

Figure 1: Boek 2

#	Omschrijving
5	line en stroke
6	Bal die eeuwig rechts gaat
7	rect en fill

Contents

Voorwoord	1
line en stroke	2
Bal die eeuwig naar rechts gaat	22
rect en fill	38

Voorwoord



Figure 1: Het logo van De Jonge Onderzoekers

Dit is het Processing boek van de Dojo. Processing is een programmeertaal. Dit boek leert je die programmeertaal.

Over dit boek

Dit boek heeft een CC-BY-NC-SA licentie.



Figure 2: De licentie van dit boek

(C) Dojo Groningen 2016-2018

Het is nog een beetje een slordig boek. Er zitten tiepvauten in en de opmaak is *niet altijd even mooi*.

Daarom staat dit boek op een GitHub. Om precies te zijn, op <https://github.com/richelbilderbeek/Dojo>. Hierdoor kan iedereen die dit boek te slordig vindt minder slordig maken.

line en stroke



Figure 3: Moria, een van de allereerste games met kleur

In deze les gaan we leren hoe je gekleurde lijnen tekent.

line en stroke: opdracht 1

Run deze code:

```
void setup()
{
  size(300, 200);
}

void draw()
{
  line(0, 100, 300, 200);
}
```



`line(0, 100, 300, 200);`

‘Lieve computer, teken een lijn van (0, 100) naar (300, 200).’



(100, 200) is de pixel die 100 pixels naar links en 200 pixels onder de linkerbovenhoek van het scherm zit

line en stroke: oplossing 1

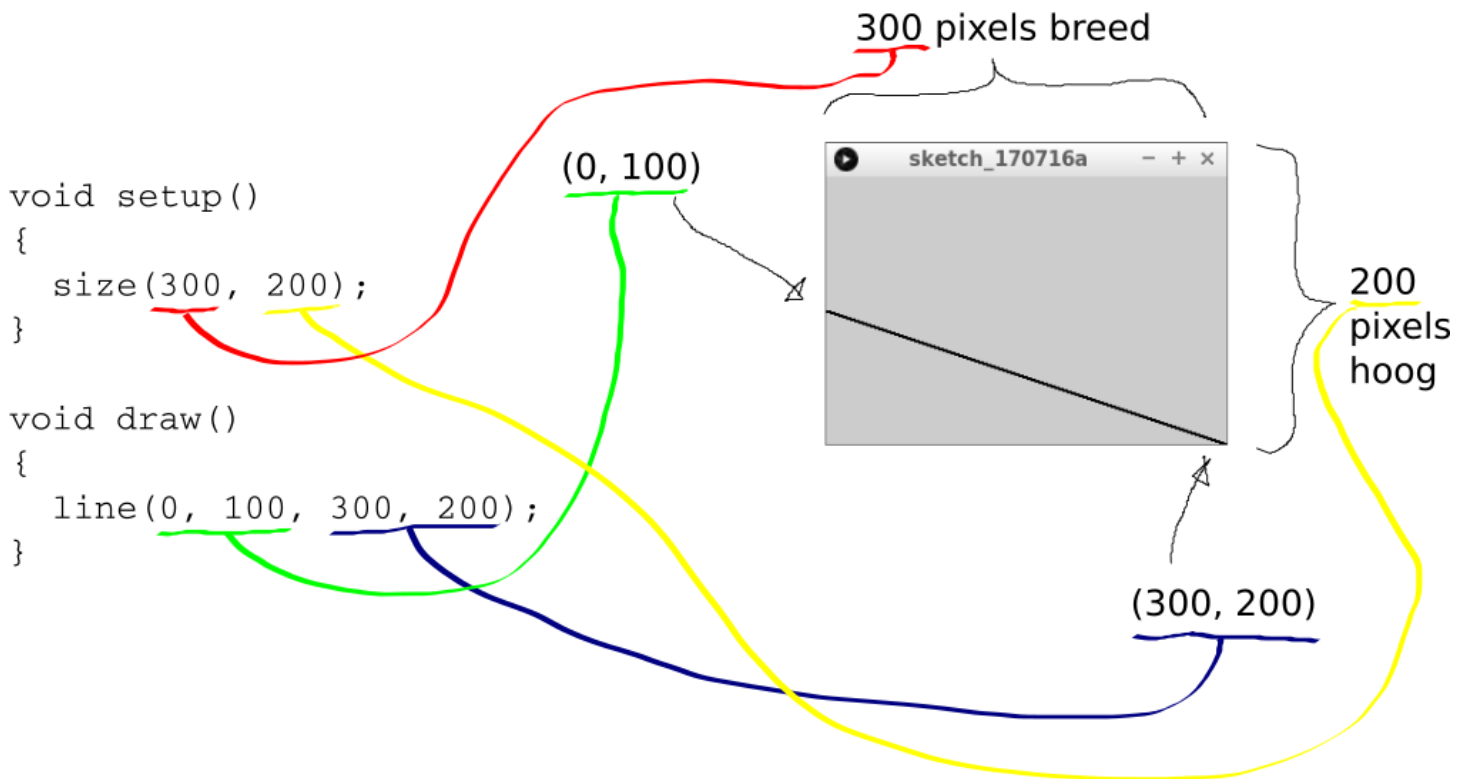


Figure 4: Oplossing 1

line en stroke: opdracht 2

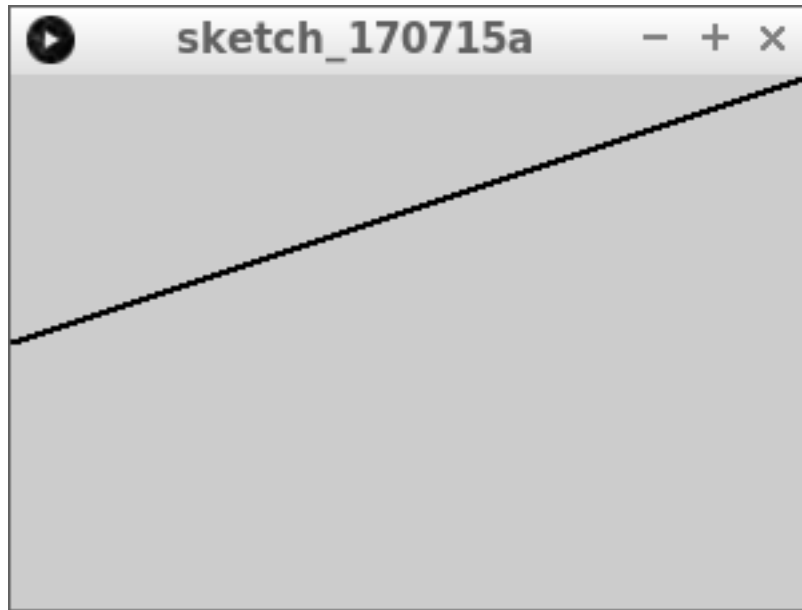


Figure 5: Opdracht 2

Laat de lijn niet naar de rechteronderhoek, maar naar de rechterbovenhoek gaan.

line en stroke: oplossing 2

```
void setup()
{
  size(300, 200);
}

void draw()
{
  line(0, 100, 300, 0);
}
```

line en stroke: opdracht 3

Laat de lijn niet links in het midden, maar linksonder beginnen

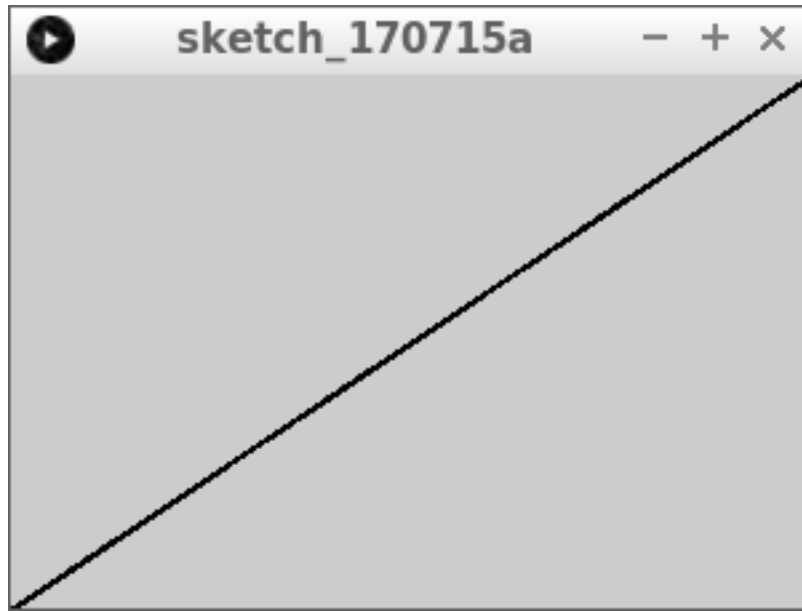


Figure 6: Opdracht 3

line en stroke: oplossing 3

```
void setup()
{
  size(300, 200);
}

void draw()
{
  line(0, 200, 300, 0);
}
```

line en stroke: opdracht 4

Laat de lijn van linksonder naar rechtsboven gaan, maar gebruik nu `width` en `height`

