



Figure 1: 1: Tekenen

Contents

Voorwoord	2
Over dit boek	3
Processing opstarten op cursuslaptop	3
Terminal starten	3
Processing starten	4
Processing slimmer starten	4
Updaten	8
Eindppdracht	9
Een Mooi Programma	9
Wat je nodig hebt	11
Code kopiëren	11
Programma uitvoeren	12
Sneltoetsen oefenen	12
Point	12
Pixels	15
Hoe zitten pixels op je beeldscherm	15
Een puntje tekenen	15
Opdrachten	16
Oplossing	17
Coördinaten	17
Vragen	18
Puntjes tekenen	18
Vragen	20
Oplossing	20
Eindopdracht	22
Line	22
Lijnen	24
Vragen	26
Oplossing	30
Eindopdracht	30
backGround	30
Kleuren	33
Vragen	33
Antwoorden	36
background	36
Opdracht	36
Oplossingen	37
Eindopdracht	38
stroke	38

stroke	39
Opdracht	41
Oplossing	41
Eindopdracht	42
Rect	42
Rechthoeken	43
Vragen	44
Oplossing	45
Eindopdracht	46
Ellipse	46
Ovalen	48
Vragen	48
Oplossing	50
Eindopdracht	52
fill	52
Kleuren	52
Vragen	52
Antwoorden	54
background	54
Opdracht	55
Oplossingen	55
Eindopdracht	56
text	56
Tekst	57
Opdracht	57
Oplossing	57
Opdracht	58
Oplossing	58
Opdracht	58
Oplossing	58
Opdracht	59
Oplossing	59
Eindopdracht	60

Voorwoord

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



Figure 1: Het logo van De Jonge Onderzoekers



Figure 2: Het logo van Codestarter

Over dit boek

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



Figure 3: De licentie van dit boek

(C) Dojo Groningen 2016

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.

Processing opstarten op cursuslaptop

We programmeren in Processing. Processing is een programma. Dit programma moeten we dus eerst opstarten.

Terminal starten

Eerst moeten we een terminal starten.

Wij hebben twee soorten laptops:

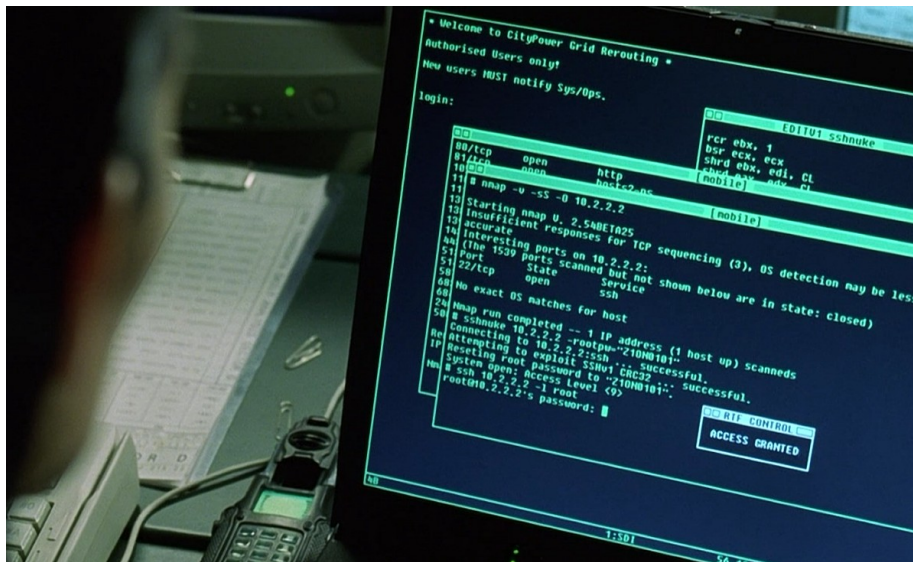


Figure 4: Hackers werken met een terminal

- Debian (spreek uit 'Debie-jen')
- Ubuntu (spreek uit 'Loe-boen-toe')

Start een Terminal:

- Debian: druk op de WIN toets (deze zit links onder, tussen CTRL en ALT). Type dan 'terminal' en dan ENTER
- Ubuntu: druk op CTRL + ALT + T

Processing starten

In de terminal, type:

```
cd Programs/processing-3.2.1
```

En type dan:

```
./processing
```

Klaar!

Processing slimmer starten

Je hoeft niet alles helemaal te typen. Een terminal kan je ook helpen. Als je in een terminal op Tab drukt, maakt deze het woord af.

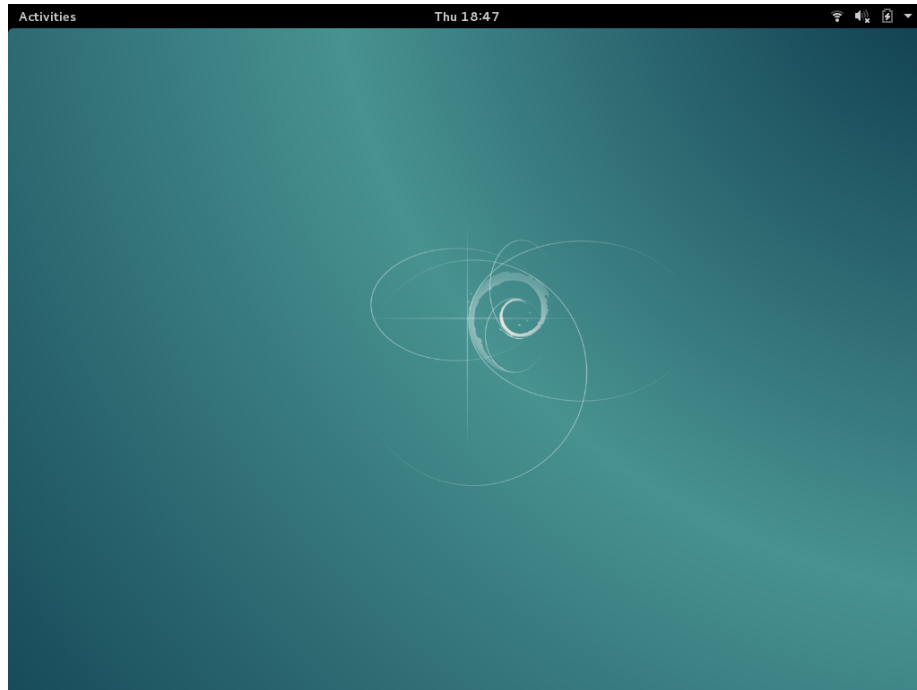


Figure 5: Debian: druk op WIN

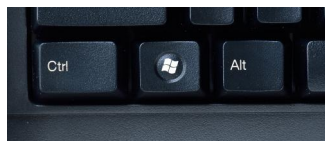


Figure 6: De WIN toets zit tussen CTRL en ALT

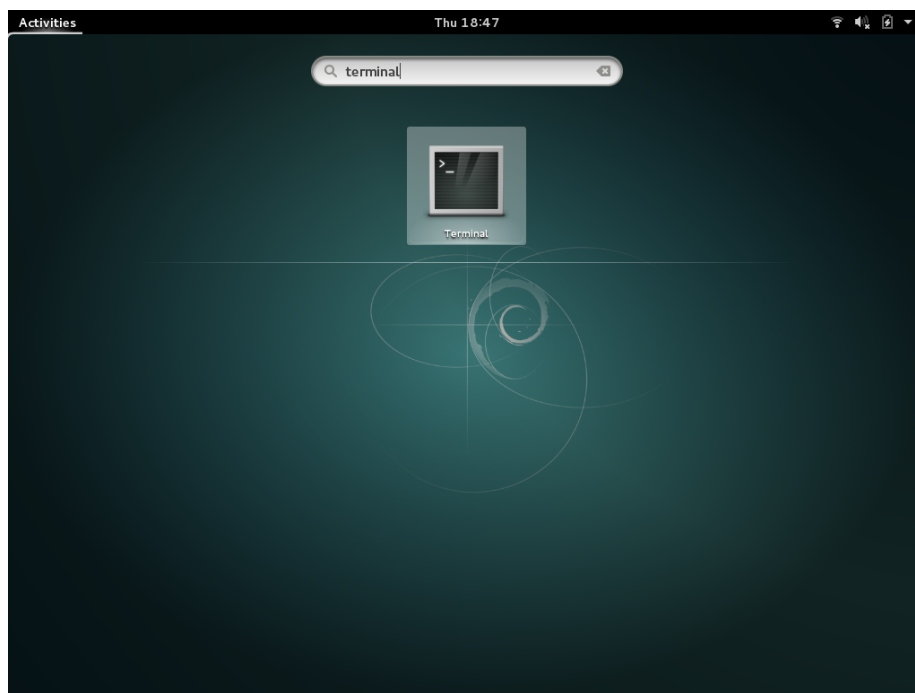


Figure 7: Debian: druk op WIN en type dan 'terminal'

