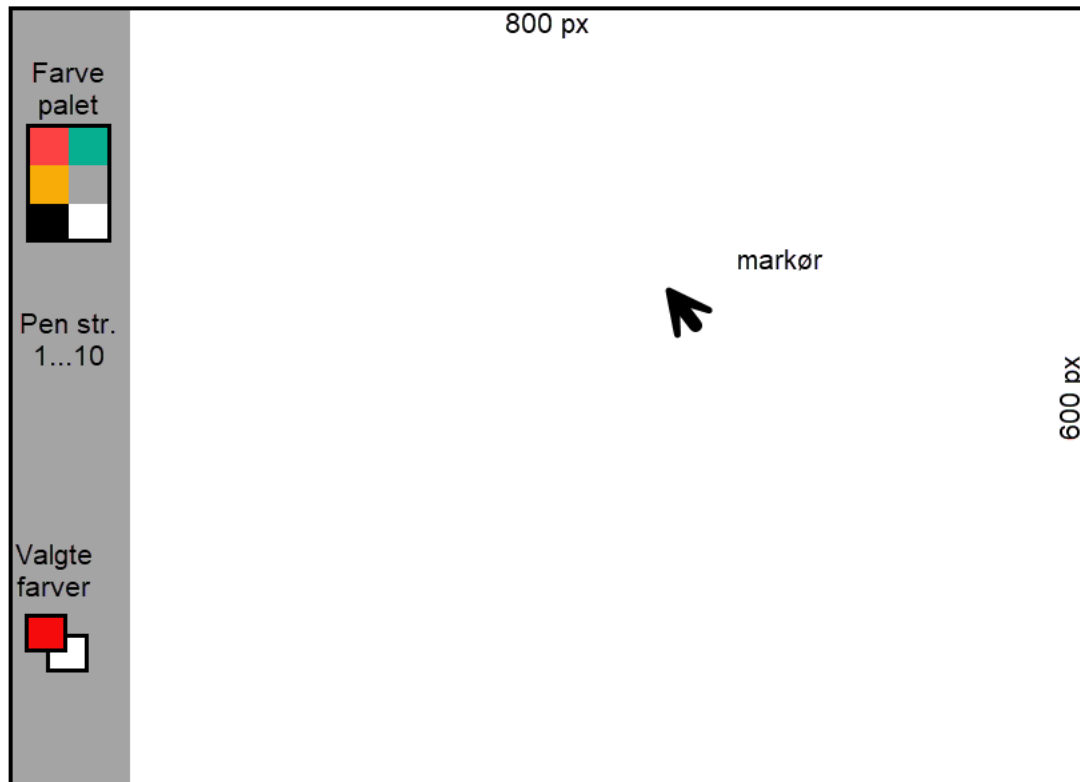


# Tegneprogram

Et lille tegne program skrevet i Python Pygame Zero

## Del 1: Layout

Målet er at lave layout som dette



## Program Vinduet

Definer program vinduet til 800 x 600

I en tom fil tilføj følgende kode.

Husk kommentar da vi senere vil henvise til dem som afsnit af koden.

```
#Paint program

#Program Vindue
WIDTH = 800
HEIGHT = 600
TITLE = "Mit Tegneprogram"
```

gem filen med navnet "Paint.py" og test koden.

## Farver i program

Lad os definere nogle farver for brug i programmet

Tilføj et nyt afsnit efter afsnittet "Program Vindue"

```
#Farver
RED = 200, 0, 0
GREEN = 0,200,0
BLACK = 0,0,0
WHITE = 255,255,255
```

Og lad os teste at det virker ved at tilføje dette lige bagefter

```
#Kode herfra
def draw():
    screen.fill(WHITE)
```

test programmet og se vinduet nu har hvid baggrund

*Opgave:*

Tilføj farverne LIGHTGREY, ORANGE, GREY i afsnittet "Farver"

## Menu område

Tilføj denne funktion til sidst i filen

```
def draw_sidebar():
    sidebar = Rect((0, 0), (120, HEIGHT))
    screen.draw.filled_rect(sidebar, LIGHTGREY)
    screen.draw.rect(sidebar, BLACK)
```

og tilpas draw() functionen

```
def draw():
    screen.fill(WHITE)
    draw_sidebar()
```

Test at det virker

## Farve Palet

På samme måde som vi lige har tilføjet menuen vil vi nu tilføje nogle mindre firkanter med farverne ovenfor.

