load('pirate.gif')
img = img.convert_alpha()

# få Rect objektet
pirate = img.get_rect()
# sæt positionen for Rect objektet
pirate.center = [100, 200]

# blit ind på Rect positionen
scr.blit(img, pirate)

# operation til at flytte Rect
pirate.move_ip(1, -3)

# blit ind på den nye Rect position
scr.blit(img, pirate)

# check om to Rect overlapper
if pirate.colliderect(enemy):
    print('piraten er fanget!')

# der er andre collide.. metoder
# kig i pygame dokumentationen
pirate.collidepoint(..)
pirate.collidelist(..)
```