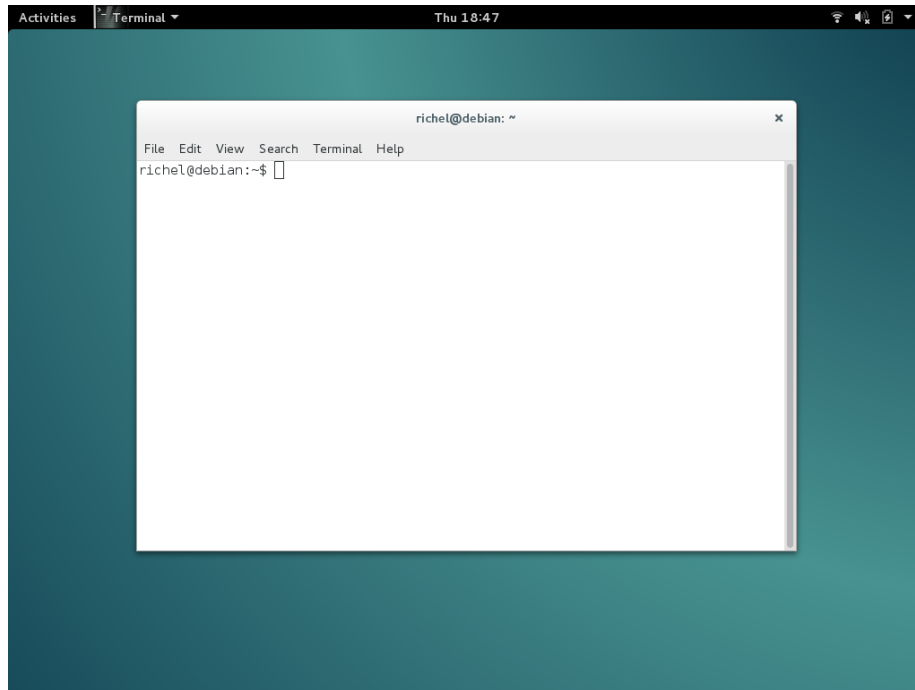


Figure 8: Debian: een terminal

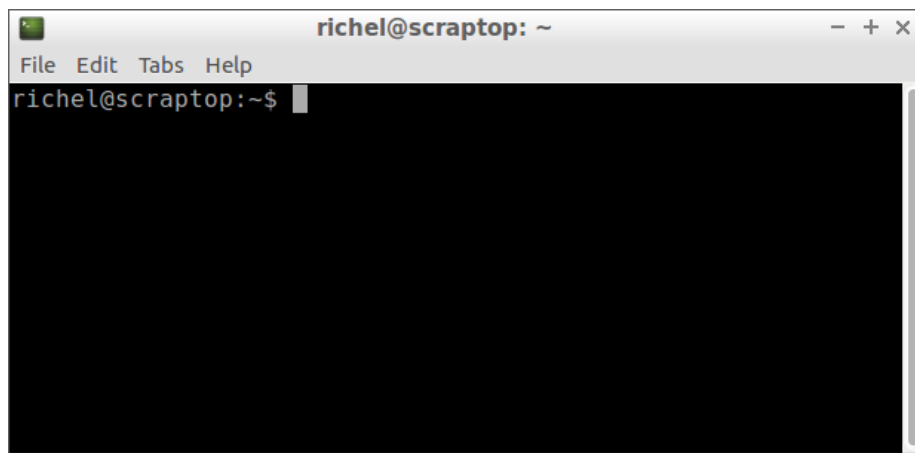


Figure 9: Lubuntu: een terminal

In de terminal, type:

```
cd Progr
```

en druk dan op 'Tab'.

Nu wordt staat er op je scherm:

```
cd Programs/
```

Maak ervan

```
cd Programs/pr
```

en druk dan op 'Tab'.

Nu staat er:

```
cd Programs/processing-3.2.1
```

druk dan op Enter.

Doe nu:

```
./p
```

en druk dan op Tab. Nu staat er:

```
./processing
```

Druk op Enter.

Klaar!

Updaten

Het kan gebeuren dat er een venster komt met een Update. Het venster vraagt of je een nieuwe versie van Processing wilt installeren. Klik op 'No'. 'No' is Engels voor 'nee'.

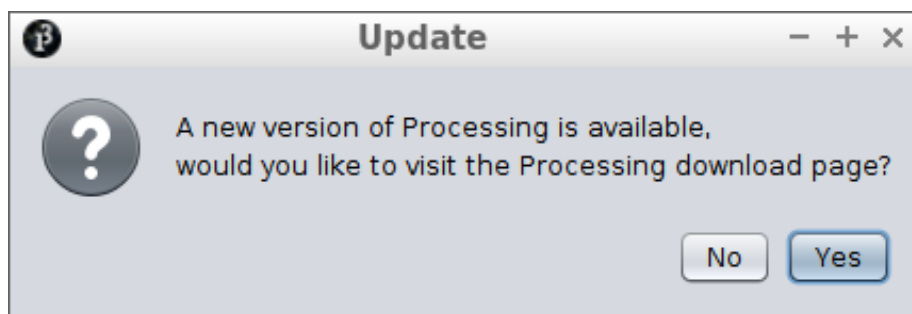


Figure 10: Processing Update venster, klik op 'No'

Eindppdracht

- Start Processing helemaal zonder hulp

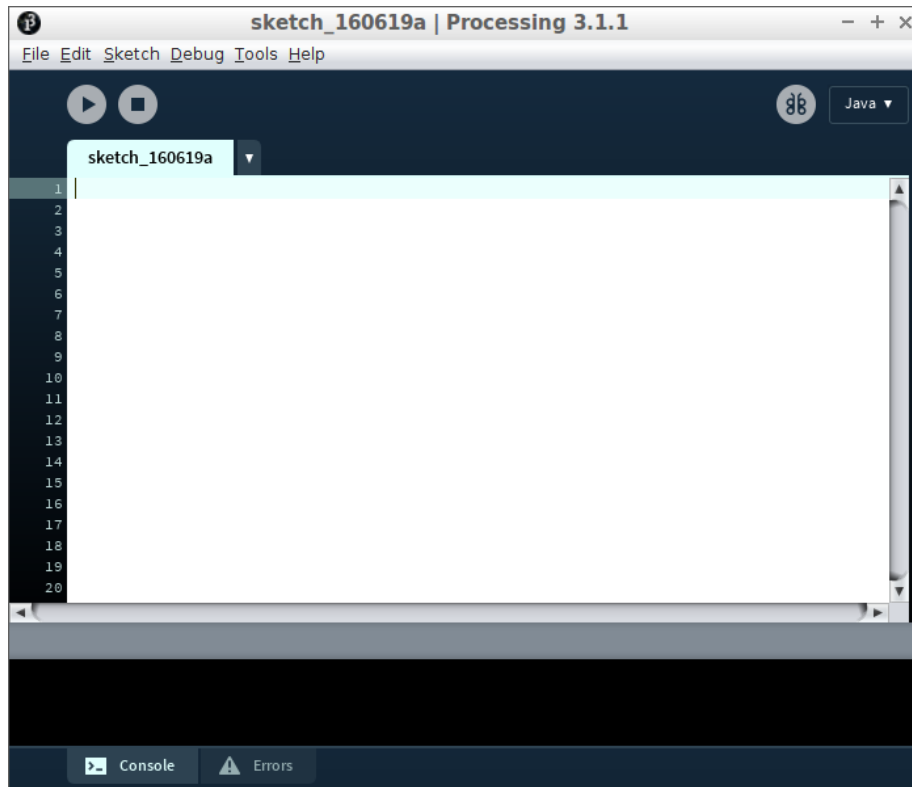


Figure 11: Processing zonder code

Een Mooi Programma

Processing is een programmeertaal ontwikkeld voor kunstenaars en erg geschikt om games en mooie dingen mee te maken.

In deze les gaan we leren

- hoe we Processing opstarten
- hoe je code naar Processing kopieert
- hoe je het programma start

Zo ziet het programma eruit:

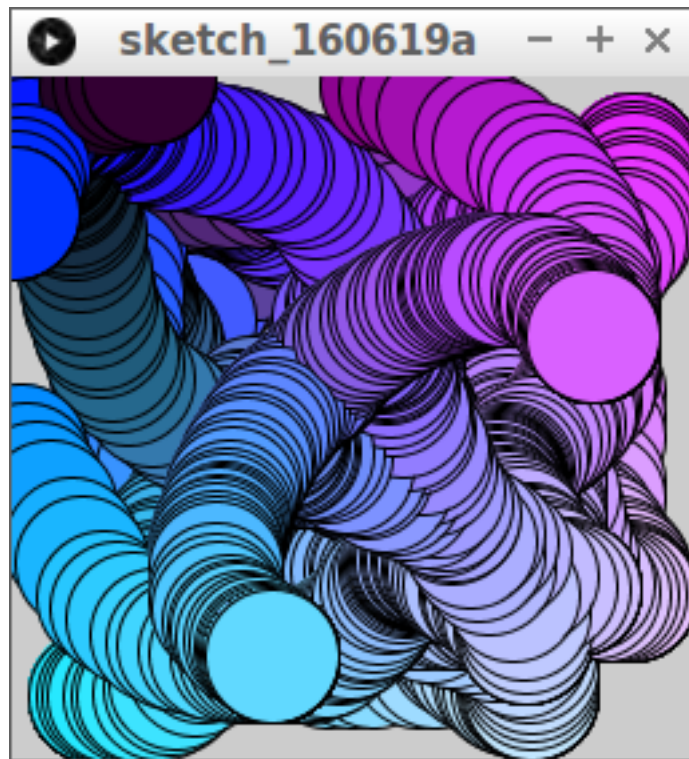


Figure 12: EenMooiProgramma

Wat je nodig hebt

Je moet Processing op kunnen starten. Hoe dat moet, hangt af van het besturingssysteem:

- Processing opstarten op cursus laptop
- Processing installeren op eigen laptop met GNU/Linux
- Processing installeren op eigen laptop met Windows

Code kopiëren

Processing begint met een leeg programma zonder code:

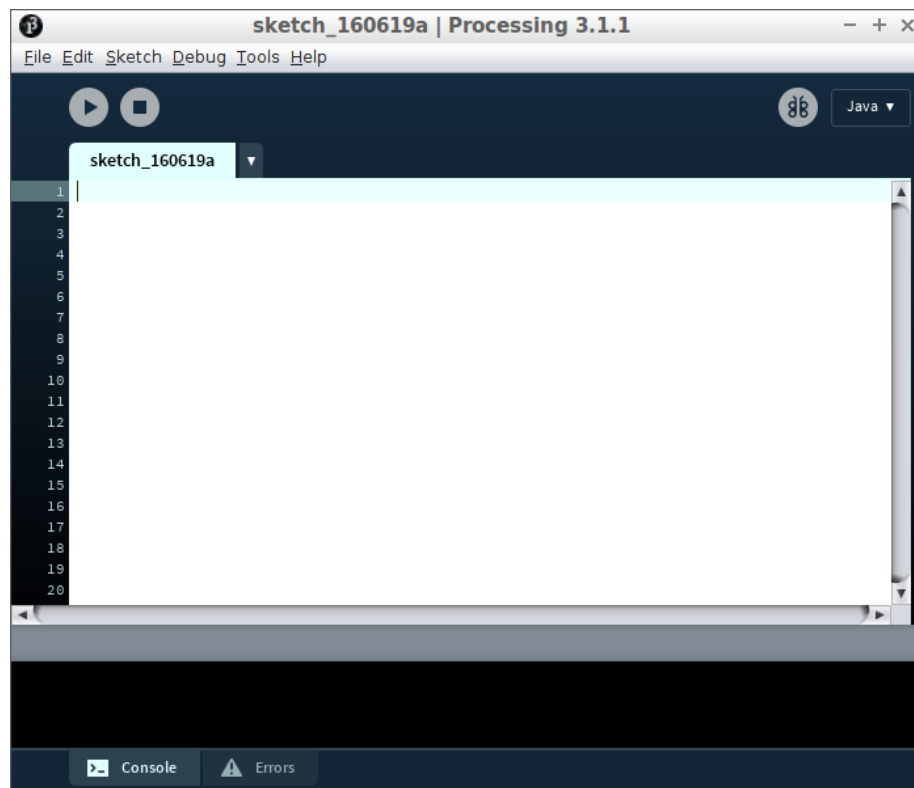


Figure 13: Processing zonder code

Dit is de programmeercode die we gaan gebruiken:

```
void setup()
{
  size(256,256);
}
```

```

}

void draw()
{
  fill(mouseX, mouseY, mouseX + mouseY);
  ellipse(mouseX, mouseY, 50, 50);
  fill(mouseY, mouseX, 255);
  ellipse(mouseY, mouseX, 50, 50);
}

```

Wat de code precies doet, leggen we later uit. Voor nu is het genoeg te weten dat het iets moois doet.

Om code te kopiëren gebruik je sneltoetsen:

- SHIFT + pijltjes: selecteren
- CTRL + A: alles selecteren
- CTRL + C: kopiëren van selectie
- CTRL + X: knippen van selectie
- CTRL + V: plakken van selectie
- Start Processing
- Kopieer deze code naar Processing

Programma uitvoeren

- Klik op de ‘Run’ knop

Als het goed is, zie je nu het programma!

Sneltoetsen oefenen

- Werk met iemand samen. Hussel de code van de andere door de war, door deze te knippen/kopiëren en te plakken. Repareer dan de code op je eigen computer

Point

Processing is een programmeertaal ontwikkeld voor kunstenaars en erg geschikt om games en mooie dingen mee te maken.

In deze les gaan we leren

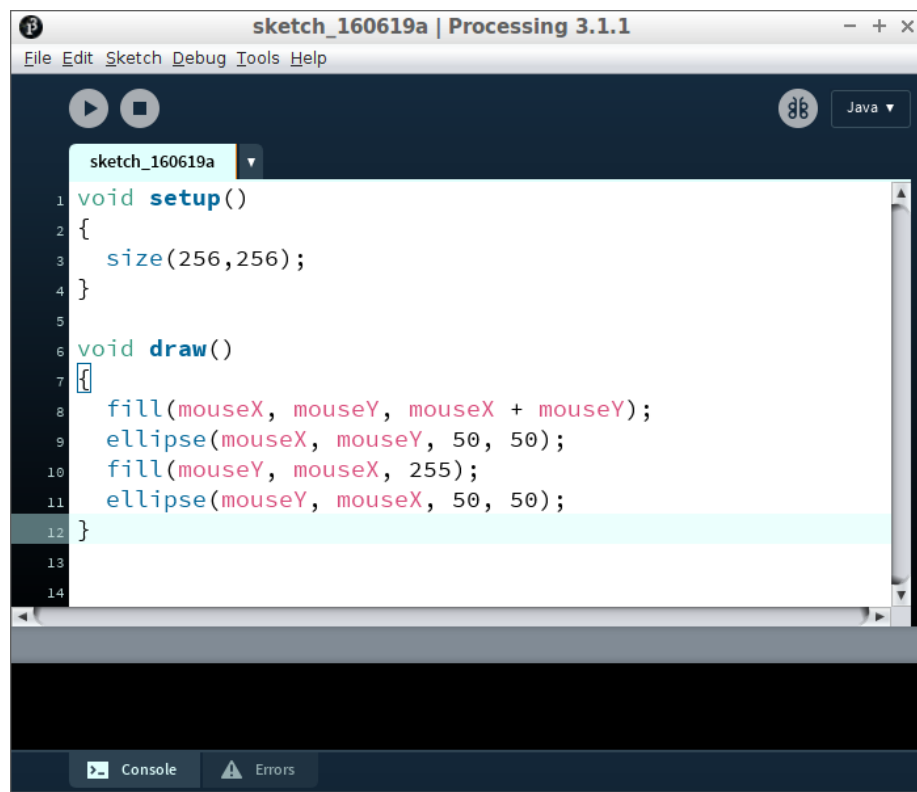


Figure 14: Processing met code

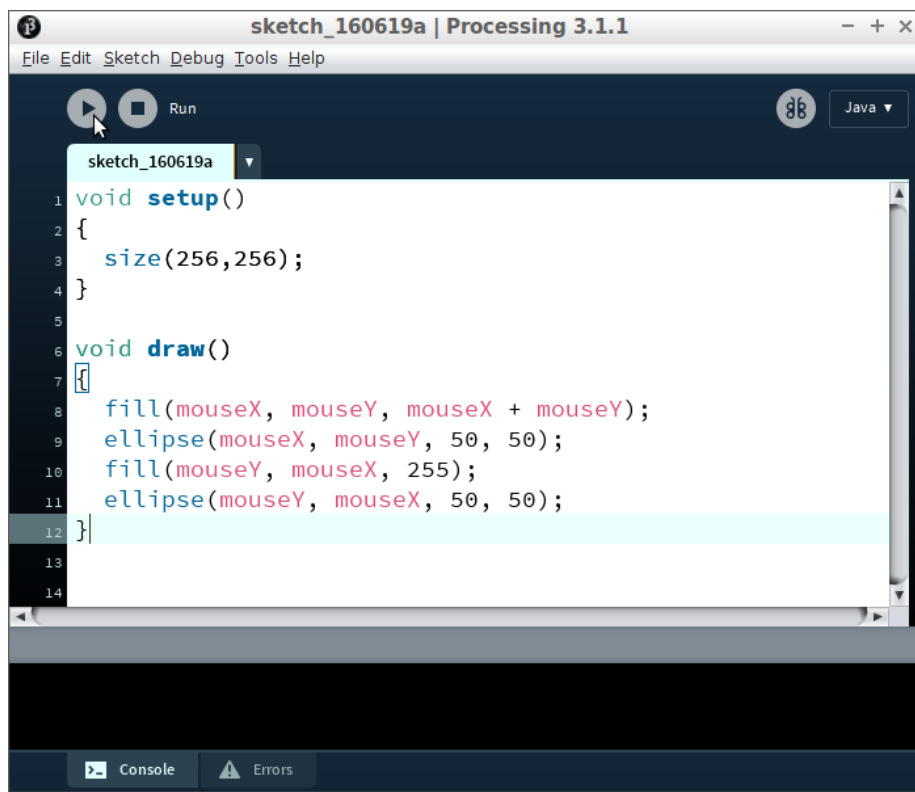


Figure 15: De Run knop

- wat pixels zijn
- hoe de pixels op een beeldscherm zitten
- hoe je puntjes tekent

Zo gaat het eruit zien:

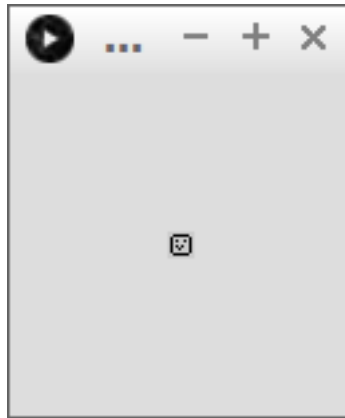


Figure 16: Point

Pixels

Pixels zijn de vierkantjes waaruit je beeldscherm is opgebouwd. Hoe meer pixels je scherm heeft, hoe scherper het beeld eruit ziet. Dat zie je goed bij oude games: die hebben minder pixels:

Hoe zitten pixels op je beeldscherm

Elke pixel op je beeldscherm heeft een soort adres. Dit adres noemen we een coördinaat. Een coördinaat bestaat uit twee getallen.

De pixel linksbovenin heeft als coördinaat (0,0). (0,0) spreek je uit als nul komma nul.

