



Figure 1: Boek 2

Contents

| | |
|---------------------------------|----|
| Voorwoord | 1 |
| line en stroke | 2 |
| Bal die eeuwig naar rechts gaat | 10 |
| rect | 21 |
| Bal die horizontaal stuitert | 29 |
| ellipse en background | 38 |
| Bal die schuin stuitert | 44 |
| text | 49 |

Voorwoord



Figure 1: Het logo van De Jonge Onderzoekers



Figure 2: Het logo van Codestarter

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 3: De licentie van dit boek

(C) Dojo Groningen 2016-2017

Het is nog een beetje een slordig boek. Zo staat bijvoorbeeld het plaatje dat eigenlijk op de kaft moet staan op pagina twee. 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 4: Moria, een van de allereerste games met kleur

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

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

Oplossing 1

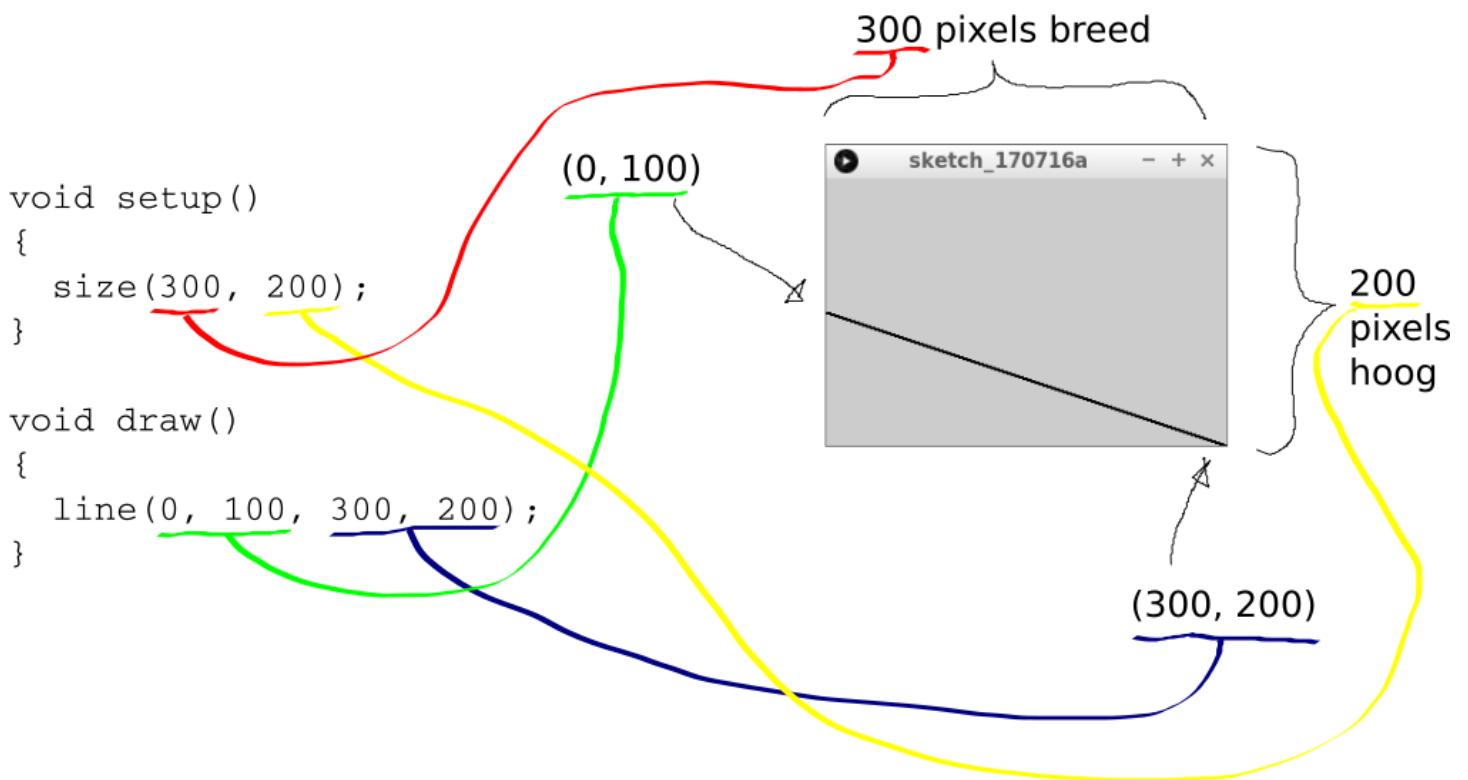


Figure 5: Oplossing 1

Opdracht 2

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

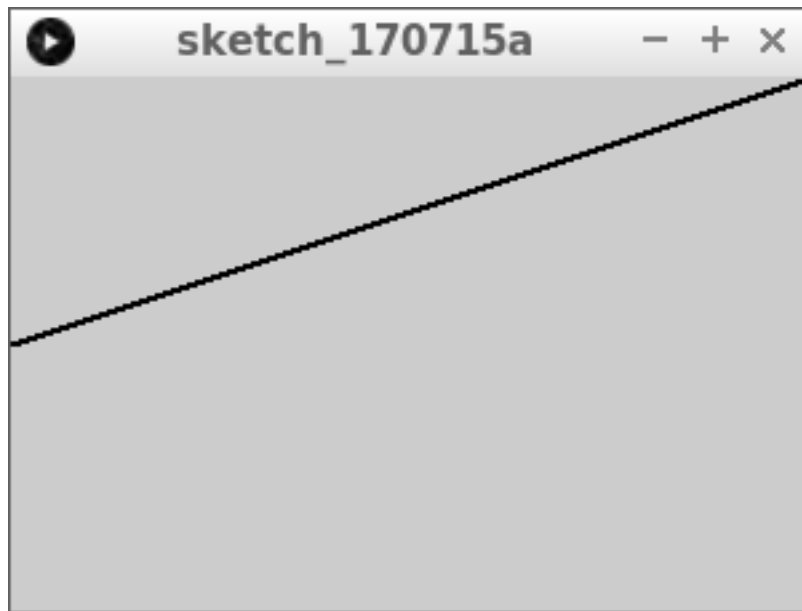


Figure 6: Opdracht 2

Oplossing 2

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

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

Opdracht 3

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

Oplossing 3

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

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

Opdracht 4

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

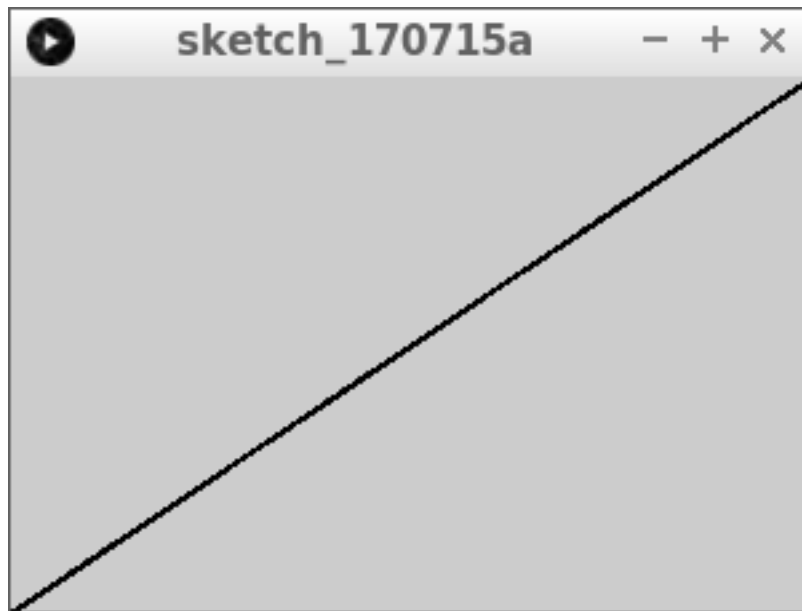


Figure 7: Opdracht 3

Oplossing 4

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

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

Opdracht 5

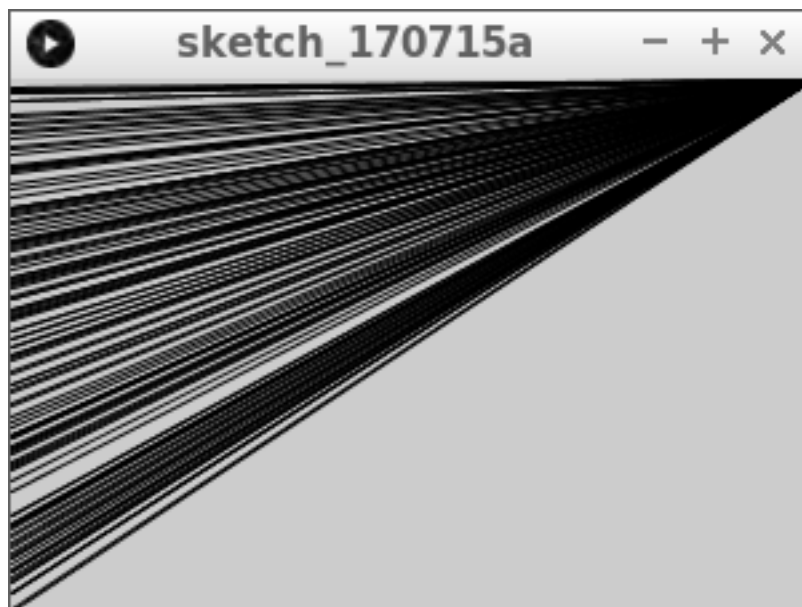


Figure 8: Opdracht 5

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);
}
```

Opdracht 6



Figure 9: Opdracht 6

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

Oplossing 6

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

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

Opdracht 7

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

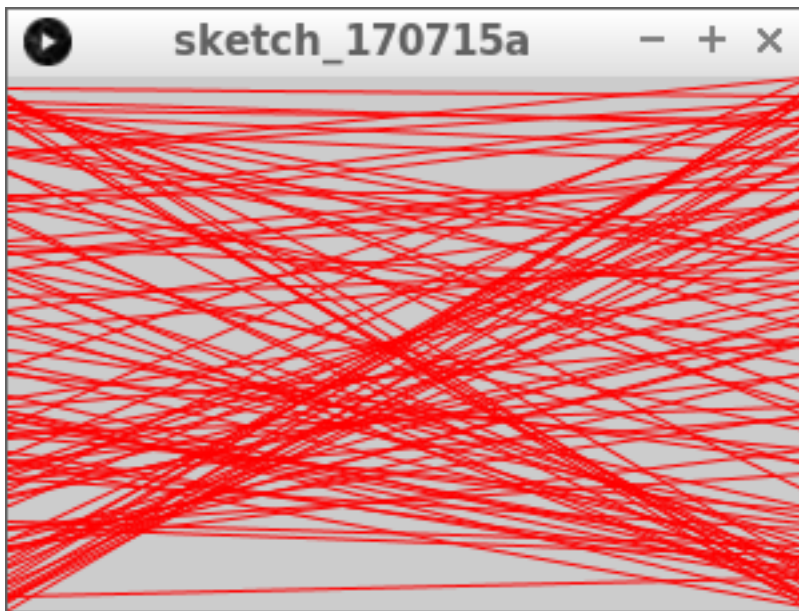


Figure 10: Opdracht 7

Oplossing

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

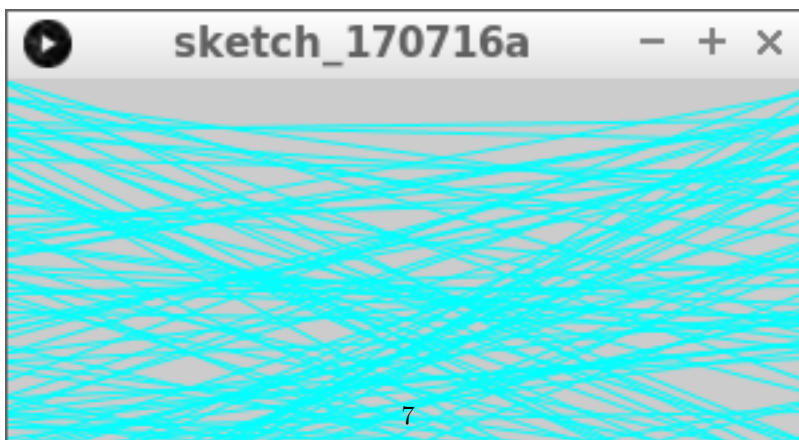
void draw()
{
  stroke(255, 0, 0);
  line(0, random(height), width, random(height));
}
```



```
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.’

Opdracht 8



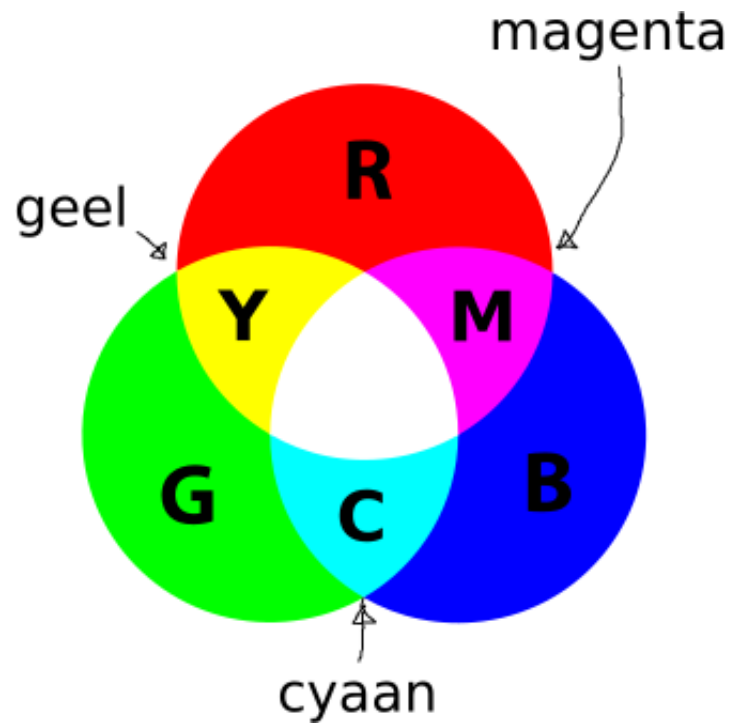


Figure 12: Kleurencirkel

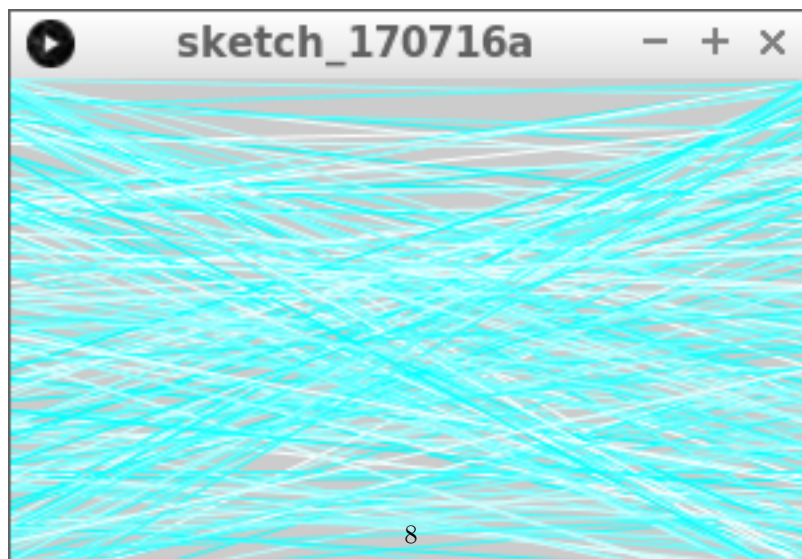
```
void draw()
{
  stroke(0, 255, 255);
  line(0, random(height), width, random(height));
}
```