Tilføj denne funktion sidst i filen

```
def draw_palet():
    colorbox = Rect((20, 20), (40, 40))
    screen.draw.filled_rect(colorbox, RED)

    colorbox = Rect((60, 20), (40, 40))
    screen.draw.filled_rect(colorbox, GREEN)
```

og tilføj kald til den i draw()

```
def draw():
    screen.fill(WHITE)
    draw_sidebar()
    draw_palet()
```

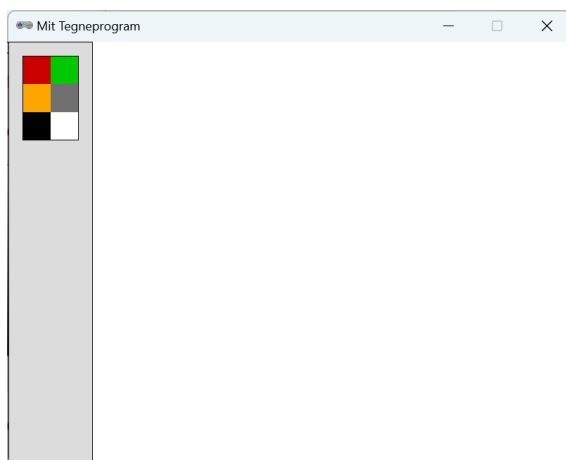
Test at du har to farver ved siden af hinanden i menu området.

### Opgave

Tilføj bokse for de sidste fire farver

Sæt en sort ramme rundt om alle farverne

Nu skal programmet gerne se således ud



### Valgte farver

Vi fortsætter med bokse for valgte farver. Ideén er at man kan se hvad farver man maler med når man kommer til at klikke på venstre og højre museknap.

Tilføj først dette afsnit lige under afsnittet "Farver"

```
#Pen
COLORLEFT = BLACK
COLORRIGHT = WHITE
```

og i bunden af filen denne funktion

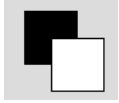
```
def draw_chosen_color():
    colorbox = Rect((30, HEIGHT-100), (40, 40))
    screen.draw.filled_rect(colorbox, COLORLEFT)
    screen.draw.rect(colorbox, BLACK)

    colorbox = Rect((50, HEIGHT-80), (40, 40))
    screen.draw.filled_rect(colorbox, COLORRIGHT)
    screen.draw.rect(colorbox, BLACK)
```

Tilføj kald til den nye funktion i draw() funktionen.

Test koden.