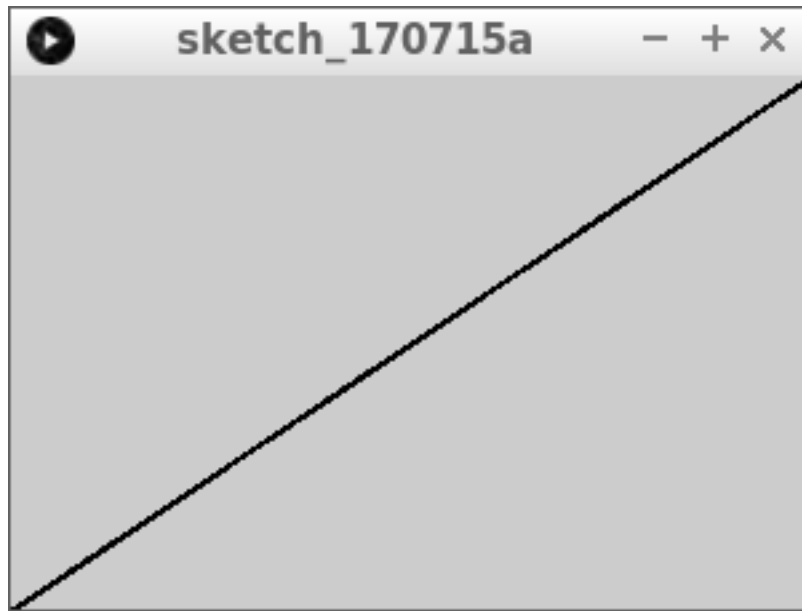


Figure 7: Opdracht 4

Oplossing 4

```
void setup()
{
  size(300, 200);
}

void draw()
{
  line(0, height, width, 0);
}
```

line en stroke: opdracht 5

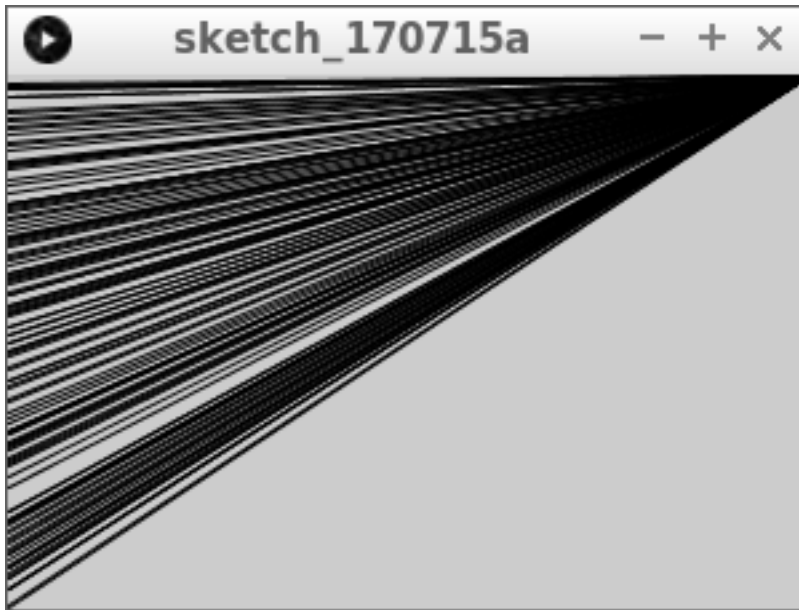


Figure 8: Opdracht 5

Laat de lijn links op een willekeurige hoogte beginnen. Dit doe je met `random`

Oplossing 5

```
void setup()
{
  size(300, 200);
}

void draw()
{
  line(0, random(height), width, 0);
}
```

line en stroke: opdracht 6



Figure 9: Opdracht 6

Laat de lijn nu ook rechts op een willekeurige hoogte eindigen.

line en stroke: oplossing 6

```
void setup()
{
  size(300, 200);
}

void draw()
{
  line(0, random(height), width, random(height));
}
```

line en stroke: opdracht 7

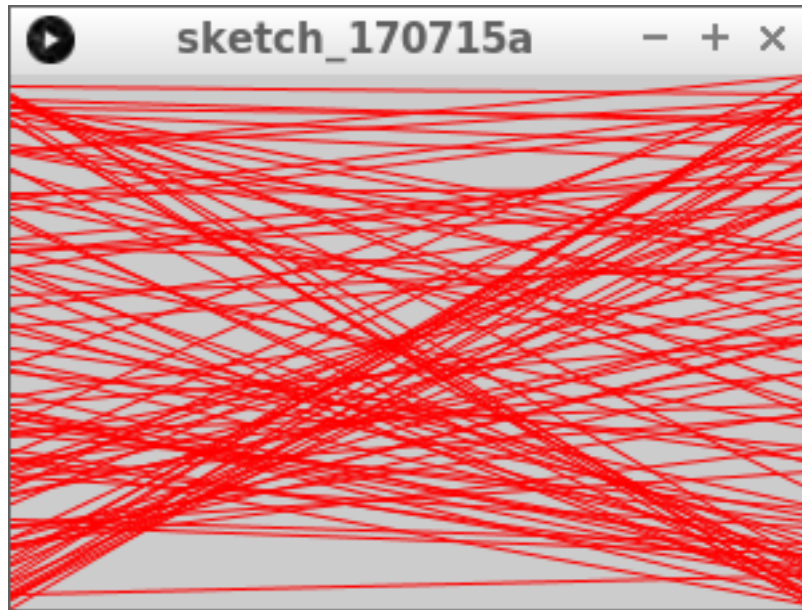


Figure 10: Opdracht 7

Zet voor line de regel `stroke(255, 0, 0);`

line en stroke: oplossing 7

```
void setup()
{
  size(300, 200);
}

void draw()
{
  stroke(255, 0, 0);
  line(0, random(height), width, random(height));
}
```



```
stroke(255, 0, 0);
stroke(255, 0, 0);
```

‘Lieve computer, geef de lijnen de kleur rood.’
‘Lieve computer, geef de lijnen de kleur vol rood, zonder groen en
zonder blauw.’

line en stroke: opdracht 8

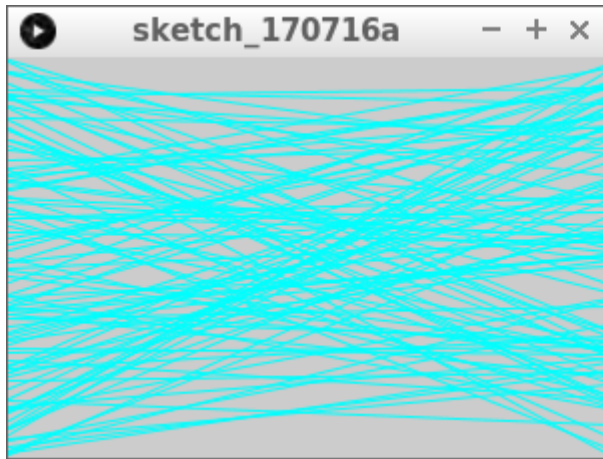


Figure 11: Opdracht 8

Maak de lijnen nu cyaan. Kijk naar figuur Kleurencirkel hoe je die maakt

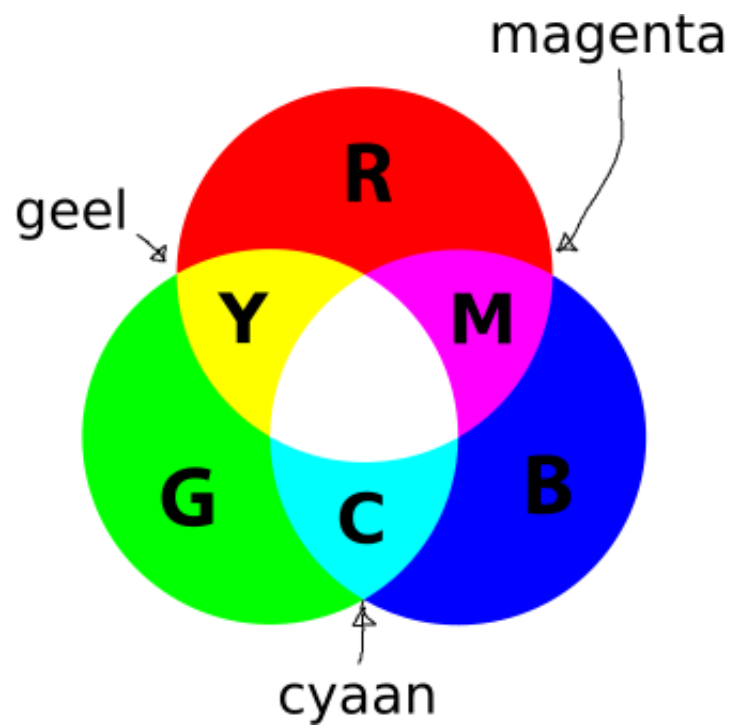


Figure 12: Kleurencirkel

line en stroke: oplossing 8

```
void setup()
{
  size(300, 200);
}

void draw()
{
  stroke(0, 255, 255);
  line(0, random(height), width, random(height));
}
```



```
stroke(0, 255, 255);
stroke(0, 255, 255);
```