```
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.’

Opdracht 9



Eindopdracht

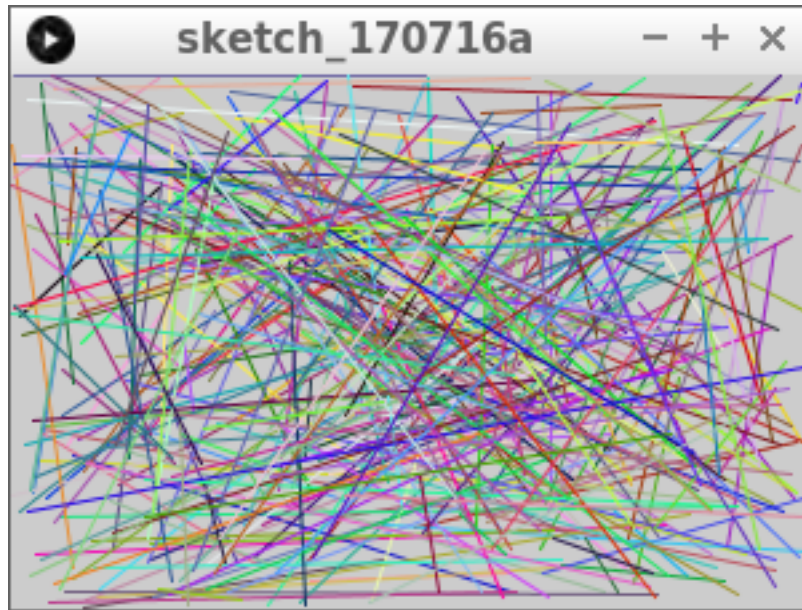


Figure 14: Eindopdracht `line` en `stroke`

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

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.

Een `if`-statement

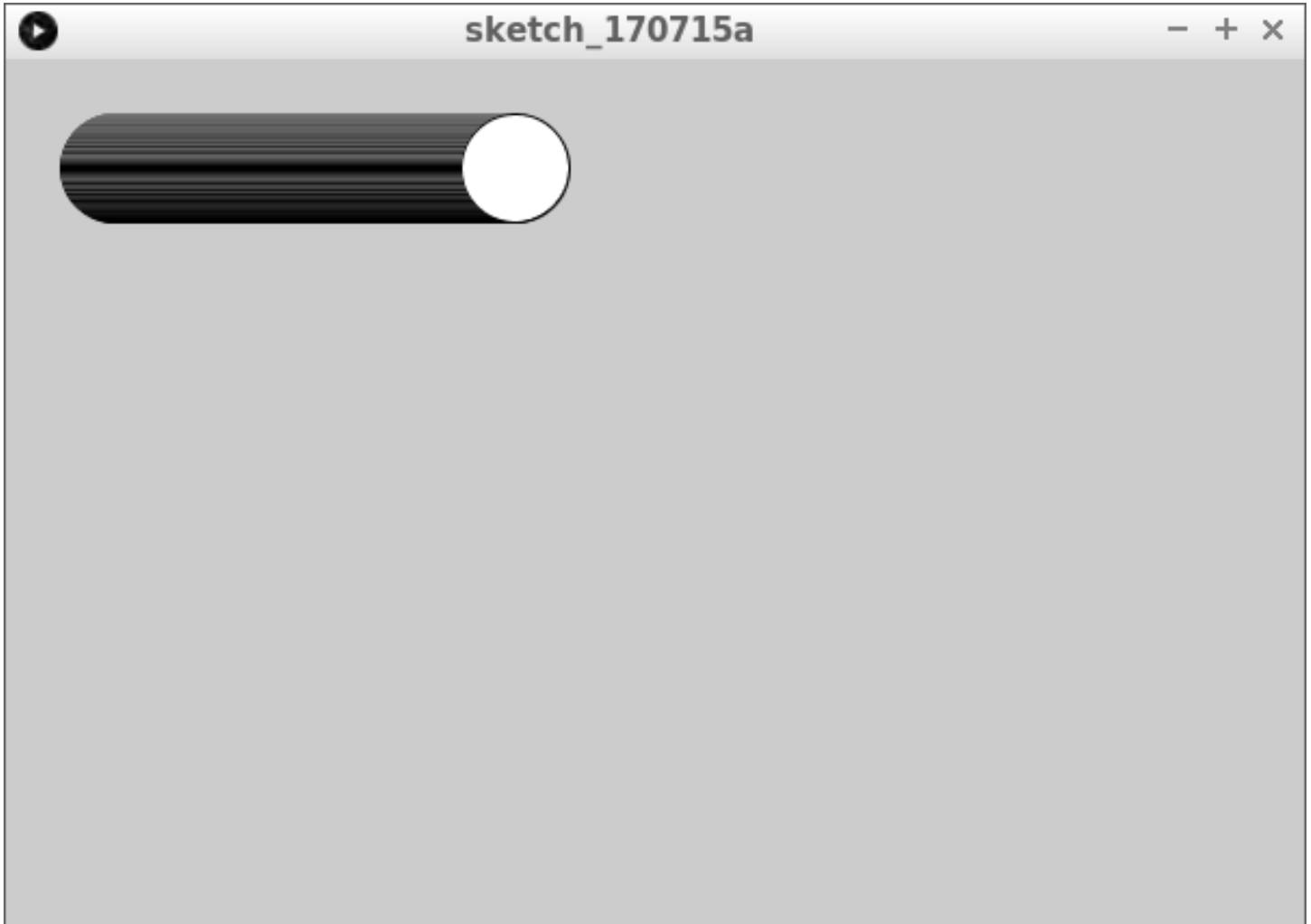


Figure 15: Verkleurende bal naar rechts

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’.

