

1) Code tot nu toe

Deze opdracht wordt afgetekend als:

- Je voor alle blokken code in je schrift hebt geschreven wat ze doen

Vorige week hebben we code gebschreven om elementen van de simulatie te laten groeien. Lees de code goed en schrijf weer in je schrift wat ieder blok doet.

Let op! Niet alle code staat in dit overzicht, lees dus goed. De codes staan ook niet in de goede volgorde.

Schrijf weer per blok in je schrift wat de code doet. Je mag hiervoor samenwerken met je buur. Je mag ook zeker kijken bij je uitwerkingen van vorige week.

```
----- Blok 1 -----  
x, y = int(x * 1.05), int(y * 1.05)
```

```
----- Blok 2 -----  
if muis_rechthoek.colliderect(kaas_rechthoek):  
    print('hap hap')
```

```
----- Blok 3 -----  
muis_rechthoek.center = (300, 400)
```

```
----- Blok 4 -----  
x, y = muis_rechthoek.size
```

```
----- Blok 5 -----  
center = muis_rechthoek.center
```

```
----- Blok 6 -----  
muis = pygame.transform.scale(muis, (x, y))  
muis_rechthoek = muis.get_rect()  
muis_rechthoek.center = center
```

```
----- Blok 7 -----  
muis = pygame.image.load("muis.png")
```

2) Maak de functie voor 'omdraaien' (als je dit nog niet had)

Deze opdracht wordt afgetekend als:

- Je in je schrift de overeenkomsten en verschillen tussen de 2 stukken code hebt opgeschreven.
- Je een functie hebt gemaakt voor omdraaien aan de rand.

In je programma zit nu veel herhaling, code die op elkaar lijkt. We willen liever code die twee of drie keer voorkomt in een **functie** zetten. Bijvoorbeeld deze twee stukken code:

```

---- stukje 1 ----
if muis_rechthoek.bottom < 0:
    muis_snelheid[1] = - muis_snelheid[1]

if muis_rechthoek.bottom > 600:
    muis_snelheid[1] = - muis_snelheid[1]

if muis_rechthoek.right < 0:
    muis_snelheid[0] = - muis_snelheid[0]

if muis_rechthoek.right > 800:
    muis_snelheid[0] = - muis_snelheid[0]

---- stukje 2 ----

if leeuw_rechthoek.bottom < 0:
    leeuw_snelheid[1] = - leeuw_snelheid[1]

if leeuw_rechthoek.bottom > 600:
    leeuw_snelheid[1] = - leeuw_snelheid[1]

if leeuw_rechthoek.right < 0:
    leeuw_snelheid[0] = - leeuw_snelheid[0]

if leeuw_rechthoek.right > 800:
    leeuw_snelheid[0] = - leeuw_snelheid[0]

```

Kijk goed naar de code en beantwoordt deze vragen in je schrift.:

- Welke stukjes zijn steeds hetzelfde?
- Welke stukjes zijn anders?

Nu gaan we een functie maken. De begint zo:

```

def draai_om_aan_de_rand(dier_rechthoek, dier_snelheid):
    if dier_rechthoek.top < 0:
        dier_snelheid[1] = - dier_snelheid[1]

    # hier komt de rest van de code

```

Maak jij de code af?

3) Gebruik de functie voor 'omdraaien'

Deze opdracht wordt afgetekend als:

- Je de nieuwe functie gebruikt voor het omdraaien aan de rand van **beide** dieren.

Nu je de functie hebt gemaakt, moet je deze ook om in je code "aanroepen". Dat betekent de functienaam gebruiken met de goede *argumenten*. Dat moet je doen voor beide dieren.

```
while True:
    # deze codes staan er al bij jou:
    pygame.display.flip()
    screen.fill(backgroundcolor)

    screen.blit(muis, muis_rechthoek)
    muis_rechthoek = muis_rechthoek.move(muis_snelheid)

    #hier komt de aanroep van de functie draai_om_aan_de_rand voor de muis

    # deze code staat er ook al:
    screen.blit(leeuw, leeuw_rechthoek)
    leeuw_rechthoek = leeuw_rechthoek.move(leeuw_snelheid)

    #hier komt de aanroep van de functie draai_om_aan_de_rand voor de leeuw
```

4) Functie voor groeien

Deze opdracht wordt afgetekend als:

- Je ook voor het groeien van beide dieren een functie gemaakt hebt.

Er zit nog meer herhaling in je code! Ook de codes voor het groeien van de dieren lijken op elkaar.

Volg deze stappen om de functie te maken:

1. Kijk goed welke regels er verschillen en overeenkomen
2. Regels die hetzelfde zijn komen in de nieuwe functie.
3. Kopieer de code voor de muis en vervang in de regels 'muis' door 'dier'
4. De functie begint zo:

```
def groei(dier):
    #hier komen de regels die hetzelfde zijn
```

5. Vergeet niet ook 2 keer een aanroep voor de code te maken!

5) Dieren vermenigvuldigen

Deze opdracht wordt afgetekend als:

- Je twee dieren van dezelfde soort in het veld hebt gezet (dus twee leeuwen or twee muizen of twee hyenas)
- Je collision detection tussen deze twee dieren hebt geprogrammeerd
- Je zorgt dat als twee dieren elkaar raken, er een derde dier van die soort in beeld komt (een baby).

In de natuur komen natuurlijk ook wel eens nieuwe dieren ter wereld. Dat ga jij programmeren. Lees je code goed en pas die aan waar nodig.

6) Eigen stempeldoel

Deze opdracht wordt afgetekend als:

- Je zelf een doel hebt bedacht, en dat hebt uitgevoerd

Uitleg. Verzin zelf iets wat er nog bij kan in je simulatie. Een achtergrond? Een extra dier? Een kangaroo met een bokshandschoen? Bedenk iets en schrijf dat in je schrift. Voer het vervolgens netjes uit. Vraag wel eerst of mevr. Hermans of meneer van Ooijen je doel goed vindt.