‘Lieve computer, geef de lijnen de kleur cyaan.’
‘Lieve computer, geef de lijnen de kleur zonder rood, vol groen en vol
blauw.’

line en stroke: opdracht 9

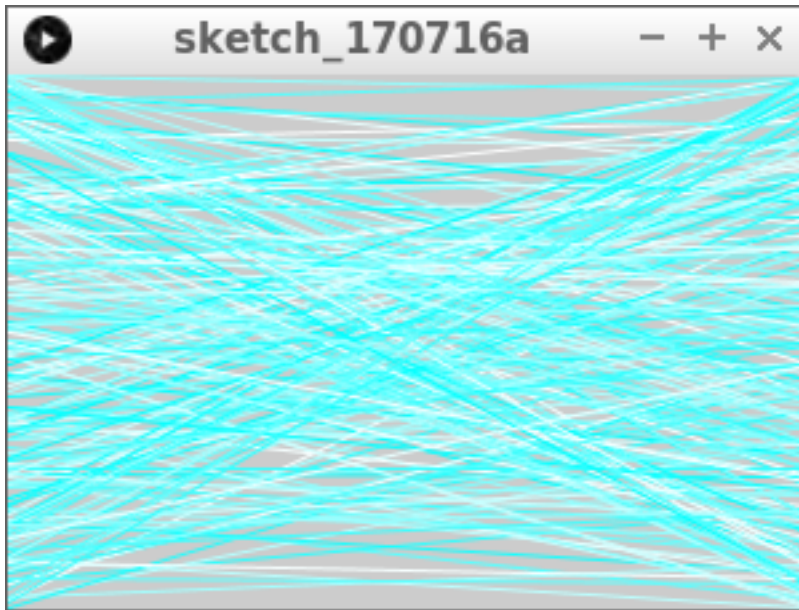


Figure 13: Opdracht 9

Laat nu de roodwaarde een willeukeurig getal van 0 tot 256 worden.

line en stroke: oplossing 9

```
void setup()
{
  size(300, 200);
}

void draw()
{
  stroke(random(256), 255, 255);
  line(0, random(height), width, random(height));
}
```

line en stroke: eindopdracht

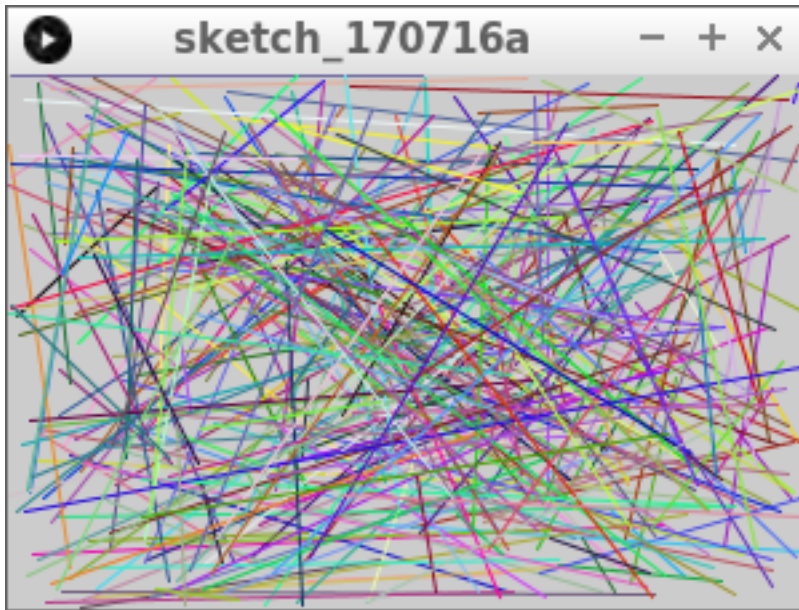


Figure 14: Eindopdracht `line` en `stroke`

Laat de lijnen nu op willekeurige plekken beginnen en eindigen. De lijnkleur moet ook willekeurig zijn.

line en stroke: links

- `line` en `stroke`: YouTube, mp4

Bal die eeuwig naar rechts gaat

In deze les gaan we een bal eeuwig naar rechts laten gaan.

We leren in deze les wat `if`-statement is. Je kunt (bijna) niet programmeren zonder `if`-statements.