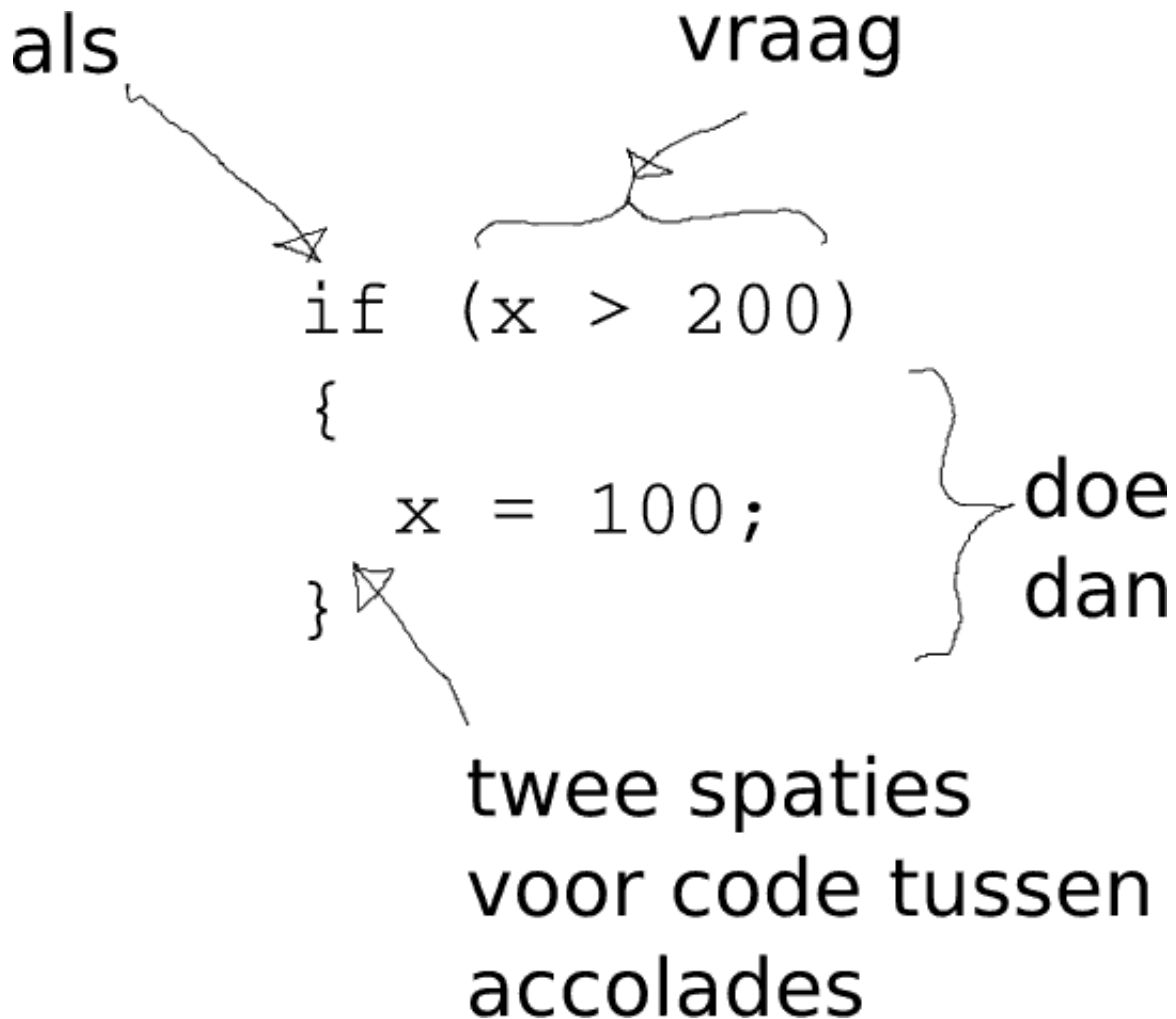




Figure 16: Een if

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

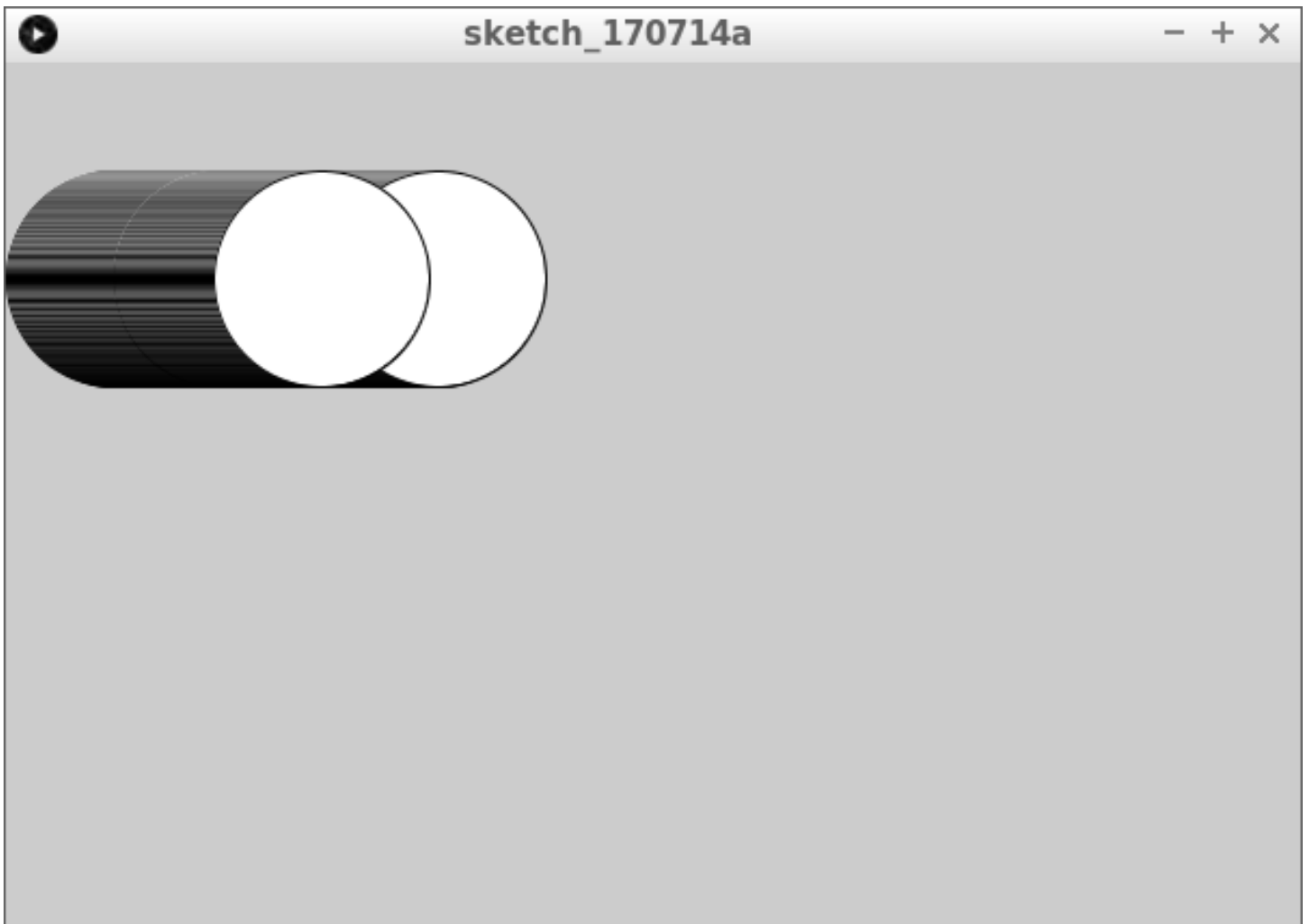


Figure 17: Opdracht 1

Opdracht 1

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

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;
    }
}
```

Opdracht 2

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

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;
    }
}
```

Opdracht 3

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

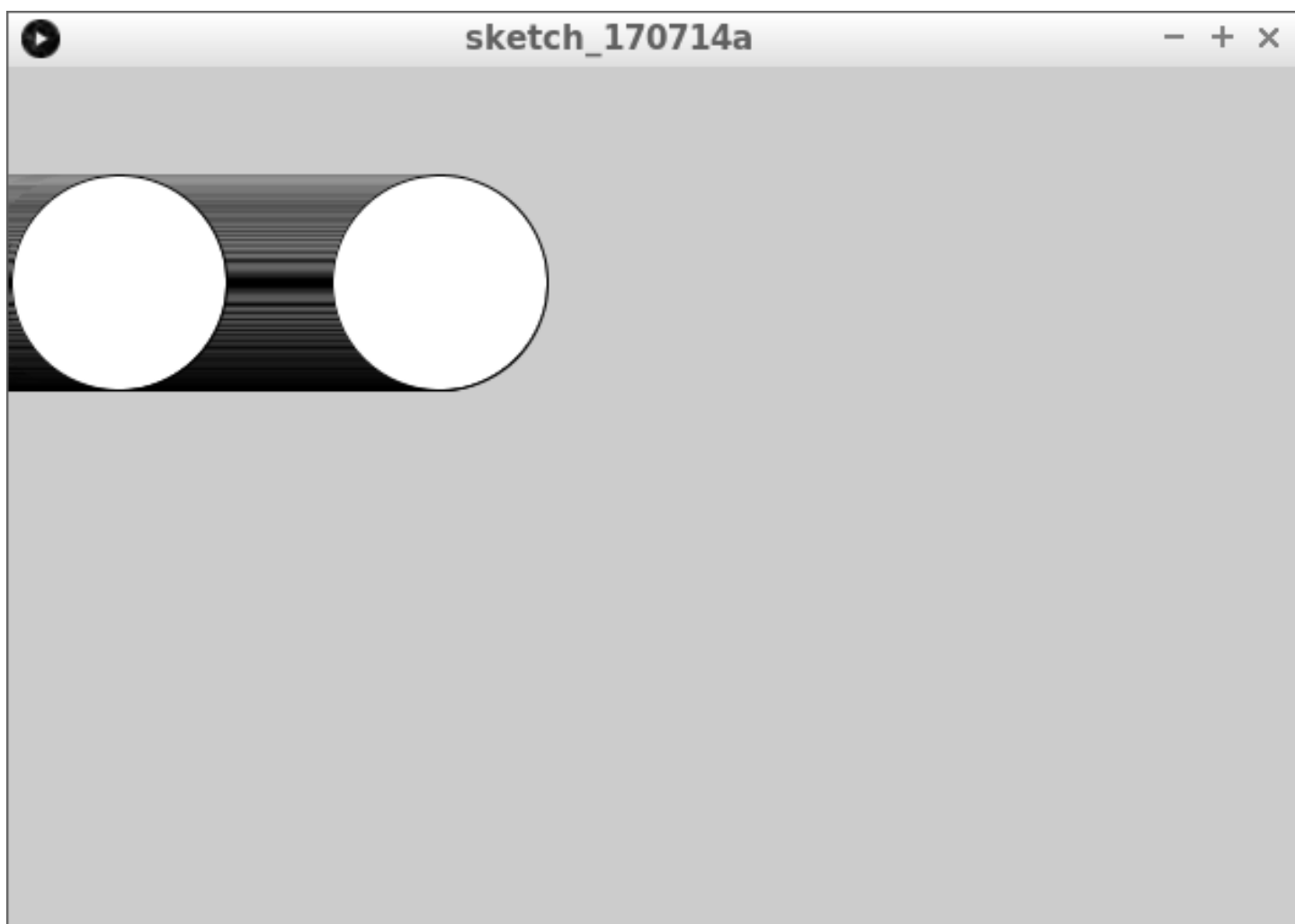


Figure 18: Opdracht 2

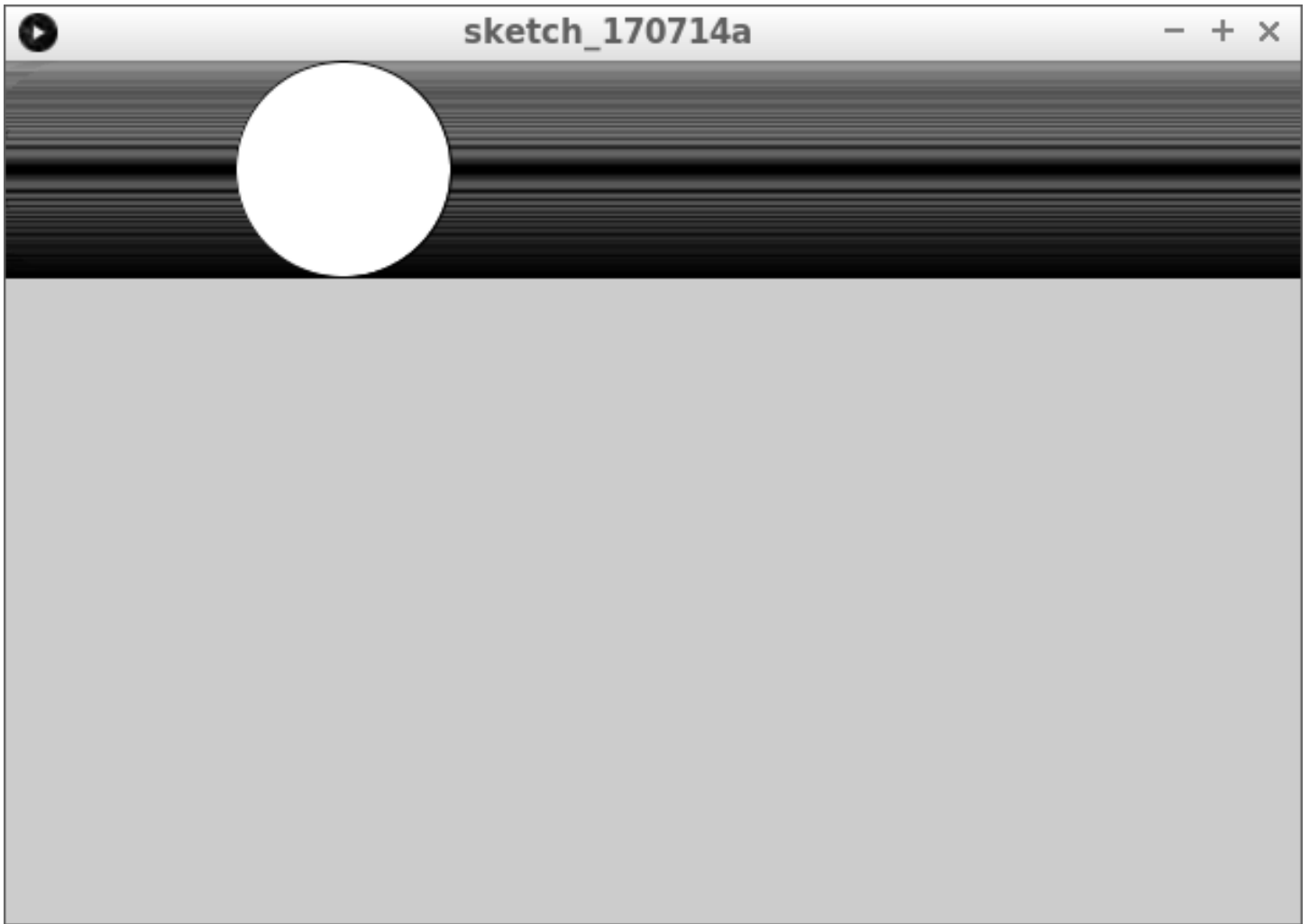


Figure 19: Opdracht 3

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 links gaat

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.



```
if (x < 100) { }
```

‘Lieve computer, als `x` kleiner is dan 100, doe dan wat tussen accolades staat.’

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

Antwoord 4

Dit is een eeuwig naar links gaande bal:

```
float x = 650;

void setup()
```





Figure 20: Opdracht 4

```

{
  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 omlaag gaat

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.

Opdracht 5

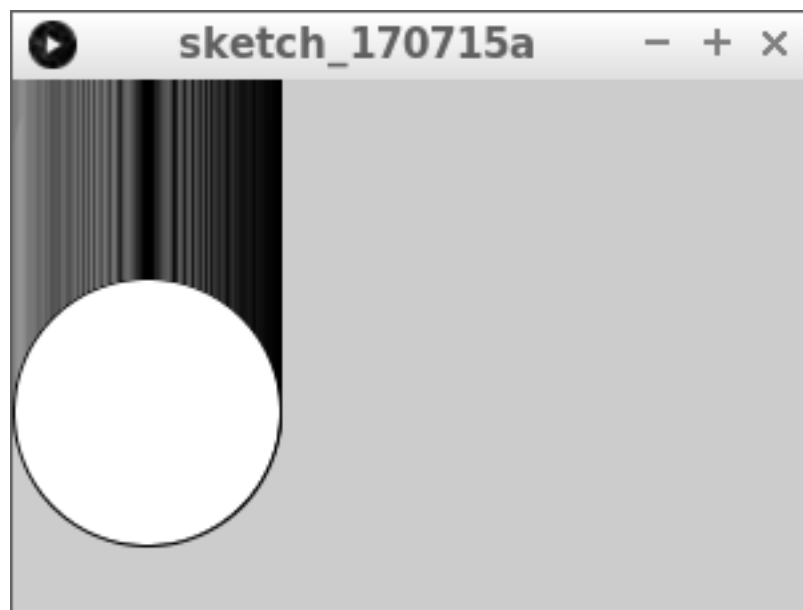


Figure 21: Opdracht 5

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

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 schuin gaat

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

Opdracht 6

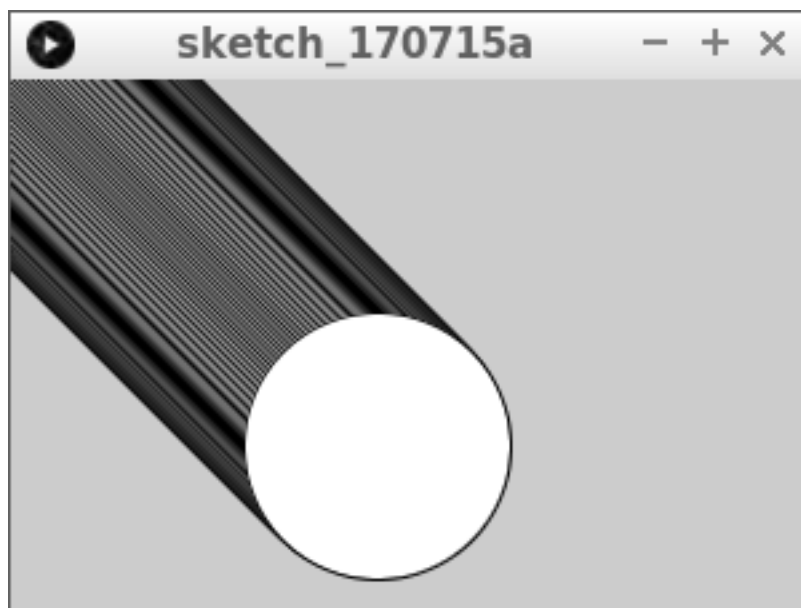


Figure 22: Opdracht 6

- 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

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;
  }
}
```

Eindopdracht

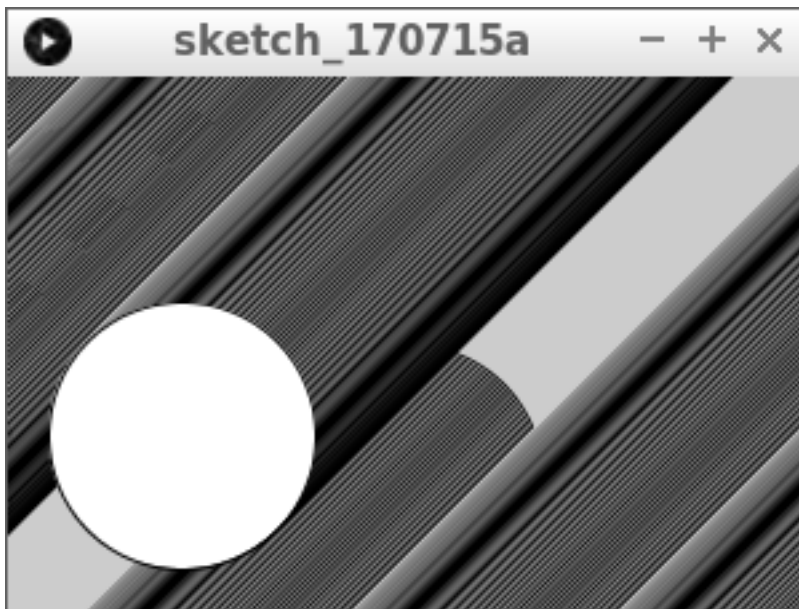


Figure 23: Eindopdracht Bal Eeuwig Naar Rechts

Laat de bal nu eeuwig schuin naar linksonder gaan.

rect

Vierkanten worden veel gebruikt in games.