Een puntje tekenen

Hier zie je code om een puntje te tekenen:

```
void setup()
{
  size(200,160);
}
```



```
void draw()
{
    point(0,0);
}
```

Op het scherm ziet dit er zo uit:



Figure 17: Een puntje

Als je goed kijkt zie je linksbovenin een zwart puntje. Als we inzoomen, zien we het beter:

De zwarte stip is de pixel die we hebben getekent!

Opdrachten

- 1. Wat doet `size(200,160)` ook alweer?
- 2. Verander `point(0,0)` naar `point(1,0)` en run het programma. Wat zie je?
- 3. Verander `point(0,0)` naar `point(0,1)` en run het programma. Wat zie je?
- 4. Wat betekent het eerste getal van een coördinaat?
- 5. Wat betekent het tweede getal van een coördinaat?



Figure 18:

Oplossing

- 1. Daarmee maak je het venster 200 pixels breed en 160 pixels hoog
- 2. Het puntje verschuift een pixel naar rechts
- 3. Het puntje verschuift een pixel omlaag
- 4. Het eerste getal van een coördinaat bepaalt hoeveel pixels naar *rechts* het puntje wordt getekent
- 5. Het tweede getal van een coördinaat bepaalt hoeveel pixels naar *onder* het puntje wordt getekent

Coördinaten

Een beeldscherm heeft veel coördinaten. In dit plaatje zie je de coördinaten van de pixels linksbovenin het scherm:

Dan zie je dat het eerste getal is hoeveel pixels naar recht hiervan is. Zo zit (1,0) rechts van (0,0). Het tweede getal is hoeveel pixels je onder (0,0) zit. Zo zit (0,1) onder (0,0).

Op deze manier kun je elke pixel op je beeldscherm vinden.

(0,0)	(0,1)	(0,2)	(0,3)	(0,4)	(0,5)	(0,6)	(0,7)	(0,8)	(0,9)
(1,0)	(1,1)	(1,2)	(1,3)	(1,4)	(1,5)	(1,6)	(1,7)	(1,8)	(1,9)
(2,0)	(2,1)	(2,2)	(2,3)	(2,4)	(2,5)	(2,6)	(2,7)	(2,8)	(2,9)
(3,0)	(3,1)	(3,2)	(3,3)	(3,4)	(3,5)	(3,6)	(3,7)	(3,8)	(3,9)
(4,0)	(4,1)	(4,2)	(4,3)	(4,4)	(4,5)	(4,6)	(4,7)	(4,8)	(4,9)
(5,0)	(5,1)	(5,2)	(5,3)	(5,4)	(5,5)	(5,6)	(5,7)	(5,8)	(5,9)
(6,0)	(6,1)	(6,2)	(6,3)	(6,4)	(6,5)	(6,6)	(6,7)	(6,8)	(6,9)
(7,0)	(7,1)	(7,2)	(7,3)	(7,4)	(7,5)	(7,6)	(7,7)	(7,8)	(7,9)
(8,0)	(8,1)	(8,2)	(8,3)	(8,4)	(8,5)	(8,6)	(8,7)	(8,8)	(8,9)
(9,0)	(9,1)	(9,2)	(9,3)	(9,4)	(9,5)	(9,6)	(9,7)	(9,8)	(9,9)

Figure 19: Pixel coördinaat

Vragen

- Welke coördinaten heb je nodig om een horizontale lijn te tekenen, die door (0,0) gaat en drie pixels lang is?
- Welke coördinaten heb je nodig om een verticale lijn te tekenen, die door (1,0) gaat en drie pixels hoog is?
- Welke coördinaten heb je nodig om een diagonale lijn te tekenen, die door (0,0) gaat en drie pixels lang is?
- Welke coördinaten heb je nodig om een vierkantje te tekenen, die door (0,0) gaat, twee pixels breed en twee pixels breed is?

Puntjes tekenen

Dit is de programmeercode die je nodig hebt:

```
void setup()
{
  size(10,10);
}

void draw()
{
```

```
// 0123456789
// 0 .....
// 1 ..XXXXXX..
// 2 .X.....X.
// 3 .X.X..X.X.
// 4 .X.....X.
// 5 .X.X.X..X.
// 6 .X..X...X.
// 7 .X.....X.
// 8 ..XXXXXX..
// 9 .....
```

```
// Bovenkant hoofd
point(2,1);
point(3,1);
point(4,1);
point(5,1);
point(6,1);
point(7,1);
```

```
// Rechterkant hoofd
point(8,2);
point(8,3);
point(8,4);
point(8,5);
point(8,6);
point(8,7);
```

```
// Onderkant hoofd
point(2,8);
point(3,8);
point(4,8);
point(5,8);
point(6,8);
point(7,8);
```

```
// Linkerkant hoofd
point(1,2);
point(1,3);
point(1,4);
point(1,5);
point(1,6);
point(1,7);
```

```
// Ogen
```

```

point(3,3);
point(6,3);

// Mond
point(3,5);
point(4,6);
point(5,5);

}

```

Dit is wat de code doet:

- `void setup() {}`: de setup functie, de computer voert een keer alles tussen de accolades uit
- `size(10, 10)`: maak een scherm van 10 pixels breed en 10 pixels hoog
- `;`: de puntkomma geeft in Processing het einde aan van een zin. Dit is ongeveer hetzelfde met een punt in schrijftaal.
- `void draw() {}`: de draw functie, de computer voert steeds alles tussen de accolades uit
- `//`: commentaar: de rest van de regel wordt niet gelezen door de computer. Hier kun je dingen neerzetten die speciaal bedoelt zijn voor mensen, zoals jezelf. In dit programma zie je dat het poppetje is getekent in commentaar, met de coördinaten aan de zijkant. Ook zie je dat de programmeur zegt wat er getekend wordt: zo kun je de fouten sneller vinden
- `point(2,1)`: teken een puntje op coördinaat (2,1).

Vragen

- Start Processing. Kopieer deze code en run het programma
- Laat de smiley een pixel breder lachen
- Draai de lach verticaal om, zodat de smiley sip gaat kijken
- Geef de smiley punk haar door drie pixels bovenop het hoofd erbij te tekenen
- Geef de smiley een sik van een pixel

Oplossing

```

void setup()
{
  size(10,10);
}

void draw()
{

```

```
// 0123456789
// 0 ..X.X..X..
// 1 ..XXXXXX..
// 2 .X.....X.
// 3 .X.X..X.X.
// 4 .X.....X.
// 5 .X..XX..X.
// 6 .X.X..X.X.
// 7 .X.....X.
// 8 ..XXXXXX..
// 9 ....X.....
```

```
// Bovenkant hoofd
point(2,1);
point(3,1);
point(4,1);
point(5,1);
point(6,1);
point(7,1);
```

```
// Rechterkant hoofd
point(8,2);
point(8,3);
point(8,4);
point(8,5);
point(8,6);
point(8,7);
```

```
// Onderkant hoofd
point(2,8);
point(3,8);
point(4,8);
point(5,8);
point(6,8);
point(7,8);
```

```
// Linkerkant hoofd
point(1,2);
point(1,3);
point(1,4);
point(1,5);
point(1,6);
point(1,7);
```

```
// Ogen
```

```

point(3,3);
point(6,3);

// Mond
point(3,6);
point(4,5);
point(5,5);
point(6,6);

// Punkhaar
point(2,0);
point(4,0);
point(7,0);

// Sik
point(9,4);
}

```

Eindopdracht

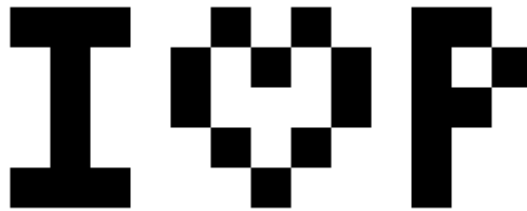


Figure 20: Point Eindopdracht

Hier zie je een uitvergroot plaatje. Maak deze in het klein na met pixels.

Line

Zonder lijnen kun je bijna geen games maken. Een van de allereerste successgames, Asteroids, bestond vooral uit lijnen:

Je kunt een lijn tekenen met een boel puntjes, maar de `line` functie werkt gemakkelijker.

In deze les gaan we leren

- hoe je lijnen tekent

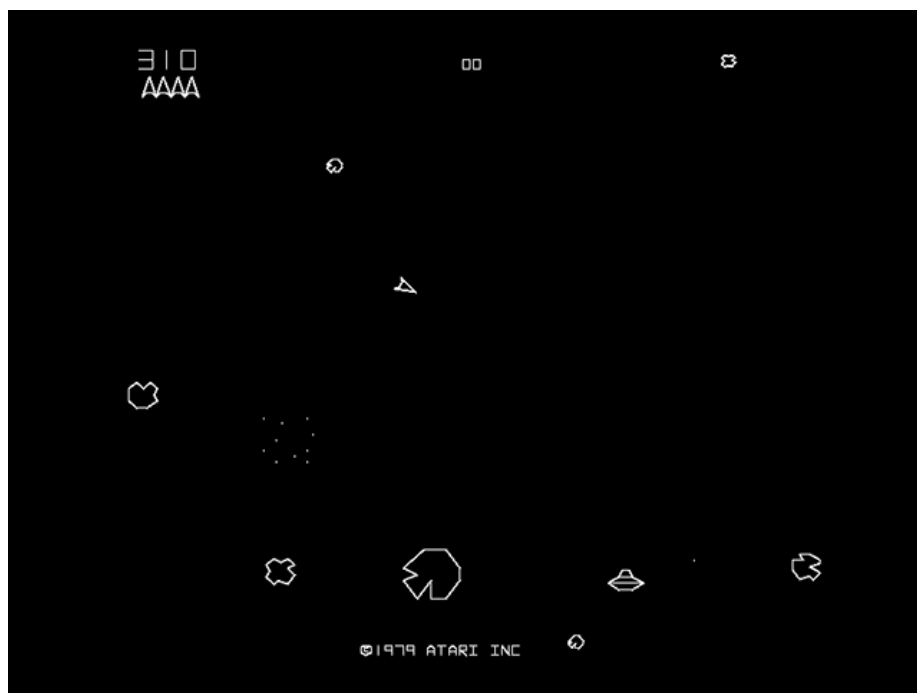


Figure 21: Asteroids

Zo gaat het eruit zien:

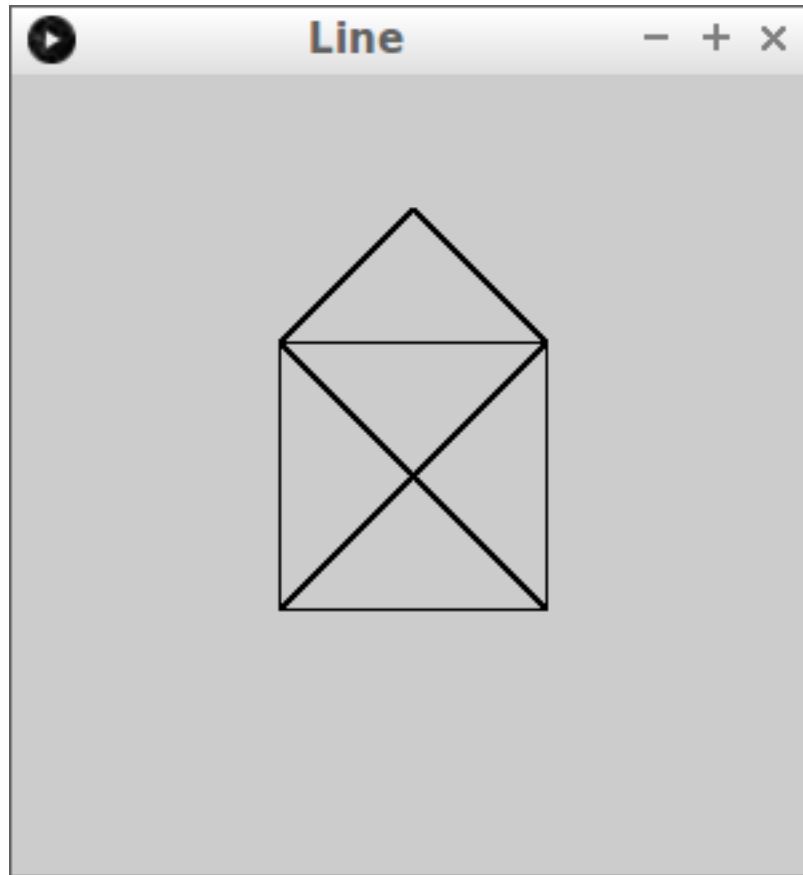


Figure 22: Line

Kun je nog geen puntjes tekenen? Ga dan naar de les waarin je puntjes leert tekenen

Lijnen

Een lijn bestaat uit pixels. Om een lijn te tekenen, moet je een beginpixel en eindpixel kiezen. Processing tekent dan zelf de pixels ertussenin.

Om in Processing een lijn te tekenen, gebruik je de functie `line`. De functie `line` heeft vier getallen nodig. Deze vier getallen zijn twee coördinaten achter elkaar. De eerste twee getallen zijn de coördinaat van het ene eind van de lijn. De laatste twee getallen zijn de coördinaat van het ander eind van de lijn.

Hier zie je een lijn die gaat van coördinaat (1,2) naar (3,2):

(0, 0)	(1, 0)	(2, 0)	(3, 0)	(4, 0)
(0, 1)	(1, 1)	(2, 1)	(3, 1)	(4, 1)
(0, 2)				(4, 2)
(0, 3)	(1, 3)	(2, 3)	(3, 3)	(4, 3)
(0, 4)	(1, 4)	(2, 4)	(3, 4)	(4, 4)

Figure 23: Lijn 1

In Processing teken je deze lijn met:

```
line(1,2,3,2);
```

De volgorde van de twee coördinaten maakt niet uit.

Lijnen kunnen ook schuin gaan.

Hier zie je een lijn die gaat van coördinaat (2,4) naar (1,1):

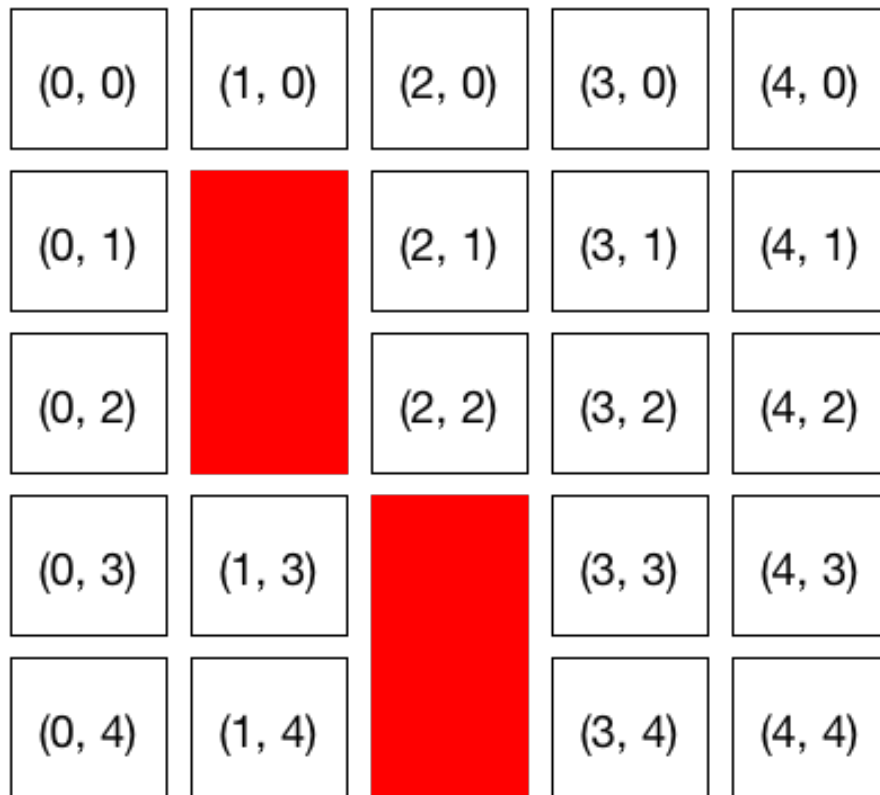


Figure 24: Lijn 2

In Processing teken je deze lijn met:

```
line(2,4,1,1);
```

Vragen

- 1. Hierboven staat een lijn. Wat zijn de begin- en eindcoördinaat van die lijn?

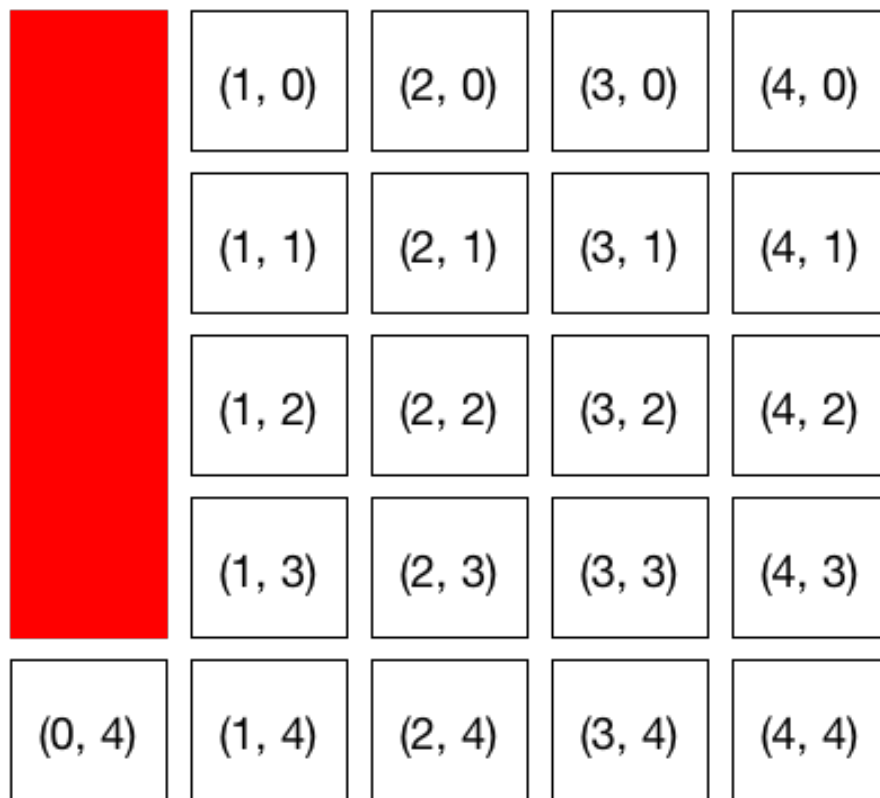


Figure 25: Lijn 3

(0, 0)	(1, 0)	(2, 0)	(3, 0)	(4, 0)
(0, 1)	(1, 1)	(2, 1)		(4, 1)
(0, 2)	(1, 2)	(2, 2)		(4, 2)
(0, 3)	(1, 3)	(2, 3)		(4, 3)
(0, 4)	(1, 4)	(2, 4)		(4, 4)

Figure 26: Lijn 4

- 2. Hierboven staat een lijn. Wat zijn de begin- en eindcoördinaat van die lijn?