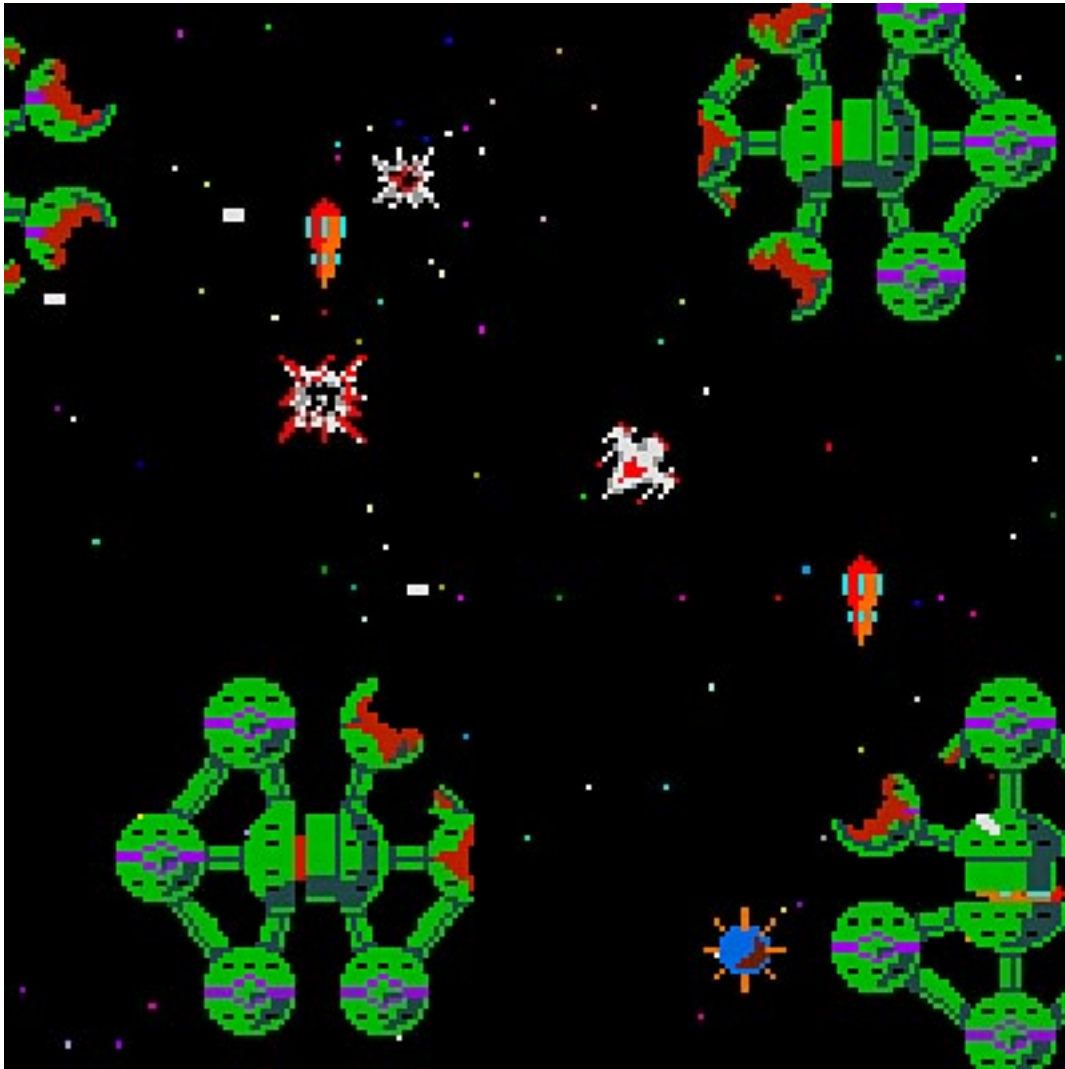


Figure 15: In Bosconian kun je ook eeuwig naar rechts gaan

Bal die eeuwig naar rechts gaat: intro

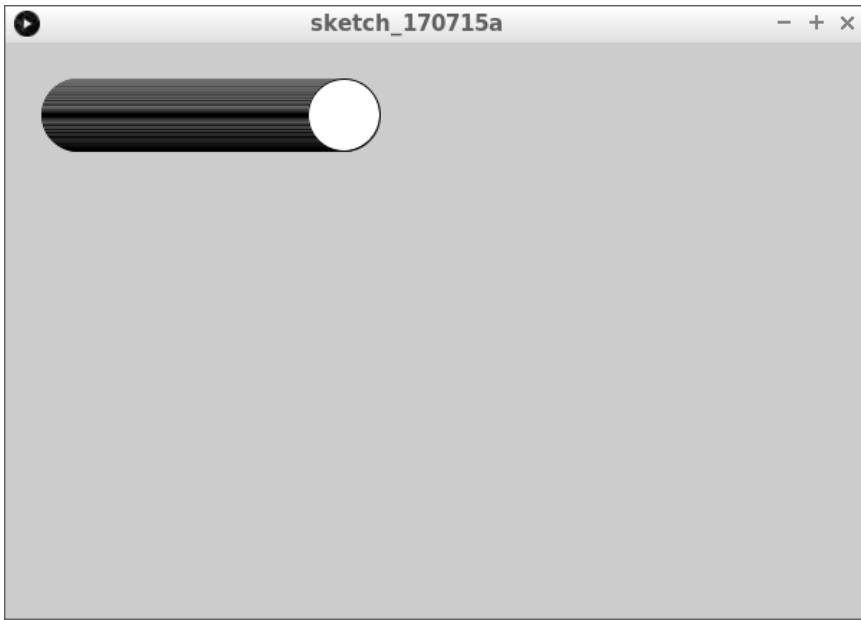


Figure 16: Verkleurende bal naar rechts

Dit is een bal die naar rechts gaat:

```
float x = 50;

void setup()
{
  size(600, 400);
}

void draw()
{
  ellipse(x, 50, 50, 50);
  x = x + 1;
}
```

Nadeel: de bal komt nooit meer terug in het scherm.

We willen kunnen zeggen: ‘Lieve computer, *als* de bal te ver naar rechts is, dan teleporteer je de bal naar rechts’. `if` is Engels voor ‘als’.

Zo zou dit kunnen:

```
if (x > 200)
{
    x = 100;
}
```

Het teken > betekent ‘groter dan’. Preciezer zeg je: ‘Lieve computer, *als* x meer is dan 200, zet x dat op 100’. `if` is Engels voor ‘als’.

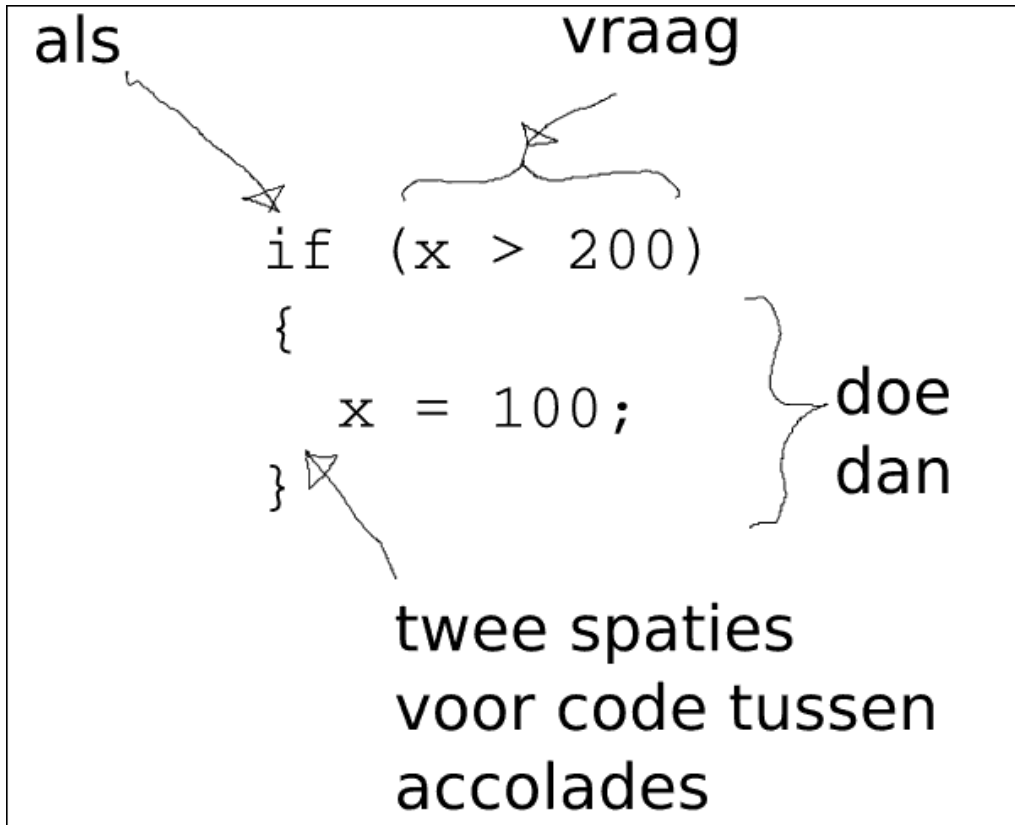




Figure 17: Een `if`

	
<code>if (x > 200) { }</code>	‘Lieve computer, als x groter is dan 200, doe dan wat tussen accolades staat.’
<code>x = 100;</code>	‘Lieve computer, stop het getal 100 in x.’