Hier zie je een van de beroemdste games ooit:

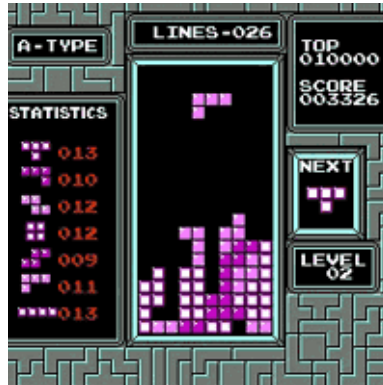


Figure 24: Tetris

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

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;
  }
}
```

Oplossing 1

Opdracht 2

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

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

Oplossing 2

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

```
void setup()
{
```

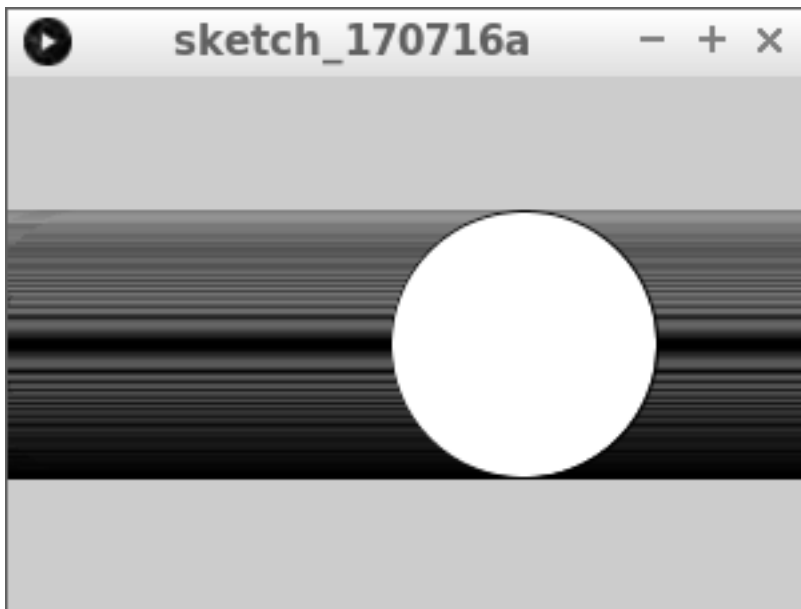


Figure 25: Oplossing 1

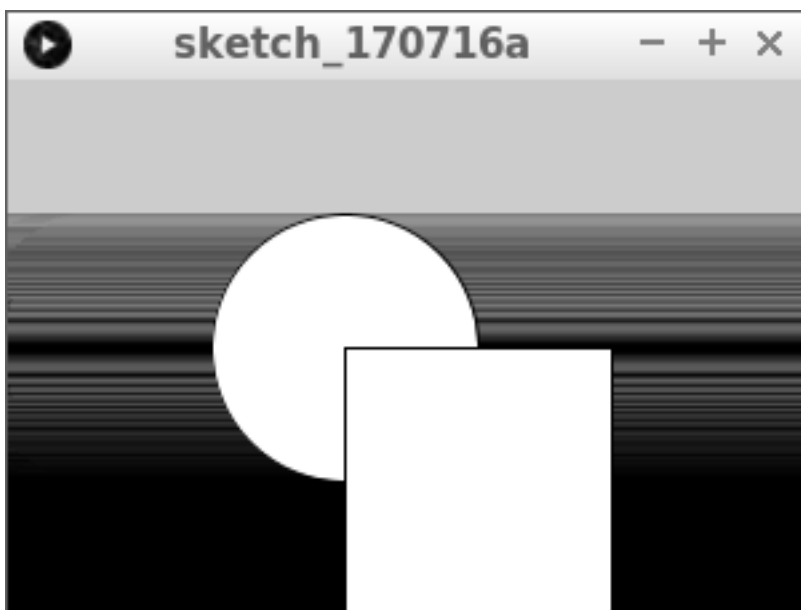


Figure 26: Oplossing 2

```

    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(100, 200, 300, 400)`

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

Opdracht 3



Figure 27: Opdracht 3

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

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;
  }
}

```

Opdracht 4

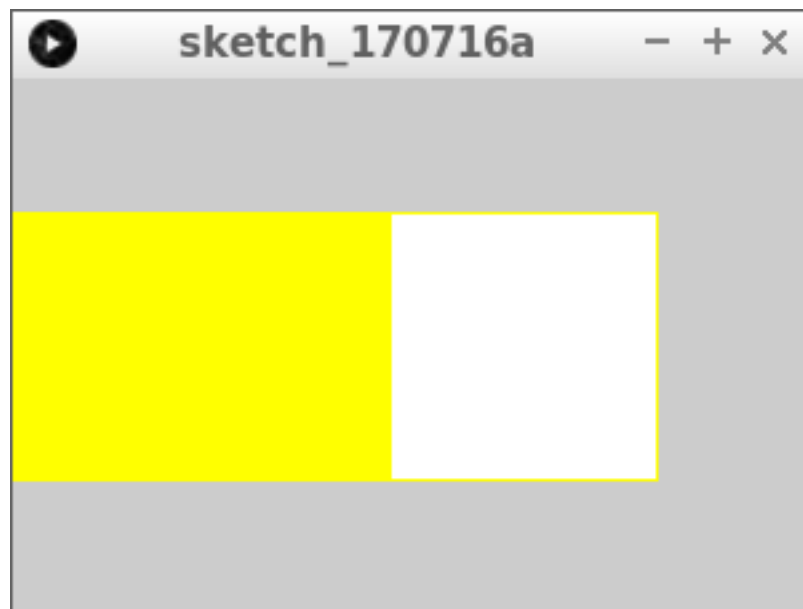


Figure 28: Opdracht 4

Haal de onzichtbare ovaal weg en maak de lijnkleur geel. Hoe je geel maakt, kun je zien in figuur [Kleurencirkel](#).

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;
  }
}

```

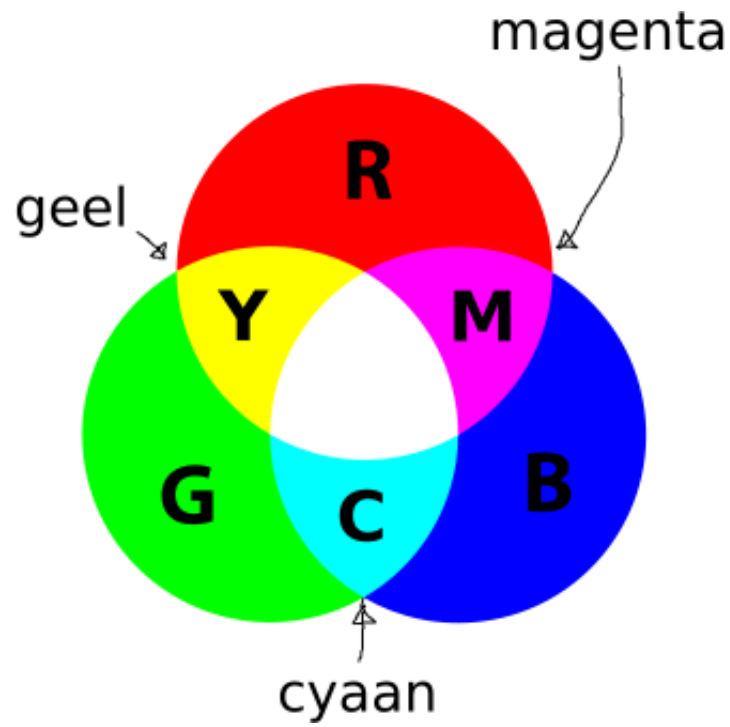


Figure 29: Kleurencirkel

Opdracht 5

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

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

Oplossing

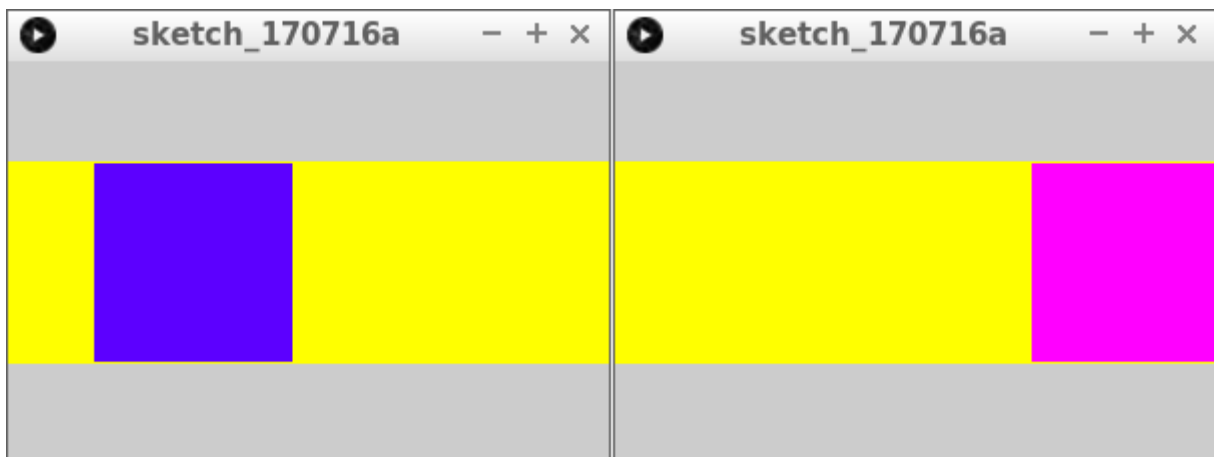


Figure 30: Opdracht 5

```
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.’

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.

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;
  }
}

```

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

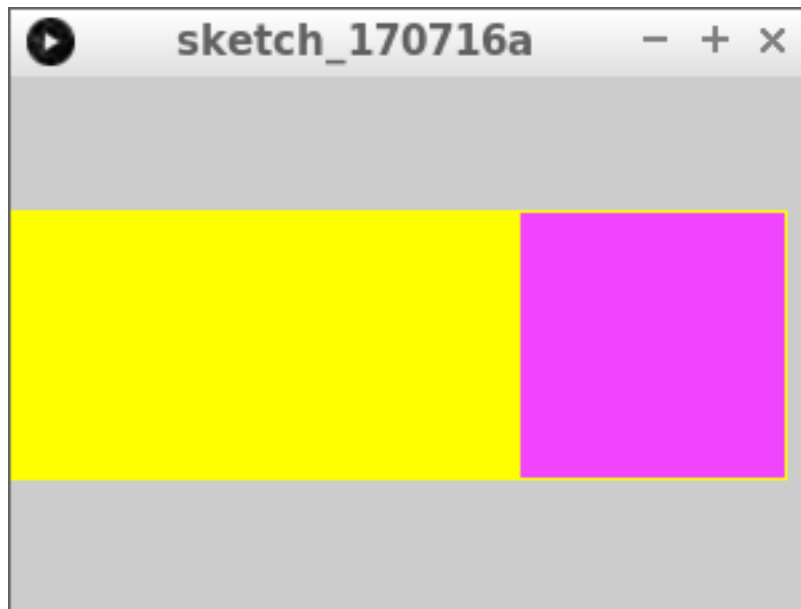


Figure 31: Opdracht 7

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;
  }
}
```

Opdracht 8

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

Oplossing 8

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

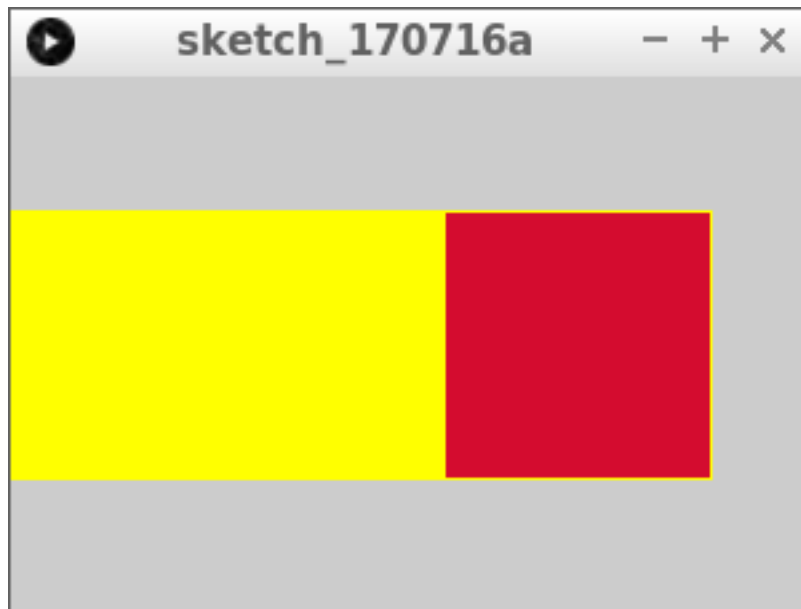


Figure 32: Opdracht 8