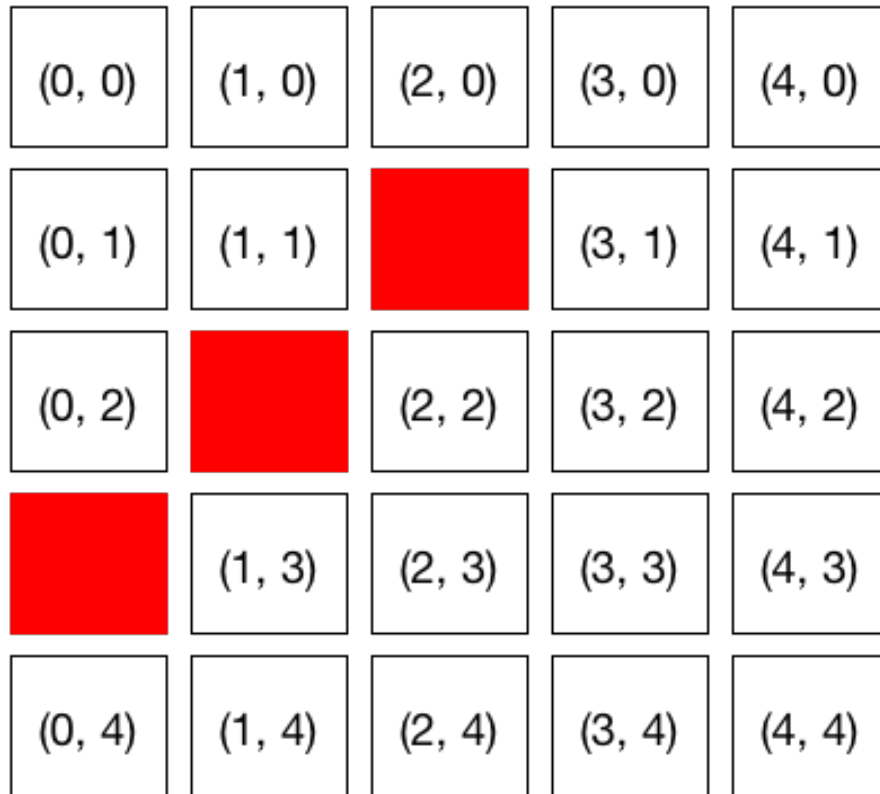


Figure 27: Lijn 5

- 3. Hierboven staat een lijn. Wat zijn de begin- en eindcoördinaat van die lijn?
- 4. Een lijn gaat van coördinaat (0,0) naar (10,0). In welke richting gaat de lijn? Wat is het Processing commando?
- 5. Een lijn gaat van coördinaat (0,0) naar (0,10). In welke richting gaat de lijn? Wat is het Processing commando?
- 6. Een lijn gaat van coördinaat (0,0) naar (10,10). In welke richting gaat de lijn? Wat is het Processing commando?
- 7. Een lijn gaat van coördinaat (30,20) naar (20,20). In welke richting gaat de lijn? Wat is het Processing commando?
- 8. Een lijn gaat van coördinaat (10,20) 20 pixels naar rechts. Welke

coördinaat heeft het eindpunt? Wat is het Processing commando?

- 9. Een lijn gaat van coördinaat (10,30) 10 pixels naar rechtsomhoog. Welke coördinaat heeft het eindpunt? Wat is het Processing commando?

Oplossing

- 1. (0,0) en (0,3)
- 2. (3,1) en (3,4)
- 3. (2,1) en (0,3)
- 4. Van links naar rechts/horizontaal. `line(0,0,10,0)`
- 5. Van onder naar boven/verticaal. `line(0,0,0,10)`
- 6. Schuin/diagonaal. `line(0,0,0,10)`
- 7. Van rechts naar links/horizontaal. `line(30,20,20,20)`
- 8. (30,20). `line(10,20,30,20)`
- 9. (20,20). `line(10,30,20,20)`

Eindopdracht

Hierboven staat een tekening. Maak deze tekening zo goed mogelijk na in Processing.

backGround

Zonder kleur zien games er minder mooi uit.

Een van de allereerste games met kleur heette Moria:

Om goed kleuren te kunnen gebruiken, moeten we leren hoe kleuren werken.

In deze les gaan we leren

- hoe kleuren werken

Zo gaat het eruit zien:

Je hoeft maar weinig te weten, behalve hoe je een mooi programma maakt

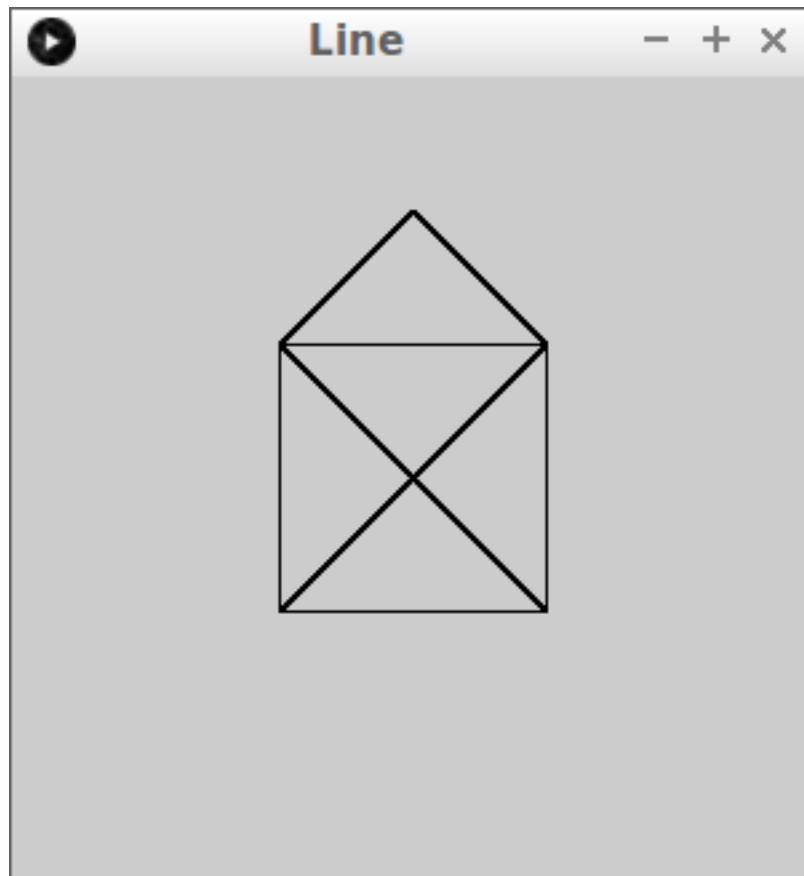


Figure 28: Line



Figure 29: Moria

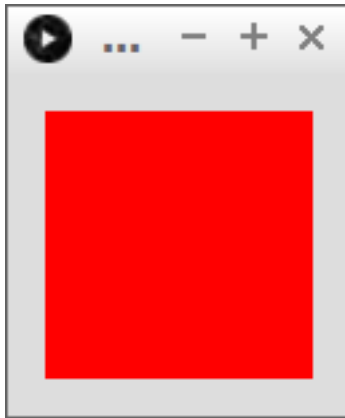


Figure 30: background

Kleuren

Als je goed naar een beeldscherm kijkt, zie je dat elke pixel uit drie lampjes bestaat:

De lampjes hebben de kleuren rood, groen en blauw.

Omdat de lampjes zo klein zijn, ziet ons oog van afstand de menging van deze drie lampjes. Zo zien de lampjes rood en groen er samen uit als geel. Hier zie je hoe de kleuren mengen:

Om wit te krijgen, heb je alledrie de kleuren nodig.

Vragen

- 1. Welke drie kleuren lampjes heeft een pixel?
- 2. Met welke lampjes samen maak je geel?
- 3. Met welke lampjes samen maak je cyaan/lichtblauw?
- 4. Met welke lampjes samen maak je magenta/paars?
- 5. Met welke lampjes samen maak je wit?
- 6. Met welke lampjes samen maak je zwart?
- 7. Met welke lampjes samen maak je grijs?
- 8. Met welke lampjes samen maak je oranje?