Bal die eeuwig naar rechts gaat: opdracht 1

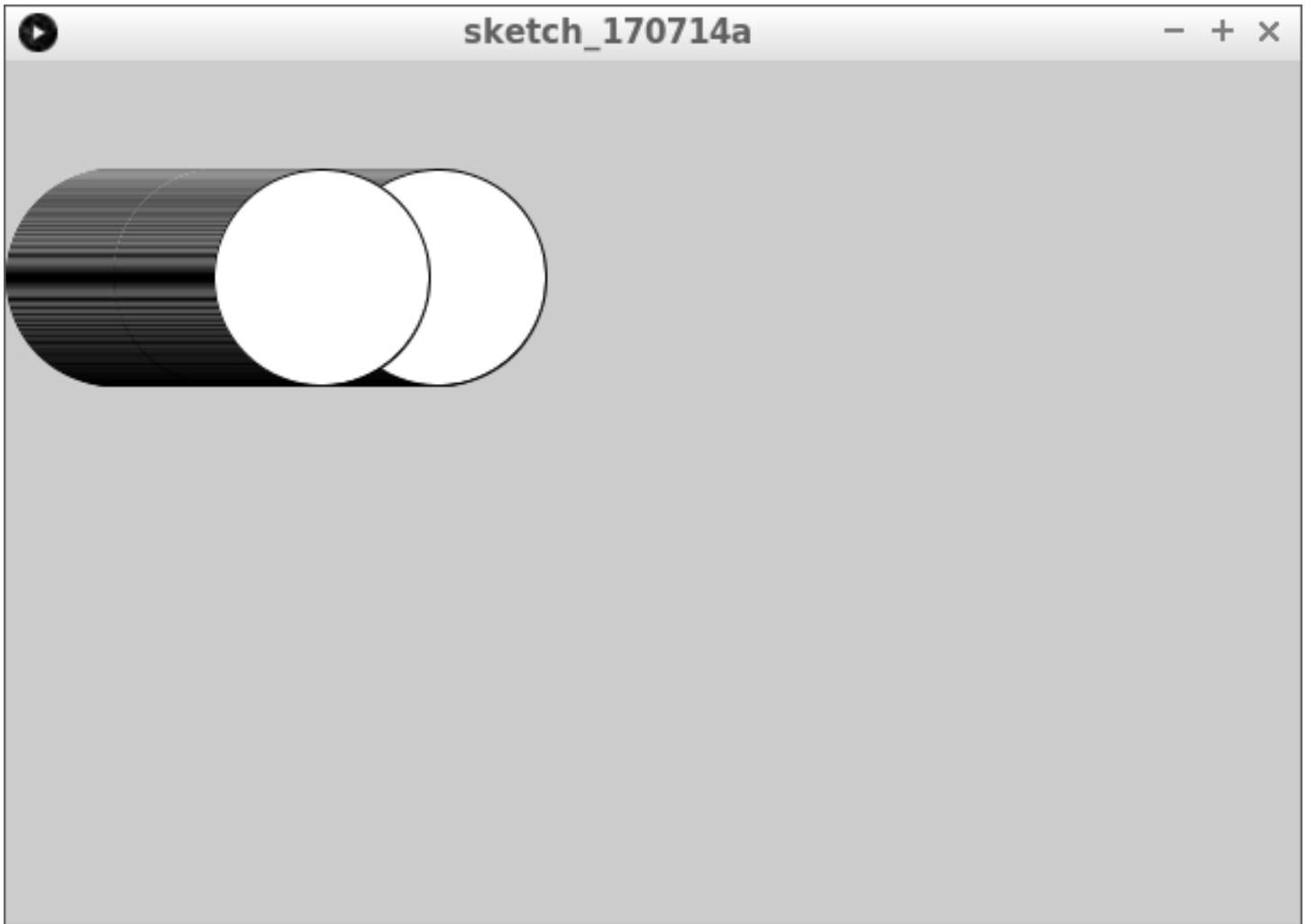


Figure 18: Opdracht 1

Zet de `if` tussen de code van het programma. Zet de `if` aan het eind van `draw`, voor de accolade sluiten `}`.

Bal die eeuwig naar rechts gaat: oplossing 1

De code wordt dan:

```
float x = 50;

void setup()
{
  size(600, 400);
}

void draw()
{
  ellipse(x,100,100,100);
  x = x + 1;
  if (x > 200)
  {
    x = 100;
  }
}
```

Bal die eeuwig naar rechts gaat: opdracht 2

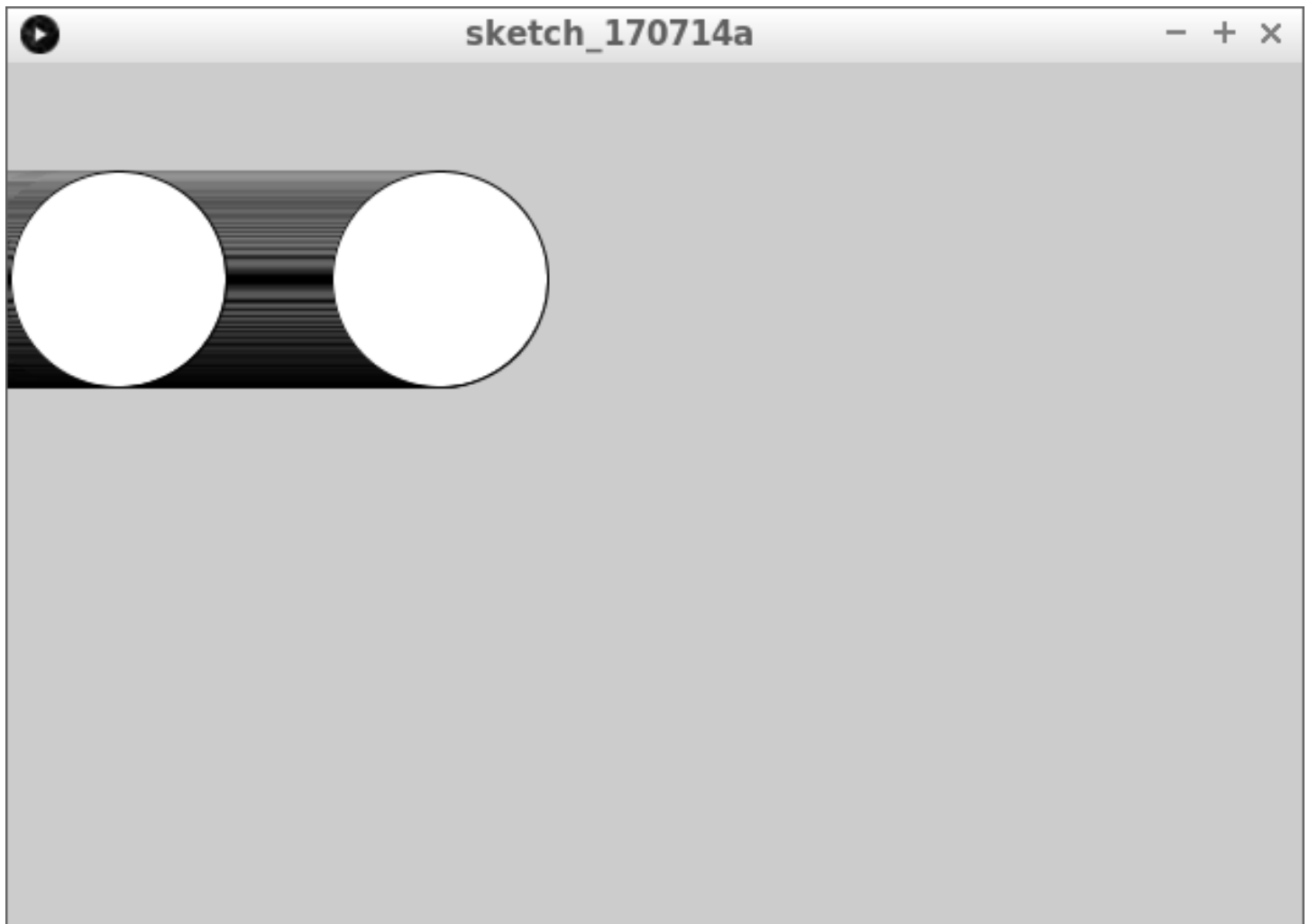


Figure 19: Opdracht 2

Zorg ervoor dat de ovaal helemaal naar de linkerkant van het scherm springt

Bal die eeuwig naar rechts gaat: oplossing 2

- Verander float x = 50 naar float x = 0 of float x = -50: allebei is goed.
- Verander x = 100 naar x = 0 of x = -50: allebei is goed.

```
float x = 50;

void setup()
{
  size(600, 400);
}

void draw()
{
  ellipse(x,100,100,100);
  x = x + 1;
  if (x > 200)
  {
    x = 0;
  }
}
```

Bal die eeuwig naar rechts gaat: opdracht 3

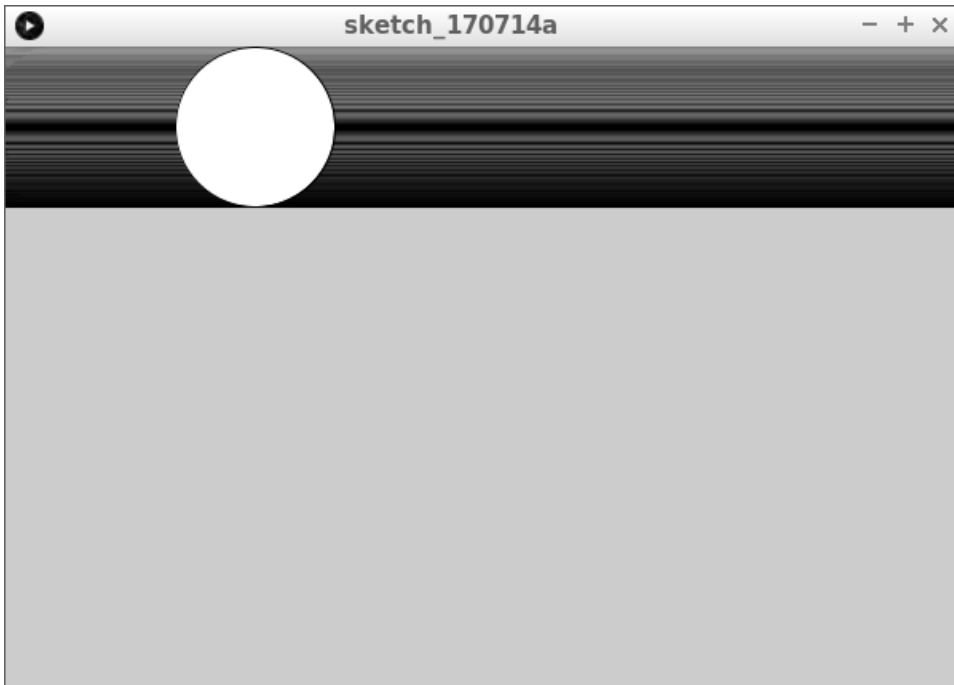


Figure 20: Opdracht 3

Zorg ervoor dat de ovaal helemaal naar rechts beweegt, voordat deze naar de linkerkant van het scherm springt

Bal die eeuwig naar rechts gaat: oplossing 3

Verander if (x > 200) naar if (x > 650).

```
float x = -50;

void setup()
{
  size(600, 400);
}

void draw()
{
  ellipse(x,50,100,100);
  x = x + 1;
  if (x > 650)
  {
    x = 0;
  }
}
```

Bal die eeuwig naar rechts gaat: opdracht 4

Gefopt! Ook al het de les 'Bal die eeuwig naar rechts gaat', we gaan toch de bal ook andere kanten op laten gaan.

