Galaktisk Galskab i PyGame Zero – Workshop 3

I workshop 2 tegnede vi stjerner som Actor-objekter og fik rumskibet til at skyde på en ufo, som flyttede sig, når den blev ramt. I workshop 3 skal vi lave skiftende stjerner, ufoer og eksplosioner.

Stjerner med forskellig grafik

Tegnigen af de enkelte stjerner ser lidt kedelig ud, fordi alle er ens. Men vi kan få forskellige stjerner, ved at bruge `random.choice` til at vælge en tilfældig fil. Hent den nye grafik i `images` på https://github.com/stee0640/pgzero2025/ og erstat `'stjerne.png'`-koden med noget i retningen af denne kode.

```
stjerne_grafik = [
    's1.png', 's2.png', 's3.png','s4.png', 's5.png', 's6.png'
]

for _ in range(100):
    ny_stjerne = random.choice(stjerne_grafik)
    stjerne = Actor(ny_stjerne)
```

- Blit og scroll evt. '`baggrund.png`' som i workshop 1 for at skabe en parallax-effekt

Brug uret til at skabe angribende ufoer

Funktionen `clock.schedule` herunder starter en timer, som kalder funktionen `skab_ufo` efter fem sekunder og derefter igen hvert halve sekund. Ufoerne bliver indsat i en liste. Husk at kalde `draw`-funktionen for hver ufo i listen fra `draw`-callback for at få den tegnet.

```
ufo_liste = []
def skab_ufo():
    ny_ufo = Actor("ufo1")
    ny_ufo.x = random.randint(0, WIDTH)
    ny_ufo.bottom = 0
    ny_ufo.vx = random.randint(-10,10)
    ny_ufo.vy = random.randint(2, 10)
    ufo_liste.append(ny_ufo)
    clock.schedule(skab_ufo, 0.5)

clock.schedule(skab_ufo, 5)
```

Ufoerne bliver altid placeret i toppen af skærmen, mens deres placering på x-aksen er tilfældig. Hver ufos hastighed vx og vy er også tilfældig. For at bevæge ufoerne og fjerne dem, når de kommer uden for skærmen, skal du kalde funktionen herunder fra `update`-callback.

```
def ufoer_update():
    for ufo in ufo_liste:
        ufo.angle += 1
        ufo.x += ufo.vx
        ufo.y += ufo.vy
        if ufo.left <= 0 or ufo.right >= WIDTH:
            ufo.vx = -ufo.vx
        if ufo.top > HEIGHT:
            ufo_liste.remove(ufo)
```

Animation af eksplosioner

De følgende to funktioner kan skabe eksplosioner på en bestemt position og lave en animation på 15 frames (et kvart sekund ved 60 frames/sek). Eksplosionens billede bliver skiftet ud i løbet af animationen. Kald `eksplosion_update` fra `update`-callback Husk også at kalde en `eksplosion.draw()` fra `draw`-callback for alle eksplosioner i listen.

```
eksplosion_liste = []

def skab_eksplosion(position):
    eksplosion = Actor('eksplosion50.png', pos = position)
    eksplosion.frames = 15
    eksplosion_liste.append(eksplosion)

def eksplosion_update():
    for eksplosion in eksplosion_liste:
        eksplosion.frames -= 1
        if eksplosion.frames < 0:
            eksplosion_liste.remove(eksplosion)
        elif eksplosion.frames == 12:
            eksplosion.image = 'eksplosion75.png'
        elif eksplosion.frames == 8:
            eksplosion.image = 'eksplosion100.png'
```

Vi skal skabe eksplosionerne, når en ufo bliver ramt. Derfor skal du kalde følgende kode, der opdaterer dine skud fra `update`-callback. Når en ufo bliver ramt, bliver den erstattet med en eksplosion på samme position, som ufoen.

```
rumskib.score = 0

def skud_update():
    for skud in skud_liste:
        skud.y -= 6
        if skud.bottom < 0:
            skud_liste.remove(skud)
        for ufo in ufo_liste:
            if skud.colliderect(ufo):
                if skud in skud_liste: # Fjern kun skud én gang
                    skud_liste.remove(skud)
                ufo_liste.remove(ufo)
                rumskib.score += 1
                skab_eksplosion(ufo.pos)
```

Koden ovenfor gemmer antallet af ramte ufoer i en egenskab `score` i `rumskib`-objektet. For at få den skrevet på skærmen skal følgende funkrion kaldes fra `draw`-callback.

```
def score_draw():
    screen.draw.text(str(rumskib.score),
        color="white", midtop=(WIDTH//2, 10),
        fontsize=70, shadow=(1, 1)
    )
```

Ufoerne kan stadig ikke ramme rumskibet. Det skal vi lave om på i næste workshop.