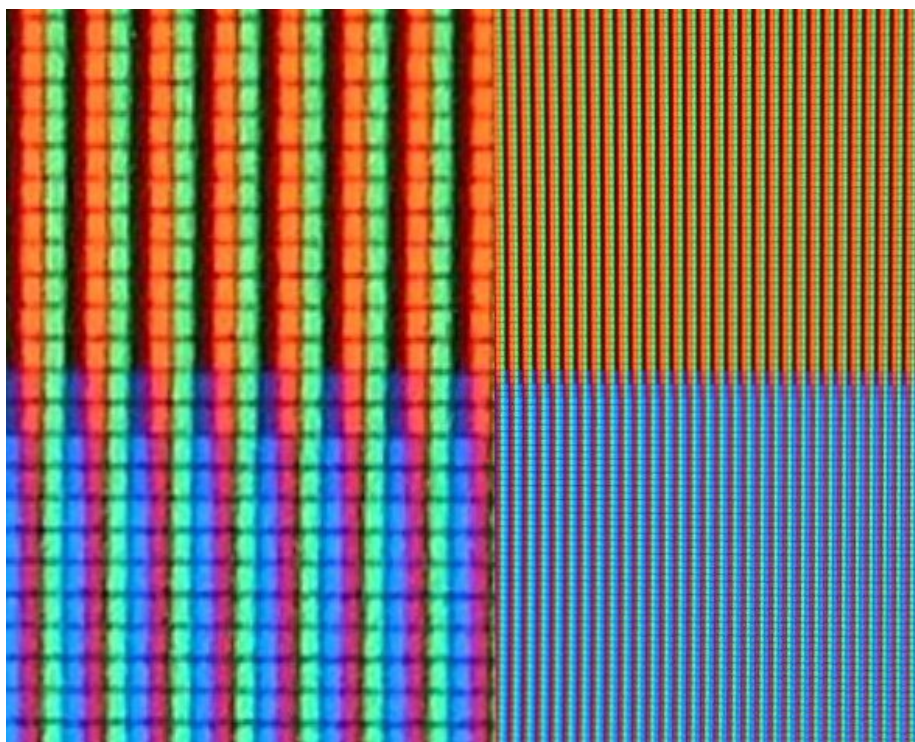


Figure 31: RGB pixels

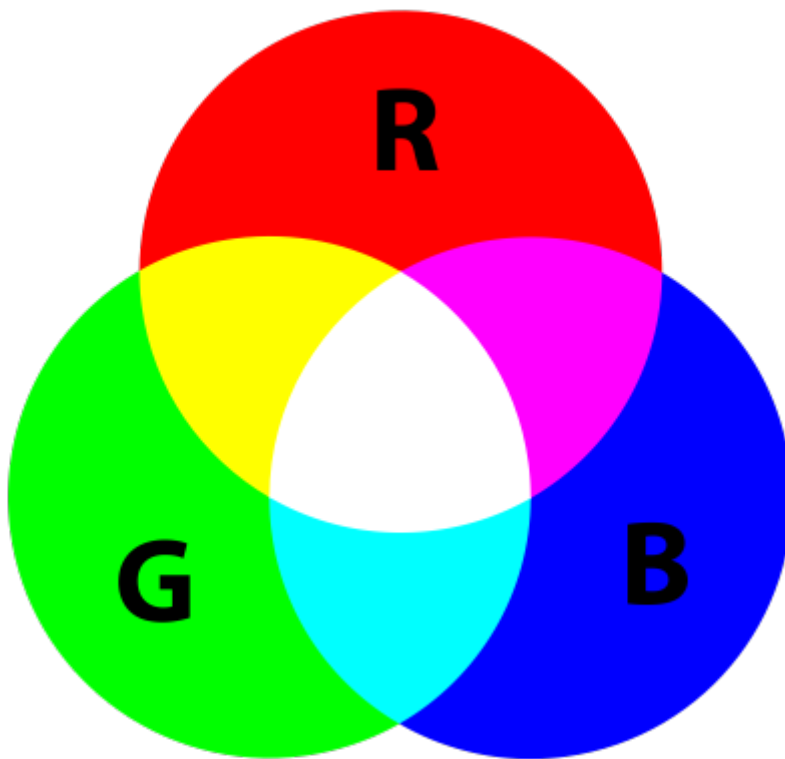


Figure 32: Additieve kleuren

Antwoorden

- 1. Rood, groen en blauw
- 2. Rood en groen
- 3. Groen en blauw
- 4. Rood en blauw
- 5. Rood en groen en blauw
- 6. Met geen lampjes: als alle lampjes uit zijn, is het zwart
- 7. Met rood en groen en blauw, maar dan moeten ze niet op hun hardst branden
- 8. Met rood op z'n hardst en groen op halve kracht

background

In Processing is er een functie om de achtergrond een kleur te geven. Deze functie heet **background**. **background** is Engels voor 'achtergrond'. **background** is een functie die drie getallen nodig heeft. Deze drie getallen bepalen hoe hard de rode, groene en blauwe lampjes gaan schijnen. Deze drie getallen noemen we de RGB waarde. Met het getal nul zeg je dat een lampje uit staat. Met het getal 255 zeg je dat een lampje aan staat. Met getallen tussen nul en 255 kun je het lampje ertussenin laten branden.

Met deze Processing code krijg je een rode achtergrond:

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

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

Opdracht

- 1. Kopieer deze code in Processing en start de code
- 2. Wat is een RGB waarde?
- 3. Wat is de RGB waarde van groen? Maak in Processing een groene achtergrond

- 4. Wat is de RGB waarde van blauw? Maak in Processing een blauwe achtergrond
- 5. Wat is de RGB waarde van geel? Maak in Processing een gele achtergrond
- 6. Wat is de RGB waarde van cyaan/lichtblauw? Maak in Processing een cyane/lichtblauwe achtergrond
- 7. Wat is de RGB waarde van magenta/paars? Maak in Processing een magenta/paarse achtergrond
- 8. Wat is de RGB waarde van wit? Maak in Processing een witte achtergrond
- 9. Wat is de RGB waarde van zwart? Maak in Processing een zwart achtergrond
- 10. Wat is de RGB waarde van grijs? Maak in Processing een grijze achtergrond
- 11. Wat is de RGB waarde van donkerrood? Maak in Processing een donkerrode achtergrond
- 12. Wat is de RGB waarde van oranje? Maak in Processing een oranje achtergrond

Oplossingen

- 1. OK
- 2. De Rood-Groen-Blauw waarde. Dit zijn drie getallen van nul tot en met 255 die bepalen hoe hard de drie kleurenlampjes branden
- 3. `background(0,255,0)`
- 4. `background(0,0,255)`
- 5. `background(255,255,0)`
- 6. `background(0,255,255)`
- 7. `background(255,255,0)`
- 8. `background(255,255,255)`
- 9. `background(0,0,0)`
- 10. `background(128,128,128)`, maar andere getallen tussen 0 en 255 zijn ook goed. Als ze maar alledrie gelijk zijn
- 11. `background(128,0,0)`, maar het eerste getal mag ook een ander getal tussen de 0 en 255 zijn

- 12. `background(255,128,0)`, maar het tweede getal mag ook een ander getal in de buurt van 128 zijn

Eindopdracht

Maak de achtergrond Thijs zijn favoriete kleur.

stroke

Zonder kleur zien games er minder mooi uit.

Een van de allereerste games met kleur heette Moria:



Figure 33: Moria

In deze les gaan we leren

- hoe je een lijn een kleur kan geven.

Zo gaat het eruit zien:

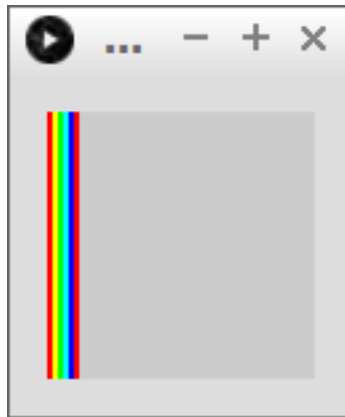


Figure 34: Stroke

Weet je nog niet hoe kleuren werken, ga dan naar de les background

stroke

In Processing is er een functie om lijnen een kleur te geven. Deze functie heet **stroke**. **stroke** is Engels voor ‘(penseel)streek’. **stroke** is een functie die drie getallen nodig heeft. Deze drie getallen zijn de RGB waarden.

Met deze Processing code krijg je een rode lijn:

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

void draw()
{
  stroke(255,0,0);
  line(10,20,30,40);
}
```

Met **stroke** zeg je: ‘vanaf nu wil ik deze lijnkleur’. Hieronder zie je hoe je twee groene en een blauwe lijn tekent:

```
void setup()
{
```

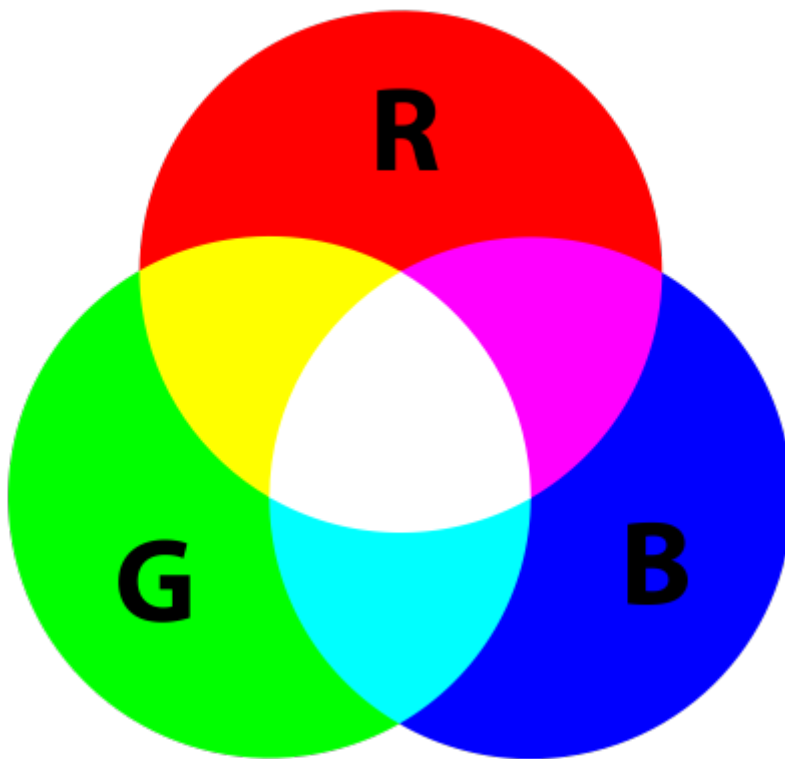



Figure 35: Kleurencirkel