det ser ikke rigtigt ud



den hvide boks skal gerne være bagved den sort

### Opgave

Byt rundt på rækkefølgende af den sort og den hvid boks

## Pen størrelse

Først skal vi sætte en start pen størrelse

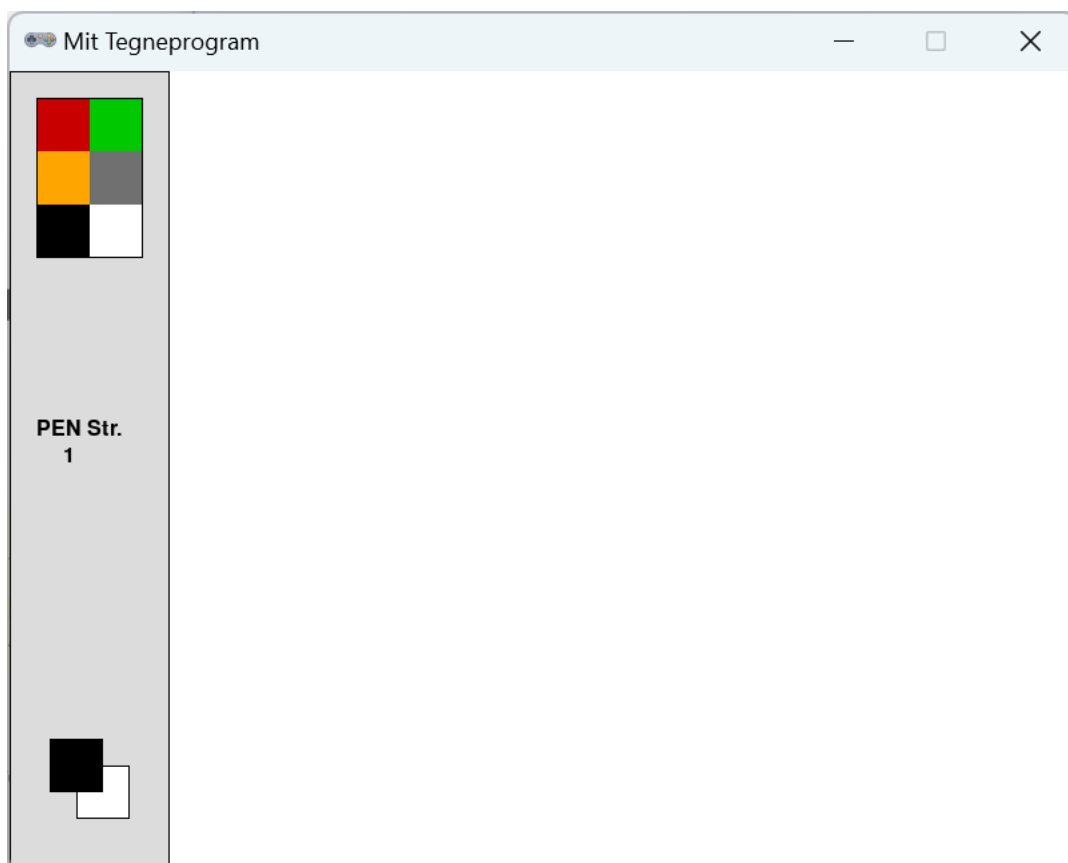
Tilføj i "Pen" afsnitte denne linje

```
PENSIZE = 1
```

Tilføj denne funktion nederst i filen

```
def draw_pen_size():
    screen.draw.text("PEN Str.", (20, HEIGHT/2-40), color=BLACK)
    screen.draw.text(str(PENSIZE), (40, HEIGHT/2-20), color=BLACK)
```

Tilføj kald til funktionen i draw() og test at dit program ser sådan her ud



## Del 2: Tastatur tryk

### Vælg pen størrelse

Vi vil lave det sådan at man kan vælge pen størrelse ved tryk på tasterne 1...0 tilføj nederst i filen denne event

```
def on_key_down(key):  
    if key == keys.K_1:  
        PENSIZE = 1  
  
    if key == keys.K_2:  
        PENSIZE = 2  
  
    if key == keys.K_0:  
        PENSIZE = 10
```

Test om det det virker.

Det gør det ikke. Kan du se hvorfor ikke?

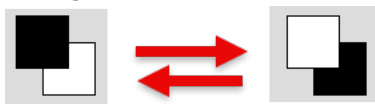
#### Opgave

Tilpas program så pen størrelse teksten virker når man trykker på henholdsvis 1,2 og 0

Tilføj kode for ændring af størrelser til 3,4,5,6,7,8 og 9

### Skift om på de to valgte farver

Vi vil gerne lave det sådan at man kan skifte rundt på de to farver



når man klikker på mellemrumstasten

Tilføj denne kode i bunden af on\_key\_down eventen

```
if key == keys.SPACE:  
    temp = COLORLEFT  
    COLORLEFT = COLORRIGHT  
    COLORRIGHT = temp
```

Og i toppen af eventen tilføj de to variabler

```
def on_key_down(key):  
    global PENSIZE, COLORLEFT, COLORRIGHT  
    if key == keys.K_1:
```