```
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;
  }
}
```

Eindopdracht

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



Figure 33: Eindopdracht `rect` en `fill`

Bal die horizontaal stuitert

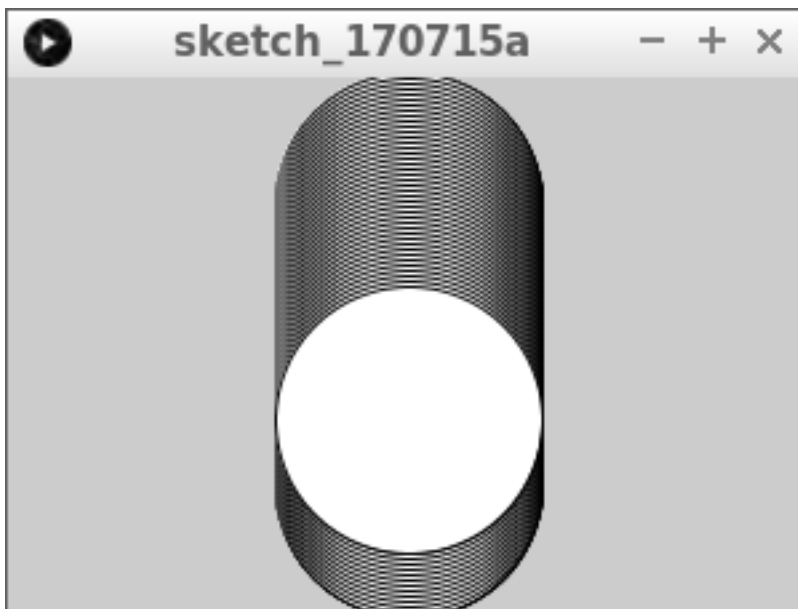


Figure 34: Eindopdracht Bal Die Horizontaal Stuitert

In deze les gaan we een bal horizontaal laten stuiteren.

Een bal met een snelheid

```
float x = 300;
float snelheid = 2;

void setup()
{
  size(600, 100);
}

void draw()
{
  ellipse(x, 50, 100, 100);
  x = x + snelheid;
}
```

```
if (x > 650)
{
    x = -50;
}
```

Opdracht 1

Type de code van Een bal met een snelheid over. Zet de waarde van snelheid op 1. Wat zie je?

Antwoord

Je ziet dat de bal rusig naar rechts beweegt. Als de bal rechts het beeld uit gaat, komt deze vanaf links weer in het beeldscherm.

```
float x = 300;
float snelheid = 1;

void setup()
{
    size(600, 100);
}

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



Figure 35: Oplossing 1

Opdracht 2

Zet de waarde van snelheid op 0. Wat zie je?

Antwoord 2

De bal staat nu stil

```

float x = 300;
float snelheid = 0;

void setup()
{
  size(600, 100);
}

void draw()
{
  ellipse(x,50,100,100);
  x = x + snelheid;
  if (x > 650)
  {
    x = -50;
  }
}

```

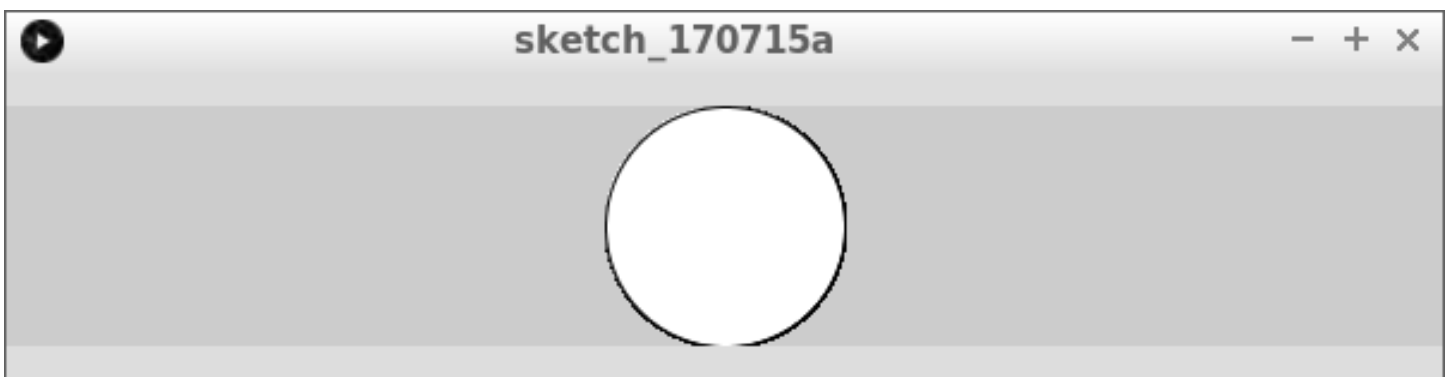


Figure 36: Oplossing 2

Opdracht 3

Zet de waarde van `snelheid` op `-1`. Wat zie je?



Het getal `-1` spreek je uit als 'min een'. Het is het getal een lager dan nul.

Antwoord 3

De bal gaat nu naar links en komt niet meer in beeld.

```

float x = 300;
float snelheid = -1;

void setup()
{
  size(600, 100);
}

void draw()

```



```

{
  ellipse(x,50,100,100);
  x = x + snelheid;
  if (x > 650)
  {
    x = -50;
  }
}

```



Figure 37: Oplossing 3

Opdracht 4



Figure 38: Opdracht 4

Zorg dat als de bal links uit beeld gaat, deze weer rechts verschijnt. Doe dit door een tweede if te maken

Oplossing 4

```

float x = 300;
float snelheid = -1;

void setup()
{
  size(600, 100);
}

void draw()
{
  ellipse(x,50,100,100);
  x = x + snelheid;
}

```

```
if (x > 650)
{
    x = -50;
}
if (x < -50)
{
    x = 650;
}
}
```

Opdracht 5

Denk na: wat moet er met de **snelheid** gebeuren om de bal te laten stuiten (van links naar rechts)?

Als je durft: probeer dit zonder verder te lezen!

Oplossing 5

Als tegen de linkerkant komt, moet de bal de bal naar rechts gaan. Als tegen de rechterkant komt, moet de bal de bal naar links gaan.





if (



x <



x >

Opdracht 6



Figure 39: Opdracht 6

Laat de bal nu van links naar rechts stuiten

Oplossing 6

```
float x = 300;  
float snelheid = 1;  
  
void setup()  
{
```

```

    size(600, 100);
}

void draw()
{
    ellipse(x, 50, 100, 100);
    x = x + snelheid;
    if (x > 550)
    {
        snelheid = -1;
    }
    if (x < 50)
    {
        snelheid = 1;
    }
}

```

Opdracht 7

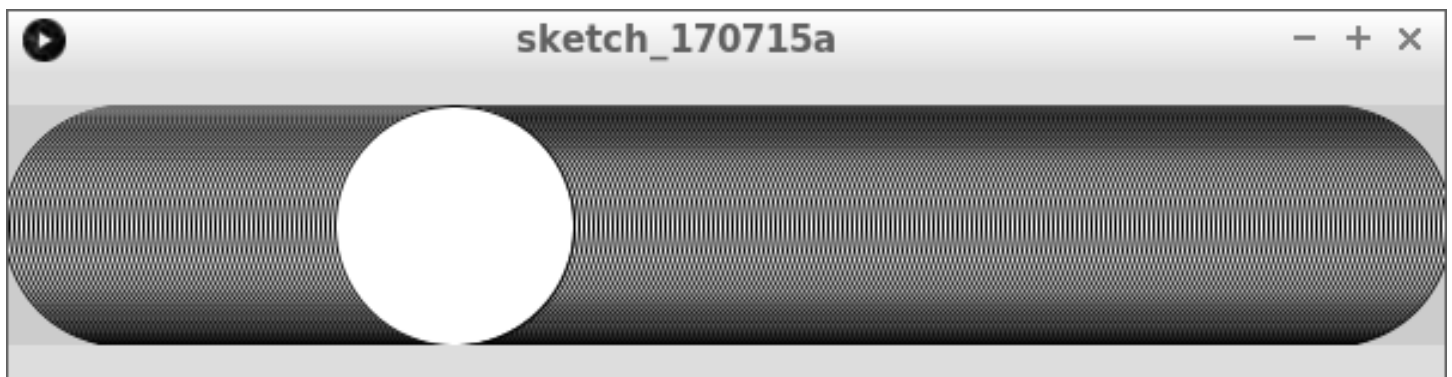


Figure 40: Opdracht 7

Laat de bal nu met een snelheid van 2 pixels per keer bewegen.

Oplossing 7

```

float x = 300;
float snelheid = 2;

void setup()
{
    size(600, 100);
}

void draw()
{
    ellipse(x, 50, 100, 100);
    x = x + snelheid;
    if (x > 550)
    {
        snelheid = -2;
    }
    if (x < 50)
    {
        snelheid = 2;
    }
}

```

```
}  
}
```

Opdracht 8

Verander in de code:

| van | naar |
|-----------------------------|------------------------------------|
| <code>snelheid = 2;</code> | <code>snelheid = -snelheid;</code> |
| <code>snelheid = -2;</code> | <code>snelheid = -snelheid;</code> |



| | |
|------------------------------------|----------------------------------------------------------------|
| <code>snelheid = -snelheid;</code> | ‘Lieve computer, verander het (min of plus)teken van snelheid’ |
|------------------------------------|----------------------------------------------------------------|



Het getal -1 heeft een minteken. Het getal 1 (ook goed: +1) heeft een plusteken.

Oplossing 8

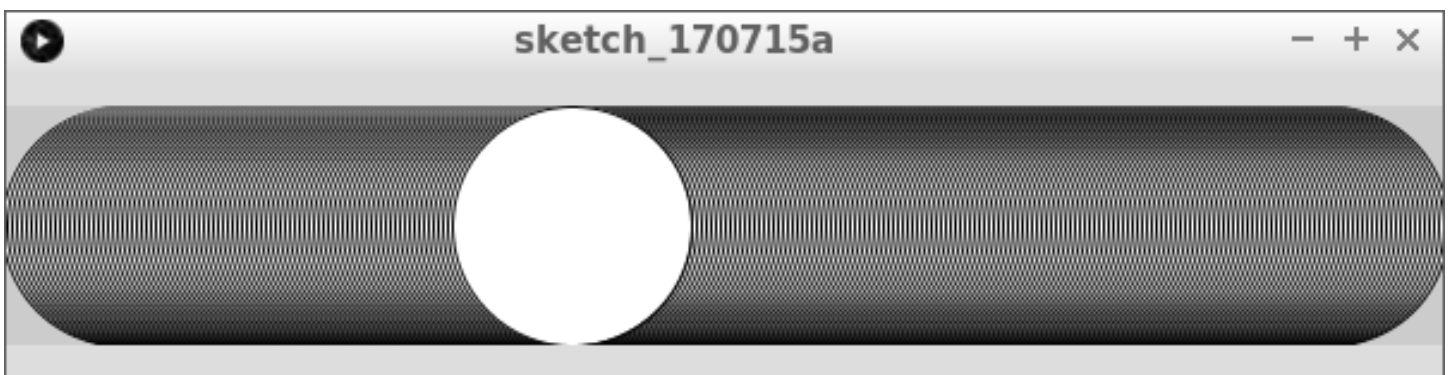


Figure 41: Opdracht 8

```
float x = 300;  
float snelheid = 2;  
  
void setup()  
{  
  size(600, 100);  
}  
  
void draw()  
{  
  ellipse(x, 50, 100, 100);  
}
```

```
x = x + snelheid;  
if (x > 550)  
{  
    snelheid = -snelheid;  
}  
if (x < 50)  
{  
    snelheid = -snelheid;  
}  
}
```

Eindopdracht

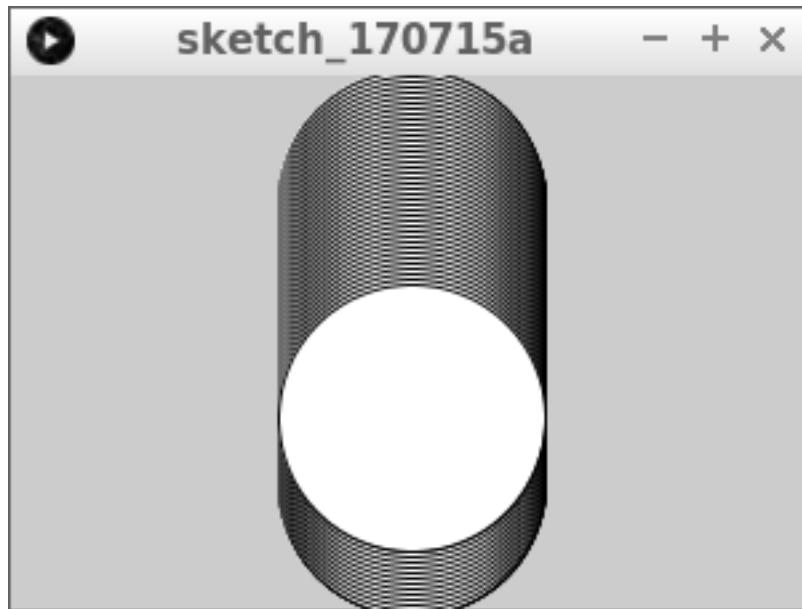


Figure 42: Eindopdracht Bal Die Horizontaal Stuitert

Laat de bal nu omhoog en omlaag stuiten.

ellipse en background

In deze les gaan we leren hoe je ovalen tekent.