We gaan nu een bal programmeren die eeuwig naar links gaat.

Wat je nu moet weten is het `if` statement om te zeggen wanneer de `x` te klein is:

```
if (x < 100)
{
    x = 500;
}
```

Hiermee zeg je: 'Lieve computer, als `x` kleiner (`<`, hier kun je een `k` van maken) is dan honderd, zet dan `x` op vijfhonderd.

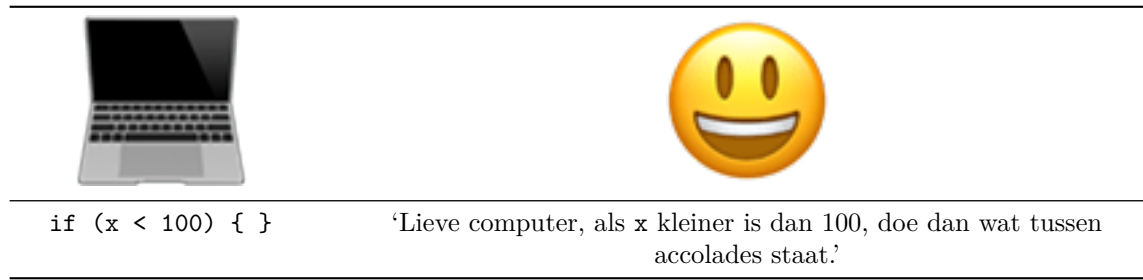


Figure 21: Opdracht 4

Maak een bal die eeuwig naar links gaat:

- De bal begint buiten het beeld
- De bal gaat helemaal het beeld uit
- Als de bal net uit het beeld uit, komt 'ie meteen de andere kant weer binnen

Bal die eeuwig naar rechts gaat: oplossing 4

Dit is een eeuwig naar links gaande bal:

```
float x = 650;

void setup()
{
  size(600, 400);
}

void draw()
{
  ellipse(x, 50, 100, 100);
  x = x - 1;
  if (x < -50)
  {
    x = 650;
  }
}
```



<code>x = x - 1</code>	‘Lieve computer, maak x een lager.’
<code>x -= 1</code>	‘Lieve computer, maak x een lager.’
<code>x--</code>	‘Lieve computer, maak x een lager.’
<code>--x</code>	‘Lieve computer, maak x een lager.’

Bal die eeuwig naar rechts gaat: opdracht 5

We hebben een bal naar rechts en naar links laten bewegen door de `x` coördinaat te veranderen. De bal kan ook naar omlaag en omhoog gaan door de `y` coördinaat te veranderen.

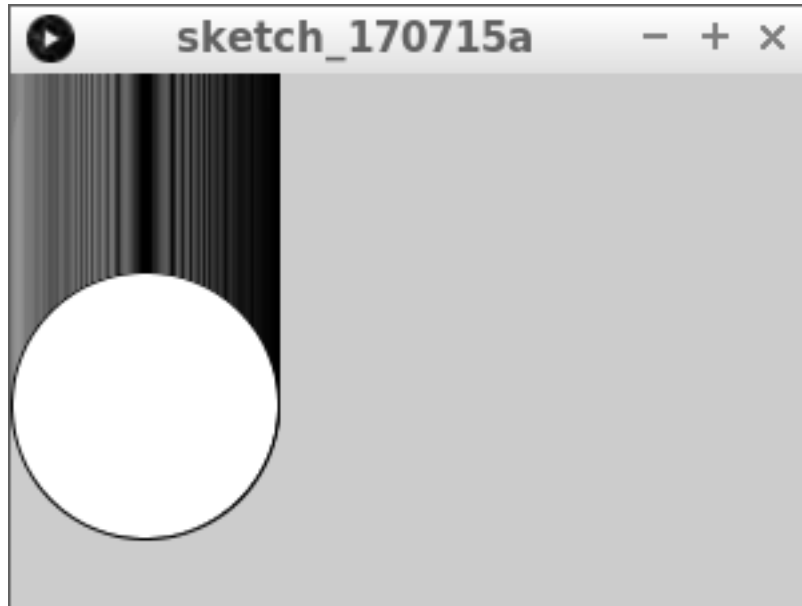


Figure 22: Opdracht 5

Schrijf een programma waarin een bal eeuwig omlaag gaat:

- maak het scherm 300 pixels breed en 200 pixels hoog
- gebruik een variable met naam `y`
- vervang de code `ellipse(x, 50, 100, 100)` door `ellipse(50, y, 100, 100)`
- als de bal omlaag uit het scherm gaat, moet de bal weer bovenin komen

Bal die eeuwig naar rechts gaat: oplossing 5

```
float y = -50;

void setup()
{
  size(300, 200);
}

void draw()
{
  ellipse(50,y,100,100);
  y = y + 1;
  if (y > 250)
  {
    y = -50;
  }
}
```

Bal die eeuwig naar rechts gaat: opdracht 6

Hoppa, nu we een variabele `x` of een `y` hebben gemaakt, gaan we beiden doen!

Als we code samenvoegen, gelden de volgende regels:

- alles wat boven de `setup` functie staat, moet daar blijven
- alles wat binnen de `setup` functie staat, moet binnen de `setup` functie blijven
- alles wat binnen de `draw` functie staat, moet binnen de `draw` functie blijven

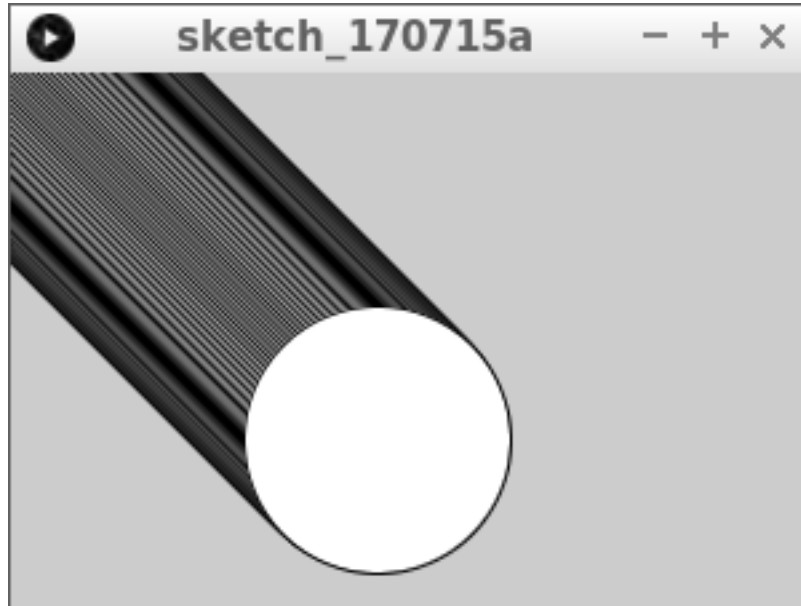


Figure 23: Opdracht 6

- Voeg de code van 'Bal die eeuwig naar rechts gaat' samen met 'Bal die eeuwig omlaag gaat'
- Verander de code zo dat de bal schuin gaat

Bal die eeuwig naar rechts gaat: oplossing 6

```
float x = -50;
float y = -50;

void setup()
{
    size(300, 200);
}

void draw()
{
    ellipse(x,y,100,100);
    x = x + 1;
    y = y + 1;
    if (x > 350)
    {
        x = -50;
    }
    if (y > 250)
    {
        y = -50;
    }
}
```

Bal die eeuwig naar rechts gaat: eindopdracht

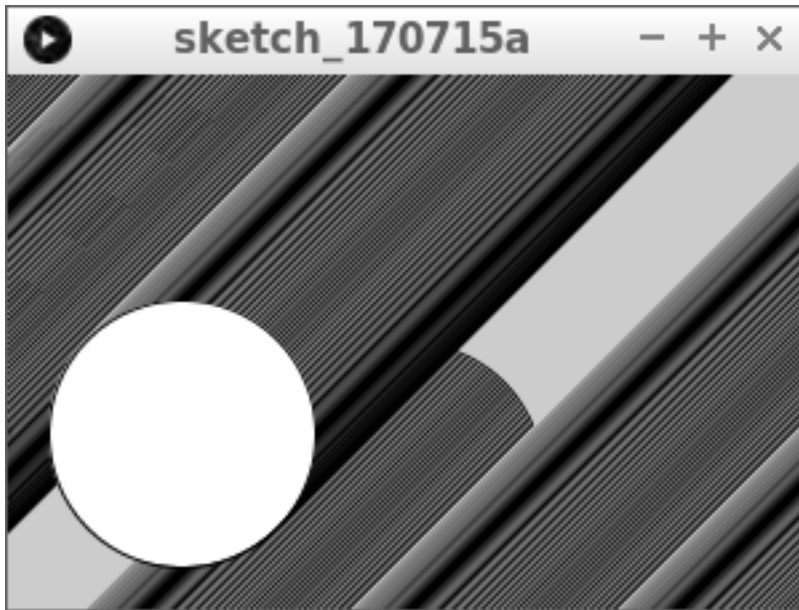


Figure 24: Eindopdracht Bal Eeuwig Naar Rechts

Laat de bal nu eeuwig schuin naar linksonder gaan.