Test om programmet virker som det skal

## Del 3: Tegn med musen

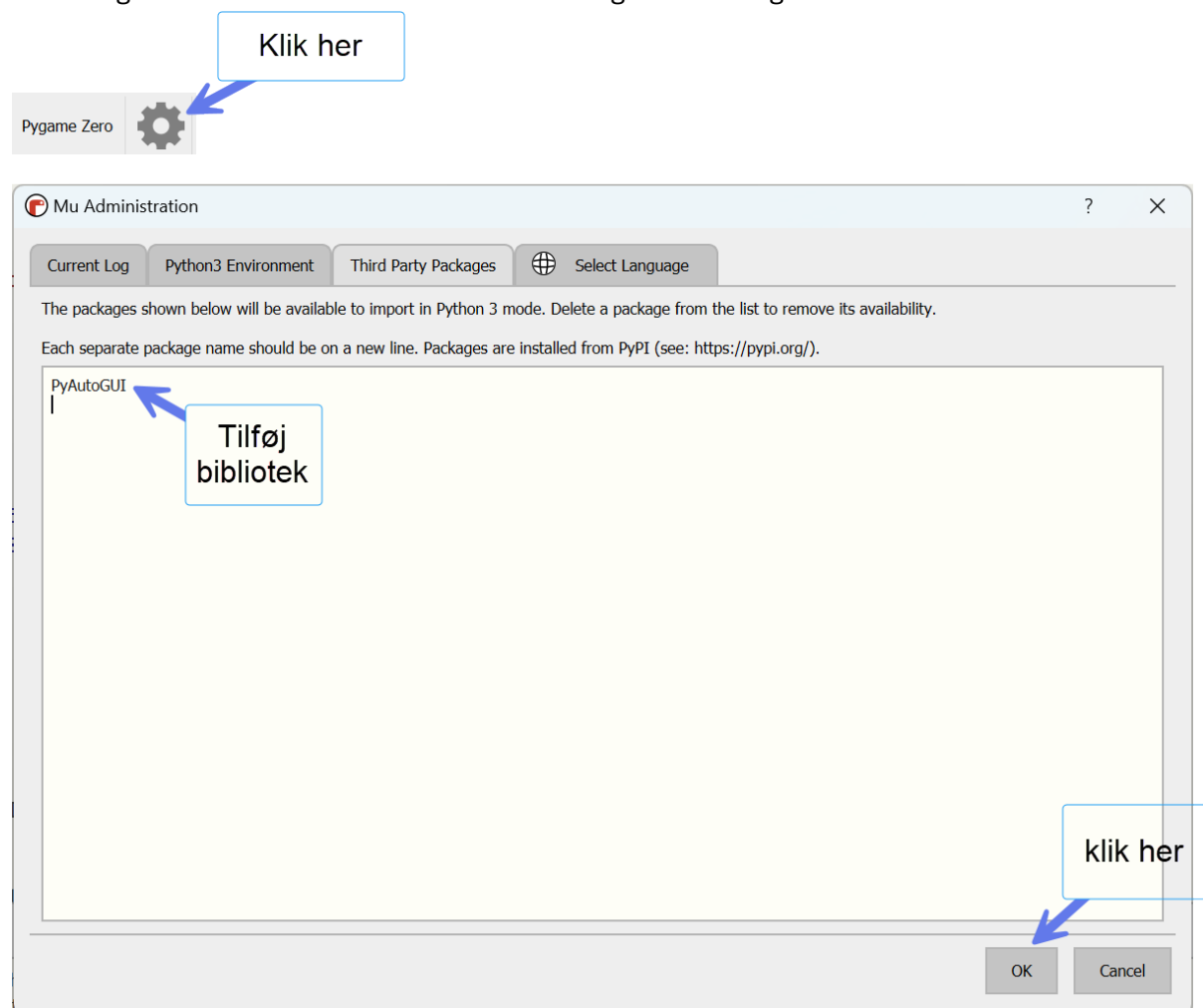
### Vælg farver

Man skal kunne køre musen henover farverne på vores farve palet og hvis man klikker henholdsvis venstre eller højre muse knap ned, så skal farverne vælges.

For at kunne detektere hvilken farve der er under musen skal vi bruge et eksternt bibliotek tilføj dette afsnit lige efter afsnittet "Paint Program"

```
#External libraries
import pyautogui
```

Vi skal også have dette bibliotek installeret. Det gøres i settings



Tilføj nu denne event i bunden af filen

```
def on_mouse_down(pos, button):
    global COLORLEFT, COLORRIGHT
    if (pos[0] > 20 and pos[0] < 100 and pos[1] > 20 and pos[1] < 100):
        x, y = pyautogui.position()
        if button == mouse.LEFT:
            COLORLEFT = pyautogui.pixel(x, y)
        if button == mouse.RIGHT:
            COLORRIGHT = pyautogui.pixel(x, y)
```

Test at det virker.