Hier zie je een beroemde game, Bubble Bobble, dat veel met cirkels werkt:



Figure 43: Bubble Bobble

Opdracht 1

Run deze code

```
float x = -50;
float snelheid = 1;

void setup()
{
  size(400, 100);
}

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

Oplossing 1

Opdracht 2

Verander het derde en vierde getal van `ellipse` (de 100s), zodat de ovaal twee keer zo hoog wordt.

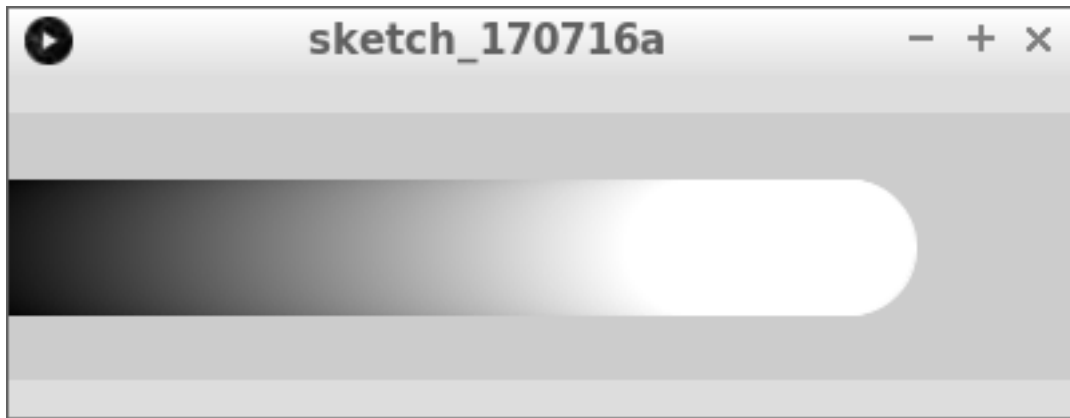


Figure 44: Oplossing 1



Figure 45: Opdracht 2

Oplossing 2

```
float x = -50;
float snelheid = 1;

void setup()
{
  size(400, 100);
}

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

Opdracht 3



Figure 46: Opdracht 3

Voeg als eerste regel van `draw` toe:

```
background(255, 0, 0);
```

Oplossing 3

```
float x = -50;
float snelheid = 1;

void setup()
{
  size(400, 100);
}

void draw()
{
  background(255, 0, 0);
```

```

stroke(x, x, x);
fill(x, x, x);
ellipse(x, height / 2, 50, 100);
x = x + snelheid;
if (x > width + 50)
{
    x = -50;
}
}

```

Opdracht 4

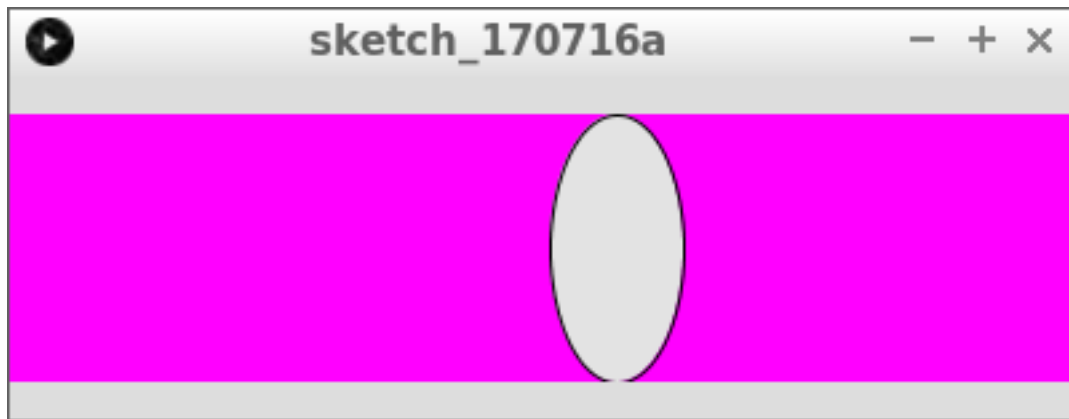


Figure 47: Oplossing 4

Maak de achtergrond nu magenta. Maak de rand van de cirkel zwart

Oplossing 4

```

float x = -50;
float snelheid = 1;

void setup()
{
    size(400, 100);
}

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

```

Opdracht 5

Zet de regel met `background` nu na `size` (binnen `setup`).

Oplossing 5

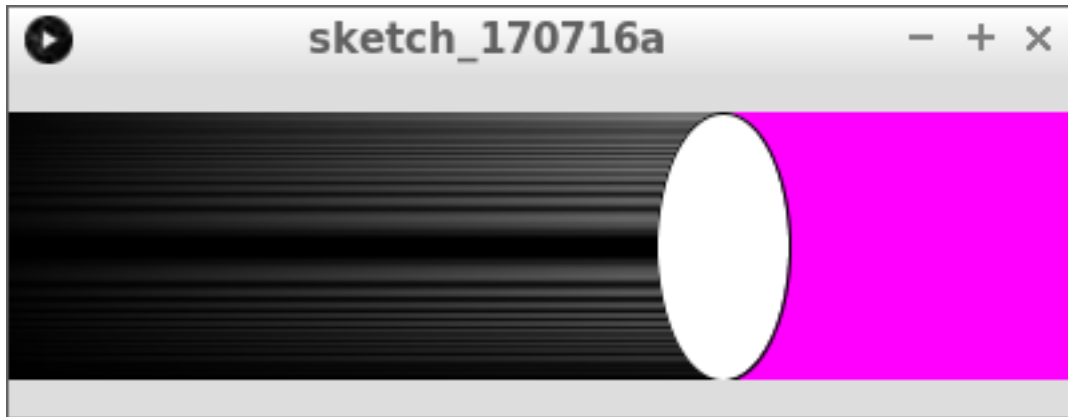


Figure 48: Oplossing 5

```
float x = -50;
float snelheid = 1;

void setup()
{
  size(400, 100);
  background(255, 0, 255);
}

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

Opdracht 6

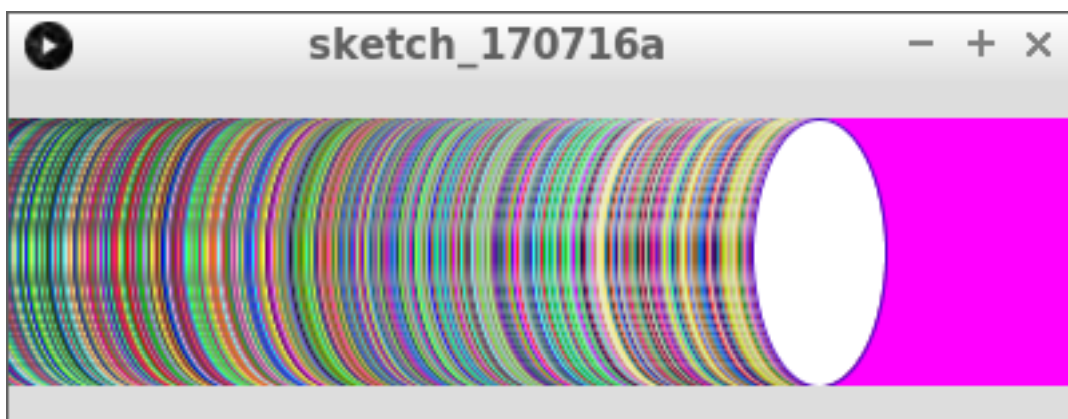


Figure 49: Opdracht 6

Maak de lijnkleur van de ovaal een willekeurige kleur: de rood, groen en blauw waarde moeten alledrie willekeurig zijn.

Oplossing 6

```
float x = -50;
float snelheid = 1;

void setup()
{
  size(400, 100);
  background(255, 0, 255);
}

void draw()
{
  stroke(random(256), random(256), random(256));
  fill(x, x, x);
  ellipse(x, height / 2, 50, 100);
  x = x + snelheid;
  if (x > width + 50)
  {
    x = -50;
  }
}
```

Eindopdracht

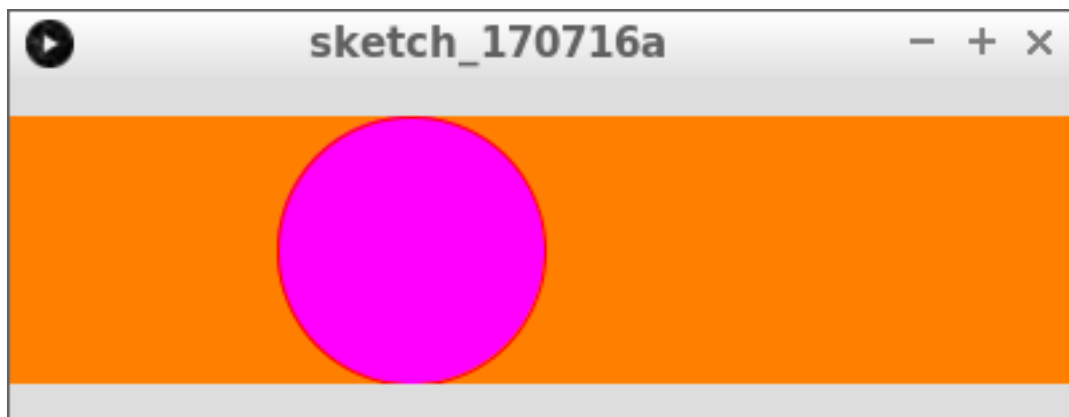


Figure 50: Eindopdracht `ellipse` en `background`

Maak de ovaal rond, even hoog als het scherm, magenta, en met een rode rand. De achtergrond moet oranje zijn (oranje is vol rood, halfvol groen en geen blauw). Er mag geen sliert komen van oude ovalen.

Bal die schuin stuitert

In deze les gaan we een bal schuin laten stuiten.

Een bal die horizontaal stuitert

```
float x = 150;
float snelheid_naar_rechts = 1;

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

void draw()
{
  ellipse(x, 100, 50, 50);
  x = x + snelheid_naar_rechts;
  if (x > 275)
  {
    snelheid_naar_rechts = -snelheid_naar_rechts;
  }
  if (x < 25)
  {
    snelheid_naar_rechts = -snelheid_naar_rechts;
  }
}
```

Opdracht 1

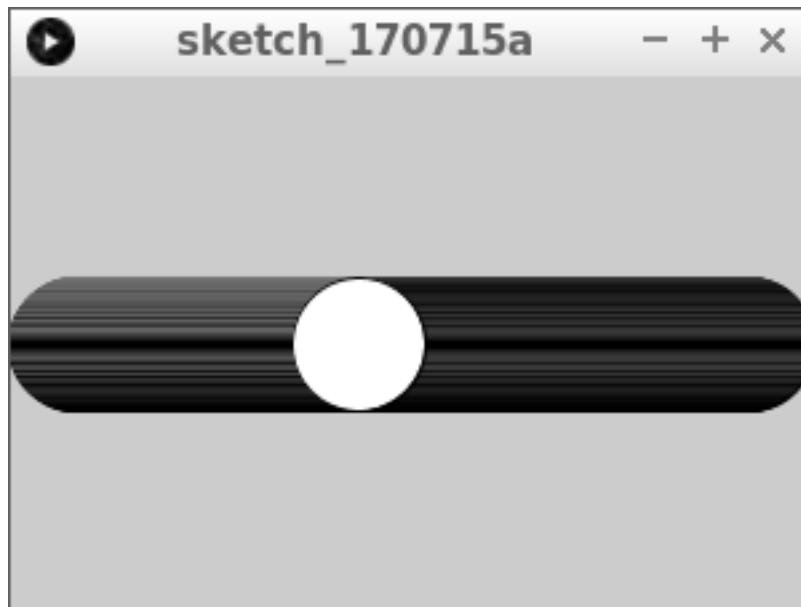


Figure 51: Opdracht 1

Maak een nieuwe variable aan met de naam `y`. Teken de ovaal op coördinaat `(x, y)`. Zorg dat de code hetzelfde blijft doen



`float y = 50;`

‘Lieve computer, onthoud het getal y met een beginwaarde van vijftig.’

Oplossing 1