Bal die eeuwig naar rechts gaat: links

- Bal die eeuwig naar rechts gaat: YouTube, mp4

rect en fill

Vierkanten worden veel gebruikt in games.

Hier zie je een van de beroemdste games ooit:

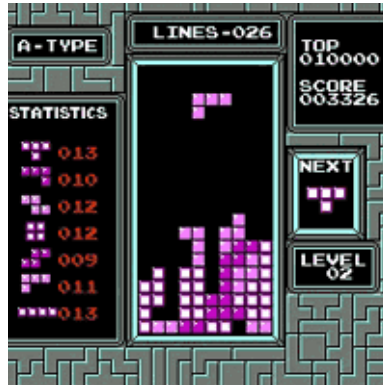


Figure 25: Tetris

Je kunt een vierkant tekenen met vier lijnen, maar de `rect` functie werkt gemakkelijker.

rect en fill: opdracht 1

Run deze code

```
float x = -50;

void setup()
{
  size(300, 200);
}

void draw()
{
  ellipse(x, height / 2, 100, 100);
  x = x + 1;
  if (x > width + 50)
  {
    x = -50;
  }
}
```


rect en fill: oplossing 1

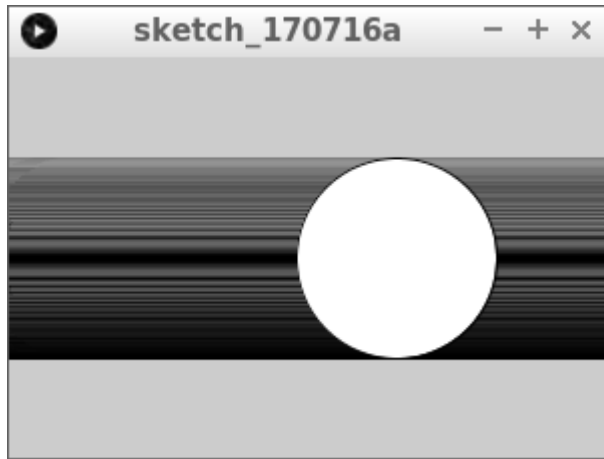


Figure 26: Oplossing 1

rect en fill: opdracht 2

Voeg de volgende extra regel toe, na `ellipse(x, height / 2, 100, 100);`:

```
rect(x, height / 2, 100, 100);
```



```
rect(100, 200, 300, 400)
```

‘Lieve computer, teken een rechthoek met (100, 200) als linkerbovenhoek,
300 pixels breed en 400 pixels hoog is.’

rect en fill: oplossing 2

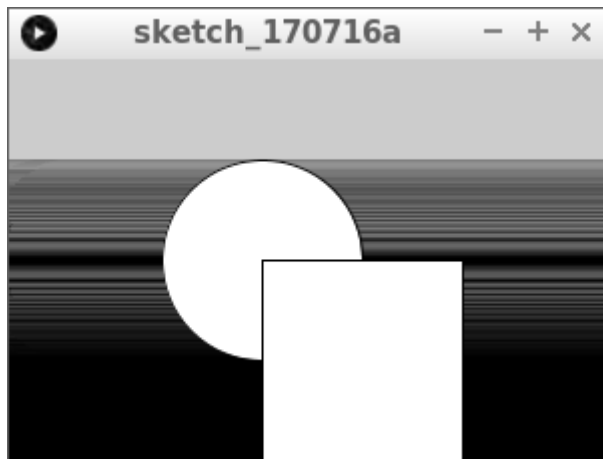


Figure 27: Oplossing 2

```
float x = -50;

void setup()
{
  size(300, 200);
}

void draw()
{
  ellipse(x, height / 2, 100, 100);
  rect(x, height / 2, 100, 100);
  x = x + 1;
  if (x > width + 50)
  {
    x = -50;
  }
}
```

rect en fill: opdracht 3



Figure 28: Opdracht 3

Teken de rechthoek nu over de ovaal. Doe dit door van de `x` en `y` coördinaat 50 af te halen.

rect en fill: oplossing 3

```
float x = -50;

void setup()
{
  size(300, 200);
}

void draw()
{
  ellipse(x, height / 2, 100, 100);
  rect(x - 50, height / 2 - 50, 100, 100);
  x = x + 1;
  if (x > width + 50)
  {
    x = -50;
  }
}
```

rect en fill: opdracht 4

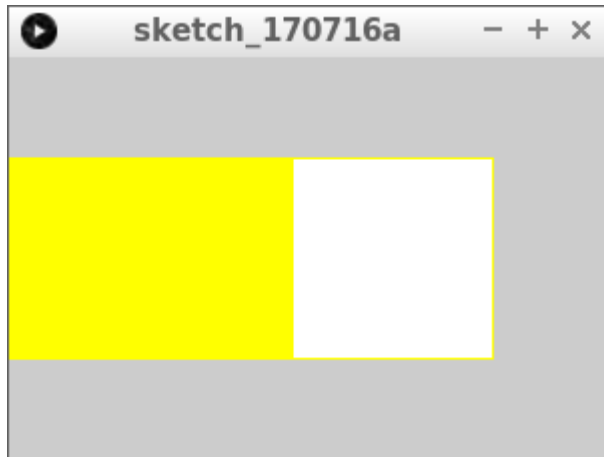


Figure 29: Opdracht 4

Haal de onzichttbare ovaal weg en maak de lijnkleur geel. Hoe je geel maakt, kun je zien in figuur Kleurencirkel.

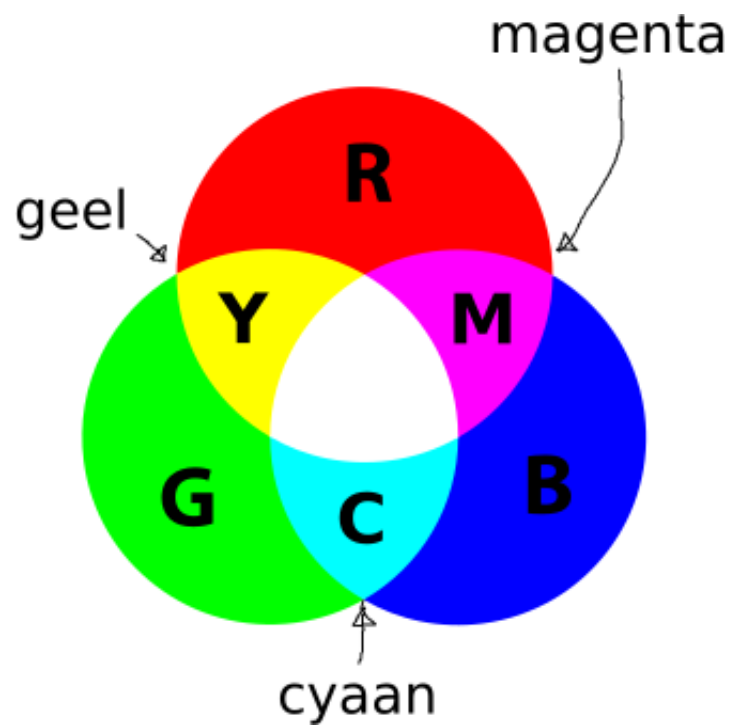


Figure 30: Kleurencirkel

rect en fill: oplossing 4

```
float x = -50;

void setup()
{
  size(300, 200);
}

void draw()
{
  stroke(255, 255, 0);
  rect(x - 50, height / 2 - 50, 100, 100);
  x = x + 1;
  if (x > width + 50)
  {
    x = -50;
  }
}
```