### Opgave

Forklare hvad koden gør i den nye event?

## Tegn med musen

For at tegne indsætter vi følgende event i bunden af filen

```
def on_mouse_move(pos, buttons):
    if mouse.LEFT in buttons:
        screen.draw.filled_circle(pos, PENSIZE, COLORLEFT)
    if mouse.RIGHT in buttons:
        screen.draw.filled_circle(pos, PENSIZE, COLORRIGHT)
```

Prøv at se om det virker.

Det gør det ikke.

### Opgave

Kan du regne ud hvorfor det ikke virker?

Prøv at finde funktionen draw() og sæt # foran linjen med screen.fill(WHITE)

Prøv at se om det virker nu

Det gør det, men nu er baggrunden sort så den farve kan man ikke tegne med.

Så det skal vi have gjort noget ved.

Der er flere ting som skal ændres. Udfordringen er at hver gang draw() bliver kaldt, så bliver alt på skærmen gentaget. Men det vi har tegnet med musen er ikke gemt. Så det skal vi have lagret. Til det skal vi bruge et array.

Tilføj følgende lige efter afsnittet "Pen"

```
#Tegnede Elementer
paitings = []
```

I dette array vil vi gemme alle de elementer som bliver tegnet på skærmen.

til at tegne skal vi bruge position, pen størrelse og farve.

Så til det opretter vi en klasse. En klasse er lidt som en opskrift. Eks. En person vil have hårfarve, højde, køn

Indsæt følgende efter afsnittet "Tegnede Elementer"

```
#Klasser
class Paint():
    def __init__(self):
        self.pos = None
        self.pensize = None
        self.color = None
```

Nu skal vi ændre eventen on\_mouse\_move så den kommer til at se sådan her ud

```
def on_mouse_move(pos, buttons):
    p = Paint()
    p.pos = pos
    p.pensize = PENSIZEN

    if mouse.LEFT in buttons:
        p.color = COLORLEFT
        paintings.append(p)

    if mouse.RIGHT in buttons:
        p.color = COLORRIGHT
        paintings.append(p)
```

Den sikre at alt hvad der skal tegnes bliver lagt ind i vores paintings array, så vi skal have lavet en funktion som tegner alle de instrukser vi har lagt der ind.

Tilføj i bunden af filen denne nye funktion

```
def paint():
    global paintings
    for p in paintings:
        screen.draw.filled_circle(p.pos, p.pensize, p.color)
```

Og før vi kan teste det skal vi lige have tilføjet paint() til draw()

Glem ikke at fjerne # fra linjen fill.screen(WHITE)

draw() skal se sådan her ud



```
def draw():
    screen.fill(WHITE)
    draw_sidebar()
    draw_palet()
    draw_chosen_color()
    draw_pen_size()
    paint()
```