```
float x = 150;
float y = 100;
float snelheid_naar_rechts = 1;

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

void draw()
{
  ellipse(x, y, 50, 50);
  x = x + snelheid_naar_rechts;
  if (x > 275)
  {
    snelheid_naar_rechts = -snelheid_naar_rechts;
  }
  if (x < 25)
  {
    snelheid_naar_rechts = -snelheid_naar_rechts;
  }
}
```

Opdracht 2

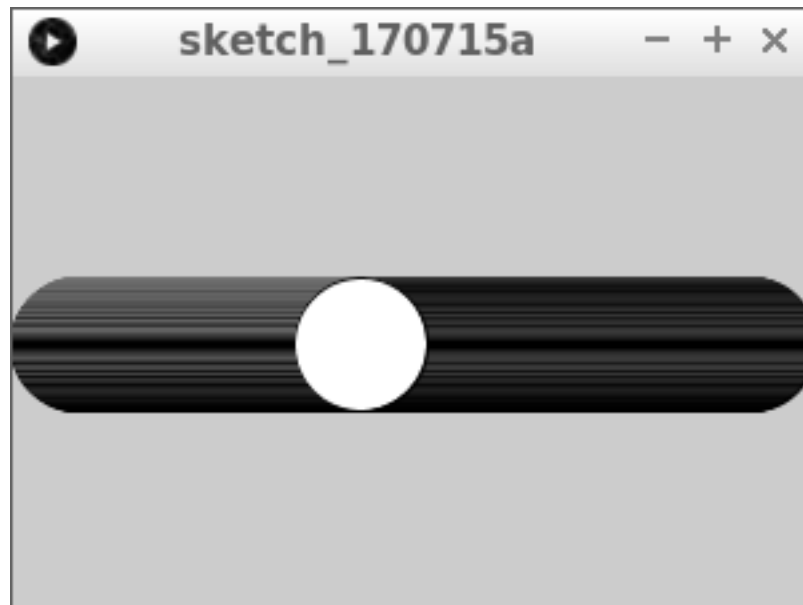


Figure 52: Opdracht 2

Maaak een nieuwe variable aan met de naam `snelheid_omlaag`. Geef deze de beginwaarde van 1. De bal hoeft nog niet te bewegen.

Oplossing 2

```
float x = 150;
float y = 100;
float snelheid_naar_rechts = 1;
float snelheid_omlaag = 1;

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

void draw()
{
  ellipse(x, y, 50, 50);
  x = x + snelheid_naar_rechts;
  if (x > 275)
  {
    snelheid_naar_rechts = -snelheid_naar_rechts;
  }
  if (x < 25)
  {
    snelheid_naar_rechts = -snelheid_naar_rechts;
  }
}
```

Opdracht 3



Figure 53: Opdracht 3

Laat de bal nu ook naar onder bewegen. Hij hoeft nog niet te stuiteren.

Oplossing 3

```
float x = 150;
float y = 100;
float snelheid_naar_rechts = 1;
float snelheid_omlaag = 1;

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

void draw()
{
  ellipse(x, y, 50, 50);
  x = x + snelheid_naar_rechts;
  y = y + snelheid_omlaag;
  if (x > 275)
  {
    snelheid_naar_rechts = -snelheid_naar_rechts;
  }
  if (x < 25)
  {
    snelheid_naar_rechts = -snelheid_naar_rechts;
  }
}
```

Opdracht 4

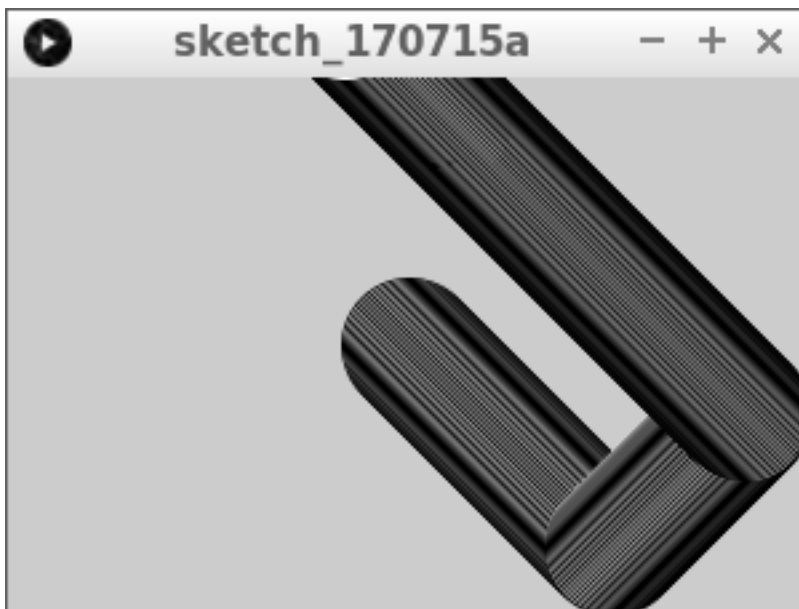


Figure 54: Opdracht 4

Laat de bal nu omhoog stuiteren als de bal de onderkant van het scherm raakt.

Oplossing 4

```
float x = 150;
float y = 100;
```



```

float snelheid_naar_rechts = 1;
float snelheid_omlaag = 1;

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

void draw()
{
  ellipse(x, y, 50, 50);
  x = x + snelheid_naar_rechts;
  y = y + snelheid_omlaag;
  if (x > 275)
  {
    snelheid_naar_rechts = -snelheid_naar_rechts;
  }
  if (x < 25)
  {
    snelheid_naar_rechts = -snelheid_naar_rechts;
  }
  if (y > 175)
  {
    snelheid_omlaag = -snelheid_omlaag;
  }
}

```

Eindopdracht

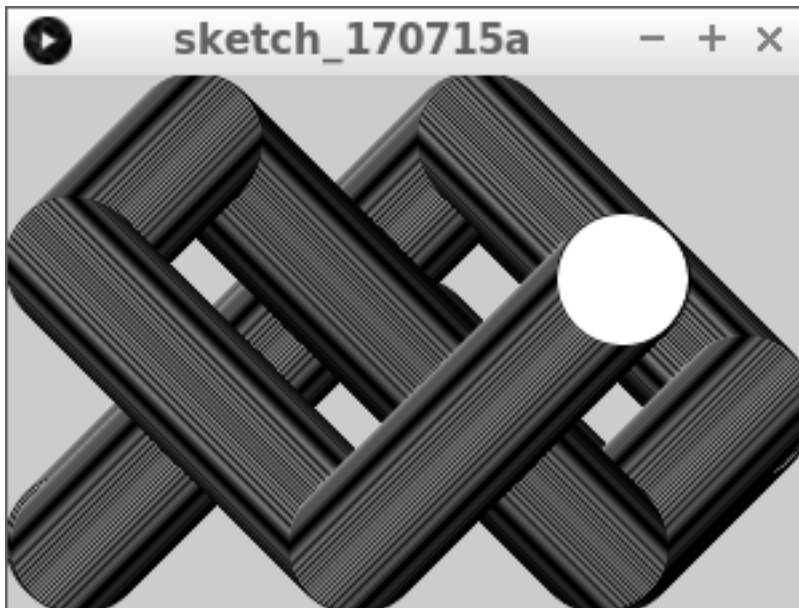


Figure 55: Eindopdracht Bal Die Schuin Stuitert

Laat nu de bal ook stuiten als deze de bovenkant raakt.

text

Tekst wordt veel gebruikt, ook in games, voor bijvoorbeeld een score.

Hier zie je 'Zork, the underground empire', een van de beroemdste tekstavonturen ooit:

A screenshot of a terminal window titled 'richel@druten: ~/Githubs/Zork'. The window has a menu bar with 'File', 'Edit', 'Tabs', and 'Help'. The terminal shows the command './Zork' being executed, followed by the game's welcome message: 'Welcome to Dungeon. This version created 11-MAR-91. You are in an open field west of a big white house with a boarded front door. There is a small mailbox here.' The prompt '>' is visible at the bottom left of the terminal area.

```
richel@druten: ~/Githubs/Zork
File Edit Tabs Help
richel@druten:~/Githubs/Zork$ ./Zork
Welcome to Dungeon.                This version created 11-MAR-91.
You are in an open field west of a big white house with a boarded
front door.
There is a small mailbox here.
>
```

Figure 56: Zork

In deze les gaan we leren

- hoe je tekst op het scherm zet
- hoe je berekeningen op het scherm zet
- hoe je tekst vergroot
- hoe je tekst een kleur geeft

Opdracht 1

Run deze code:

```
float x = 150;
float y = 100;
float snelheid_naar_rechts = 1;

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

void draw()
{
  background(0, 0, 0);
  stroke(128, 0, 0);
  fill(255, 0, 0);
  ellipse(x, y, 50, 50);
  x = x + snelheid_naar_rechts;
  if (x > 275)
  {
    snelheid_naar_rechts = -snelheid_naar_rechts;
  }
  if (x < 25)
  {
    snelheid_naar_rechts = -snelheid_naar_rechts;
  }
}
```

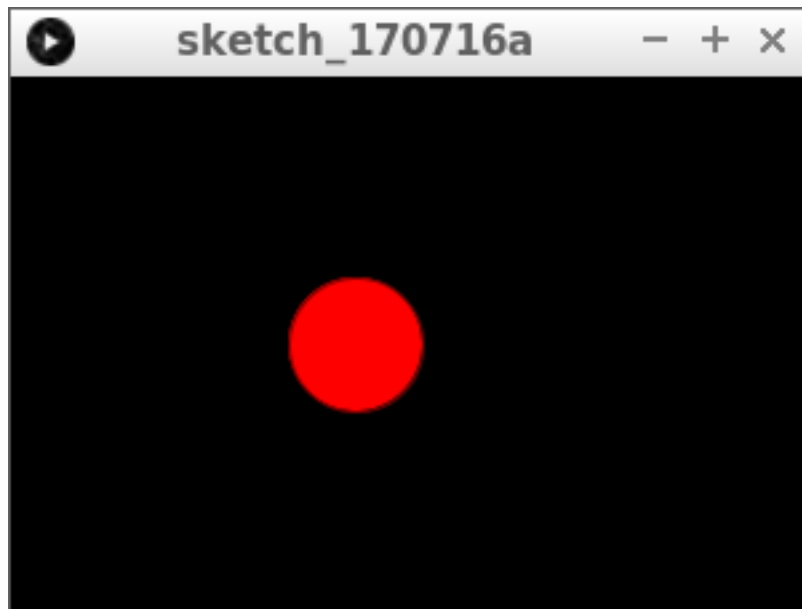


Figure 57: Oplossing 1

Oplossing 1

Opdracht 2

Voeg na `ellipse` de volgende regel toe:

```
text(x, 10, 20);
```

Oplossing 2

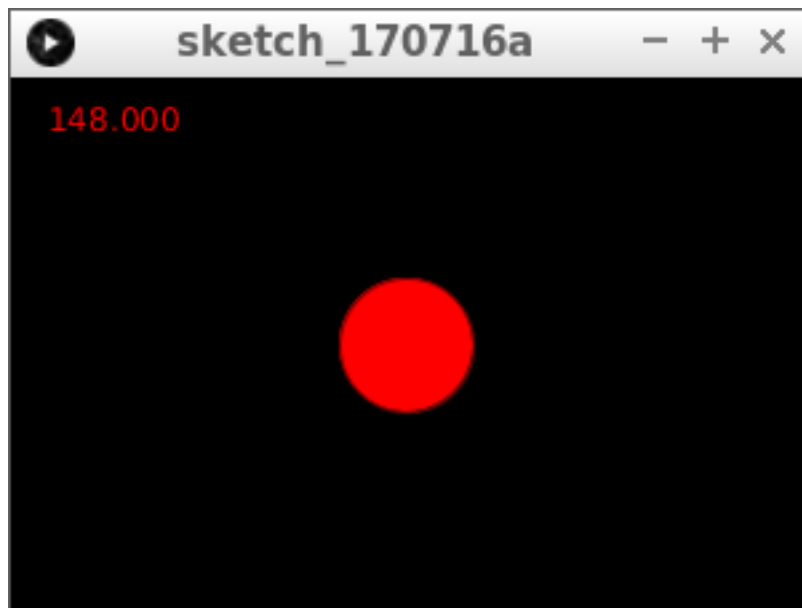


Figure 58: Oplossing 2