rect en fill: opdracht 5

Voeg de volgende regel toe na `stroke(255, 255, 0);`:

```
fill(x, 0, 255);
```


rect en fill: oplossing 5

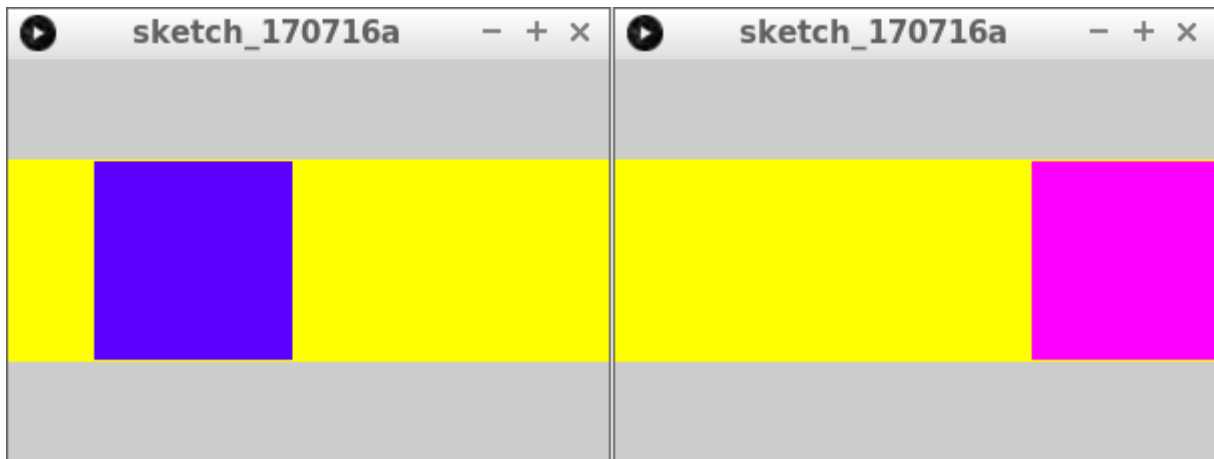


Figure 31: Opdracht 5

```
float x = -50;

void setup()
{
  size(300, 200);
}

void draw()
{
  stroke(255, 255, 0);
  fill(x, 0, 255);
  rect(x - 50, height / 2 - 50, 100, 100);
  x = x + 1;
  if (x > width + 50)
  {
    x = -50;
  }
}
```



`fill(0, 128, 255);`

‘Lieve computer, maak de invulkleur zonder rood, halvol groen en vol blauw.’

rect en fill: opdracht 6

Maak een nieuwe variable aan, **groen** met een beginwaarde van nul. **groen** wordt gebruikt als het tweede getal van **fill** (inplaats van de nul). Elke keer wordt **groen** twee hoger.

rect en fill: oplossing 6

```
float x = -50;
float groen = 0;

void setup()
{
  size(300, 200);
}

void draw()
{
  stroke(255, 255, 0);
  fill(x, groen, 255);
  rect(x - 50, height / 2 - 50, 100, 100);
  x = x + 1;
  groen = groen + 2;
  if (x > width + 50)
  {
    x = -50;
  }
}
```

rect en fill: opdracht 7

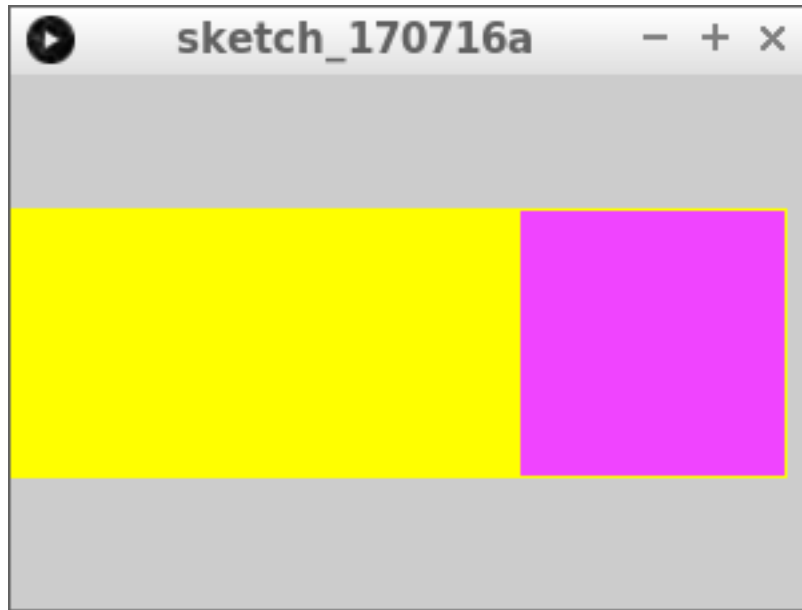


Figure 32: Opdracht 7

De variabele **groen** mag niet hoger worden dan 255. Maak een **if** erbij, dat als **groen** meer is dan 255, dat groen dan nul wordt

rect en fill: oplossing 7

```
float x = -50;
float groen = 0;

void setup()
{
  size(300, 200);
}

void draw()
{
  stroke(255, 255, 0);
  fill(x, groen, 255);
  rect(x - 50, height / 2 - 50, 100, 100);
  x = x + 1;
  groen = groen + 2;
  if (x > width + 50)
  {
    x = -50;
  }
  if (groen > 255)
  {
    groen = 0;
  }
}
```

rect en fill: opdracht 8

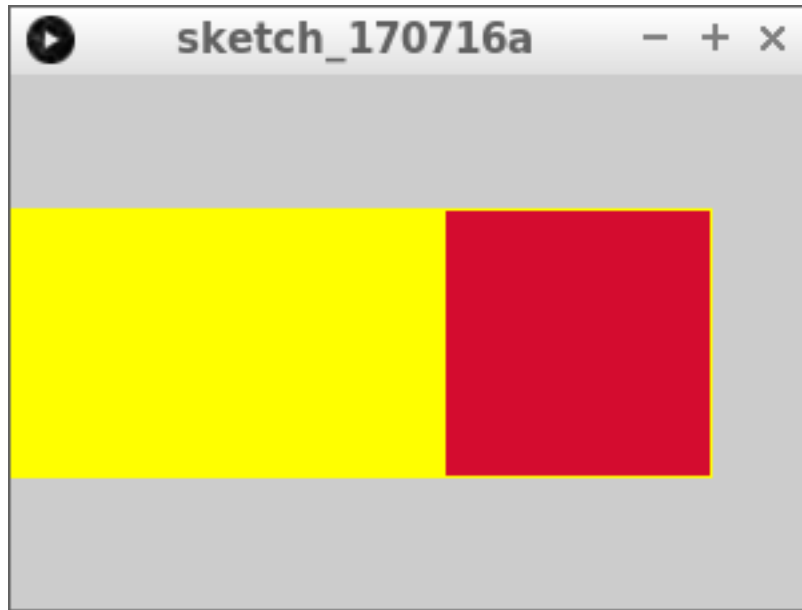


Figure 33: Opdracht 8

Maak de blauwheid van de invulkleur een willekeurig getal van nul tot 256.

rect en fill: oplossing 8

```
float x = -50;
float groen = 0;

void setup()
{
  size(300, 200);
}

void draw()
{
  stroke(255, 255, 0);
  fill(x, groen, random(256));
  rect(x - 50, height / 2 - 50, 100, 100);
  x = x + 1;
  groen = groen + 2;
  if (x > width + 50)
  {
    x = -50;
  }
  if (groen > 255)
  {
    groen = 0;
  }
}
```

rect en fill: eindopdracht

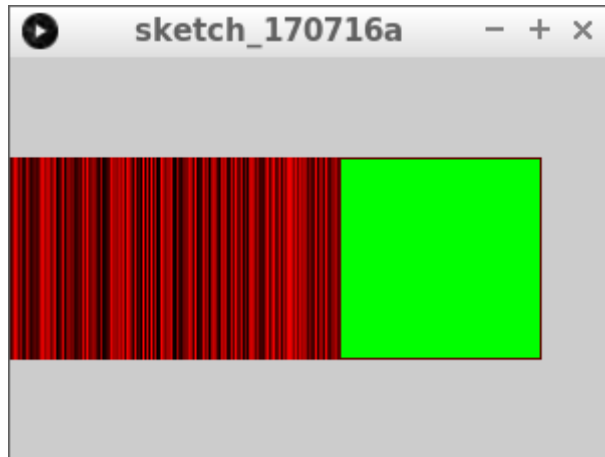


Figure 34: Eindopdracht **rect** en **fill**

Maak de lijnkleur willekeurig in roodheid (dus van zwart tot rood, groen en blauw zijn uit). Laat de invulleur gaan van zwart naar groen, waarbij groen steeds 3 hoger wordt.