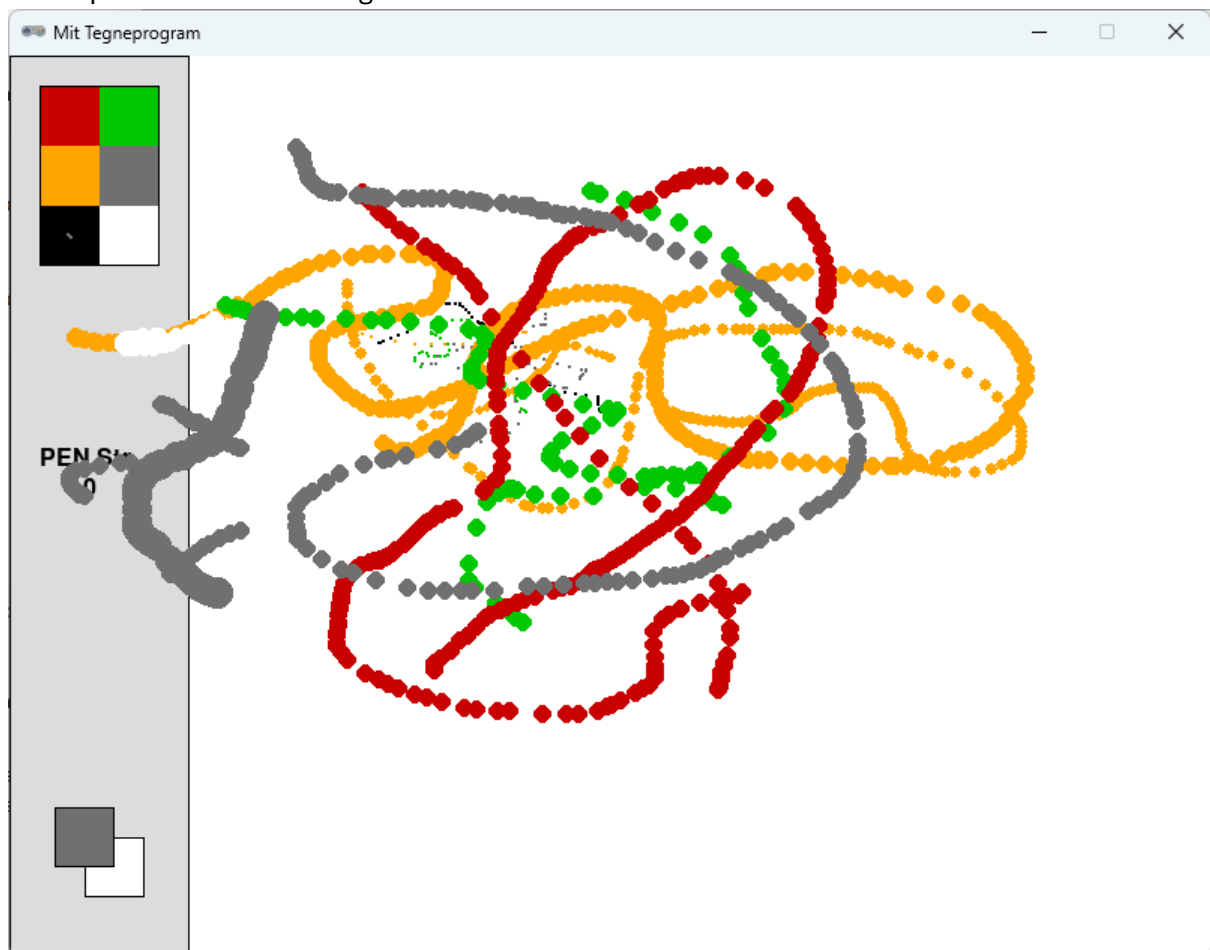
Prøv ny programmet af og se om det virker.

### Opgave

Kan du finde nogen ting der kan optimeres?

## Del 4: Optimering

Der er plads til lidt forbedringer.



Der er to ting som springer i øjnene.

1. Pen størrelse er lidt lille
2. Man kan tegne ind over menuen.

## Pen faktor

Vi kan tilføje en variable for pen faktor. Med den kan vi prøve os lidt frem til hvilken faktor der giver et ønsket resultat. Ved at bruge en variable skal vi kun ændre et sted i koden.

I afsnittet "Pen" tilføj følgende

```
PENSIZEFACTOR = 2
```

I eventen `on_mouse_move` find linjen

```
p.pensize = PENSIZE
```

Og ændre den til

```
p.pensize = PENSIZE * PENSIZEFACTOR
```

Prøv nu og se om der har gjort en forskel. Prøv med forskellige værdier for faktoren.

## Tillad ikke at tegne ind over menuen

Da vi lavede menuen satte vi dens bredde til 120px

Men hvis vi skal være opmærksom på at når vi tegner så er position på vores pen center punktet. så der skal tages højde for pen størrelsens brede (faktisk kun halvdelen, men så nøjsomme er vi ikke)

Ændre `on_mouse_move` så den kommer til at se sådan her ud.

```
def on_mouse_move(pos, buttons):
    if pos[0] > (120 + PENSIZE * PENSIZEFACTOR):
        p = Paint()
        p.pos = pos
        p.pensize = PENSIZE * PENSIZEFACTOR

        if mouse.LEFT in buttons:
            p.color = COLORLEFT
            paitings.append(p)

        if mouse.RIGHT in buttons:
            p.color = COLORRIGHT
            paitings.append(p)
```

Prøv nu og se om det er blevet bedre.

Du bør nu have en mindre tegne program.

## Opgave

Kom med forslag til funktioner til tegne programmet.