```
float x = 150;  
float y = 100;  
float snelheid_naar_rechts = 1;
```

```

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

void draw()
{
  background(0, 0, 0);
  stroke(128, 0, 0);
  fill(255, 0, 0);
  ellipse(x, y, 50, 50);
  text(x, 10, 20);
  x = x + snelheid_naar_rechts;
  if (x > 275)
  {
    snelheid_naar_rechts = -snelheid_naar_rechts;
  }
  if (x < 25)
  {
    snelheid_naar_rechts = -snelheid_naar_rechts;
  }
}

```



`text(100, 200, 300);`

'Lieve computer, zet de tekst 100 op het scherm met als linkerbovenhoek (200, 300)

`text("Hallo", 100, 200);`

'Lieve computer, zet de tekst Hallo op het scherm met als linkerbovenhoek (100, 200)

Opdracht 3

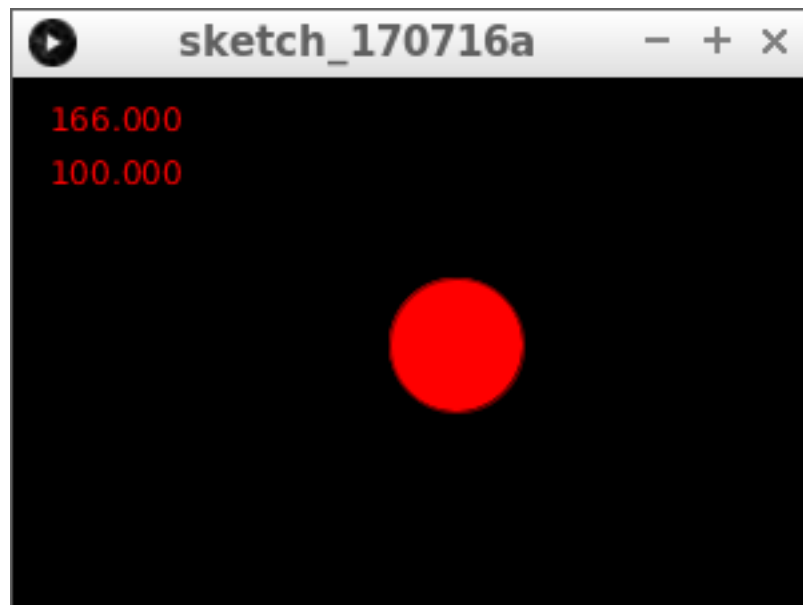


Figure 59: Opdracht 3

Het programma laat nu de x zien. Laat eronder nu ook de y coördinaat op het scherm zien.

Oplossing 3

```
float x = 150;
float y = 100;
float snelheid_naar_rechts = 1;

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

void draw()
{
  background(0, 0, 0);
  stroke(128, 0, 0);
  fill(255, 0, 0);
  ellipse(x, y, 50, 50);
  text(x, 10, 20);
  text(y, 10, 40);
  x = x + snelheid_naar_rechts;
  if (x > 275)
  {
    snelheid_naar_rechts = -snelheid_naar_rechts;
  }
  if (x < 25)
  {
    snelheid_naar_rechts = -snelheid_naar_rechts;
  }
}
```

Opdracht 4

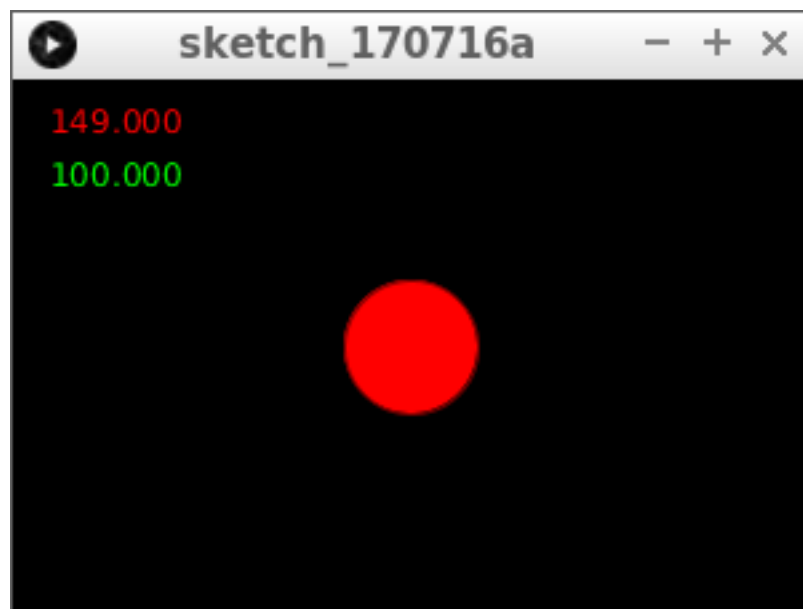


Figure 60: Opdracht 4

Met `fill` kun je de kleur van je tekst bepalen. Na de eerste `text`, zet de fill op groen.

Oplossing 4

```
float x = 150;
float y = 100;
float snelheid_naar_rechts = 1;

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

void draw()
{
  background(0, 0, 0);
  stroke(128, 0, 0);
  fill(255, 0, 0);
  ellipse(x, y, 50, 50);
  text(x, 10, 20);
  fill(0, 255, 0);
  text(y, 10, 40);
  x = x + snelheid_naar_rechts;
  if (x > 275)
  {
    snelheid_naar_rechts = -snelheid_naar_rechts;
  }
  if (x < 25)
  {
    snelheid_naar_rechts = -snelheid_naar_rechts;
  }
}
```

Opdracht 5

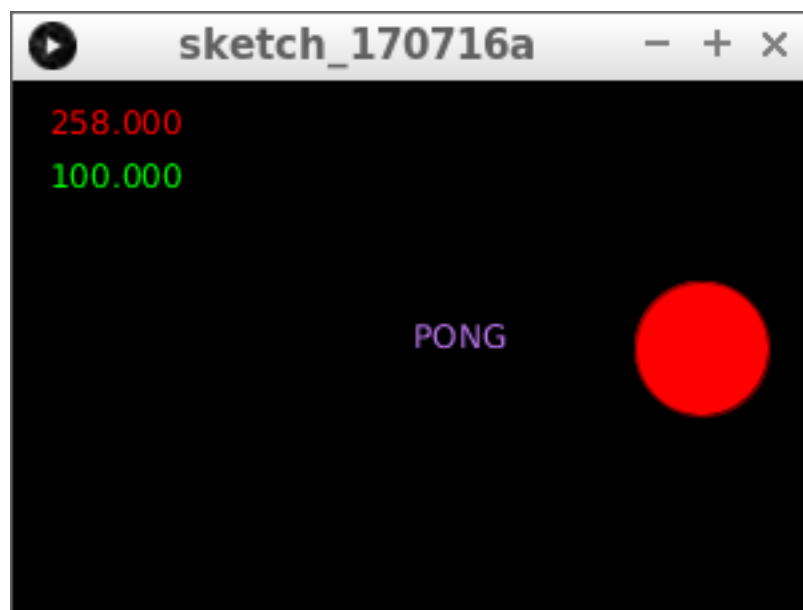


Figure 61: Opdracht 5

Zet de tekst PONG bovenin en midden op het scherm. De tekst moet een willekeurige kleur krijgen. Gebruik width om het midden te bepalen.

```

float x = 150;
float y = 100;
float snelheid_naar_rechts = 1;

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

void draw()
{
  background(0, 0, 0);
  stroke(128, 0, 0);
  fill(255, 0, 0);
  ellipse(x, y, 50, 50);
  text(x, 10, 20);
  fill(0, 255, 0);
  text(y, 10, 40);
  fill(random(255), random(255), random(255));
  text("PONG", width / 2, height / 2);
  x = x + snelheid_naar_rechts;
  if (x > 275)
  {
    snelheid_naar_rechts = -snelheid_naar_rechts;
  }
  if (x < 25)
  {
    snelheid_naar_rechts = -snelheid_naar_rechts;
  }
}

```

Opdracht 6

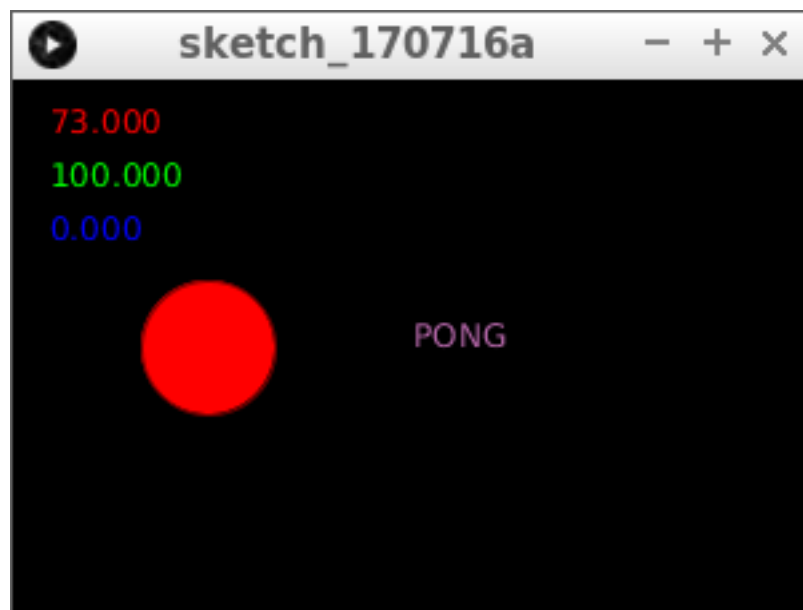


Figure 62: Opdracht 6

Maak een variabele erbij: `aantal_botsingen`. `aantal_botsingen` heeft als beginwaarde nul. Zet deze waarde op het scherm, in het wit.ex

Oplossing 6

```
float x = 150;
float y = 100;
float snelheid_naar_rechts = 1;
float botsingen = 0;

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

void draw()
{
  background(0, 0, 0);
  stroke(128, 0, 0);
  fill(255, 0, 0);
  ellipse(x, y, 50, 50);
  text(x, 10, 20);
  fill(0, 255, 0);
  text(y, 10, 40);
  fill(0, 0, 255);
  text(botsingen, 10, 60);
  fill(random(255), random(255), random(255));
  text("PONG", width / 2, height / 2);
  x = x + snelheid_naar_rechts;
  if (x > 275)
  {
    snelheid_naar_rechts = -snelheid_naar_rechts;
  }
  if (x < 25)
  {
    snelheid_naar_rechts = -snelheid_naar_rechts;
  }
}
```

Opdracht 7

Maak botsingen hoger als de bal teken de rechter muur komt.

Oplossing 7

```
float x = 150;
float y = 100;
float snelheid_naar_rechts = 1;
float botsingen = 0;

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

void draw()
{
  background(0, 0, 0);
  stroke(128, 0, 0);
  fill(255, 0, 0);
```

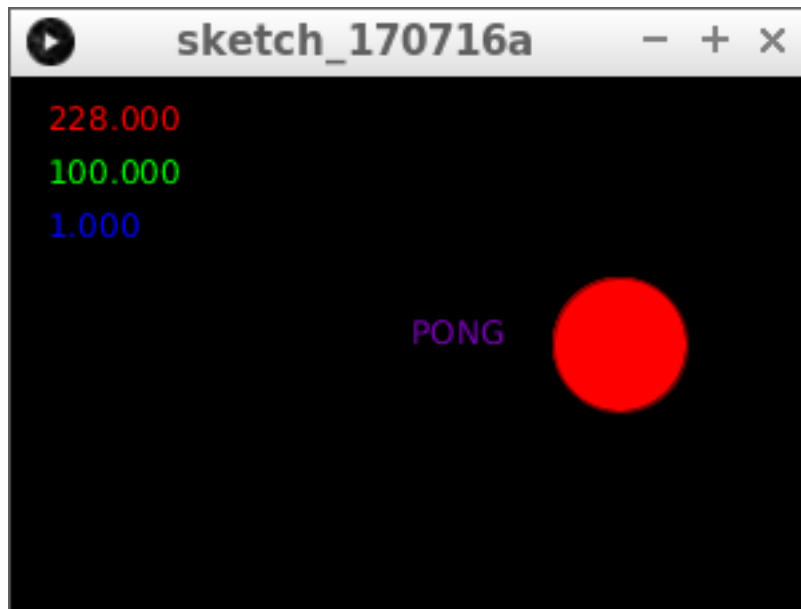



Figure 63: Opdracht 7

```

ellipse(x, y, 50, 50);
text(x, 10, 20);
fill(0, 255, 0);
text(y, 10, 40);
fill(0, 0, 255);
text(botsingen, 10, 60);
fill(random(255), random(255), random(255));
text("PONG", width / 2, height / 2);
x = x + snelheid_naar_rechts;
if (x > 275)
{
    snelheid_naar_rechts = -snelheid_naar_rechts;
    botsingen = botsingen + 1;
}
if (x < 25)
{
    snelheid_naar_rechts = -snelheid_naar_rechts;
}
}

```

Eindopdracht

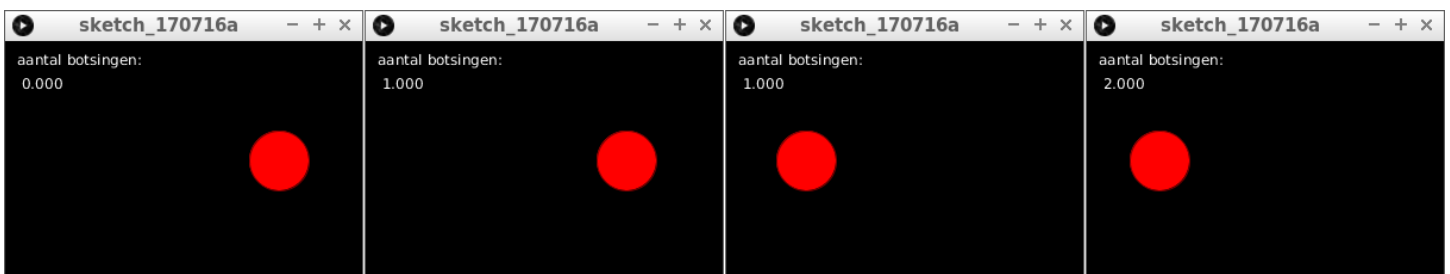


Figure 64: Eindopdracht text

Zet linksboven de tekst: `aantal botsingen:` in het wit. Zet daaronder de waarde van `botsingen` in het wit. Haal de rest van de tekst weg. Het aantal botsingen moet ook hoger worden als je links botst.