```

    size(100,100);
}

void draw()
{
    stroke(0,255,0);
    line(10,20,30,40);
    line(50,60,70,80);
    stroke(0,0,255);
    line(90,10,20,30);
}

```

Opdracht

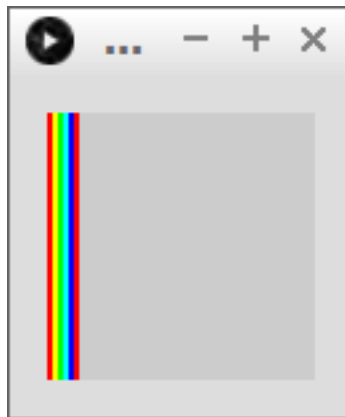


Figure 36: Stroke

- 1. Maak in Processing bovenstaande tekening na. Het venster is honderd bij honderd pixels. Elke kleur is met twee lijnen getekend. De kleuren zijn rood, geel, groen, cyaan, blauw, magenta

Oplossing

```

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

void draw()
{
    stroke(255,0,0);

```

```

line(0,0,0,100);
line(1,0,1,100);
stroke(255,255,0);
line(2,0,2,100);
line(3,0,3,100);
stroke(0,255,0);
line(4,0,4,100);
line(5,0,5,100);
stroke(0,255,255);
line(6,0,6,100);
line(7,0,7,100);
stroke(0,0,255);
line(8,0,8,100);
line(9,0,9,100);
stroke(255,0,0);
line(10,0,10,100);
line(11,0,11,100);
}

```

Eindopdracht

maak nog 4 kleuren bij het voorbeeld hier boven.

Rect

Vierkanten worden veel gebruikt in games.

Hier zie je een van de beroemdste games ooit:

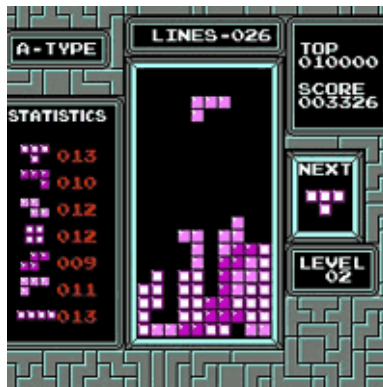


Figure 37: Tetris

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

In deze les gaan we leren

- hoe je lijnen tekent

Zo gaat het eruit zien:

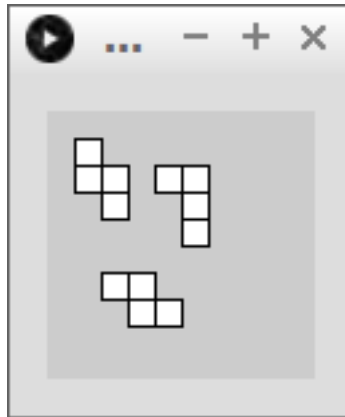


Figure 38: Rect

Kun je nog geen lijnen tekenen? Ga dan naar de les waarin je lijnen leert tekenen

Rechthoeken

Een rechthoek bestaat uit vier lijnen. Om een rechthoek te tekenen, moet je een coördinaat, breedte en hoogte geven.

Om in Processing een rechthoek te tekenen, gebruik je de functie `rect`. De functie `rect` heeft vier getallen nodig. De eerste twee getallen zijn de coördinaat van de linkerbovenhoek van de rechthoek. Het derde getal is de breedte van de rechthoek. Het vierde getal is de hoogte van de rechthoek.

Hier zie je een rechthoek met coördinaat (1,2), een breedte van drie pixels en hoogte van vier pixels:

In Processing teken je deze rechthoek met:

```
rect(1,2,3,4);
```

Hier is nog een rechthoek:

De linkerbovenhoek heeft coördinaat (2,1), hij is vier pixels breed en drie pixels hoog.

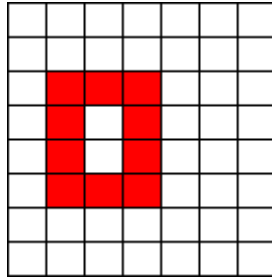


Figure 39: Rechthoek 1

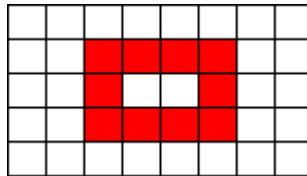


Figure 40: Rechthoek 2

Vragen

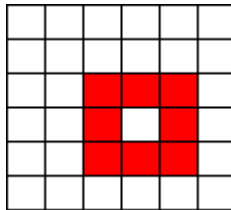


Figure 41: Rechthoek 3

- 1. Hierboven staat een rechthoek. Wat is de coördinaat van de linkerbovenhoek? Hoe breed is de rechthoek? Hoe hoog is de rechthoek?
- 2. Hierboven staat een rechthoek. Wat is de coördinaat van de linkerbovenhoek? Hoe breed is de rechthoek? Hoe hoog is de rechthoek?
- 3. Hierboven staat een rechthoek. Wat is de coördinaat van de linkerbovenhoek? Hoe breed is de rechthoek? Hoe hoog is de rechthoek?
- 4. Een rechthoek heeft als coördinaat (0,0), is twee pixels breed en drie pixels hoog. Wat is het Processing commando?
- 5. Een rechthoek heeft als coördinaat (1,2), is drie pixels breed en vier pixels hoog. Wat is het Processing commando?

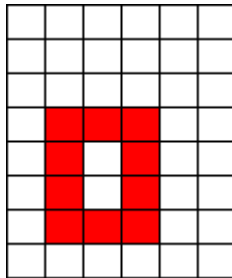


Figure 42: Rechthoek 4

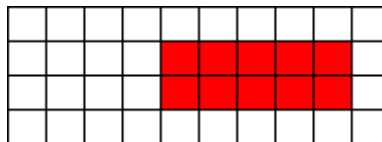


Figure 43: Rechthoek 5

- 6. Een rechthoek heeft als coördinaat (10,20), is dertig pixels breed en veertig pixels hoog. Wat is het Processing commando?

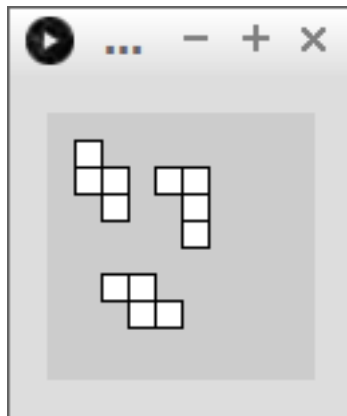


Figure 44: Rect

- 10. Hierboven staat een tekening. Maak deze tekening na in Processing

Oplossing

- 10. Zie hieronder:

```
void setup()
```

```

{
    size(100,100);
}

void draw()
{
    rect(10,10,10,10);
    rect(10,20,10,10);
    rect(20,20,10,10);
    rect(20,30,10,10);

    rect(40,20,10,10);
    rect(50,20,10,10);
    rect(50,30,10,10);
    rect(50,40,10,10);

    rect(20,60,10,10);
    rect(30,60,10,10);
    rect(30,70,10,10);
    rect(40,70,10,10);
}

```

Eindopdracht

Maak een cirkel van rectangles.

Ellipse

Cirkels en ovalen worden veel gebruikt in games.

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

Je kunt een ovaal tekenen met heel veel puntjes, maar de **ellipse** functie werkt gemakkelijker.

In deze les gaan we leren

- hoe je ovalen tekent

Zo gaat het eruit zien:

Kun je nog geen rechthoeken tekenen? Ga dan naar de les waarin je rechthoeken leert tekenen

Ovalen

Een ovaal heeft een middelpunt, breedte en hoogte. Om een ovaal te tekenen, moet je een coördinaat, breedte en hoogte geven.

Om in Processing een ovaal te tekenen, gebruik je de functie `ellipse`. De functie `ellipse` heeft vier getallen nodig. De eerste twee getallen zijn de coördinaat van het midden van de ovaal. Het derde getal is de breedte van de ovaal. Het vierde getal is de hoogte van de ovaal.

Hier zie je een ovaal met middelpunt (3,2), een breedte van vijf pixels en hoogte van drie pixels:

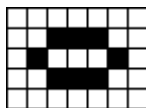


Figure 47: Ovaal 1

In Processing teken je deze ovaal met:

```
ellipse(3,2,5,3);
```

Hier is nog een ovaal:

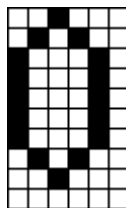


Figure 48: Ovaal 2

Het middelpunt heeft coördinaat (2,4), hij is vijf pixels breed en negen pixels hoog.

Vragen

- 1. Je wilt bovenstaande plaatje namaken. Het venster is 100 pixels breed en 100 pixels hoog. Wat is het middelpunt van de cirkel? Hoe breed is de cirkel? En hoe hoog? Hoe maak je dit in Processing?
- 2. Je wilt bovenstaande plaatje namaken. Het venster is 200 pixels breed en 100 pixels hoog. Wat is het middelpunt van de cirkel? Hoe breed is de cirkel? En hoe hoog? Hoe maak je dit in Processing?

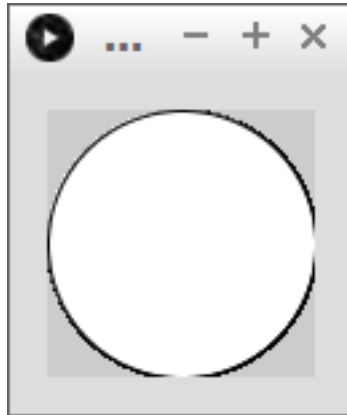


Figure 49: Ovaal 3



Figure 50: Ovaal 4

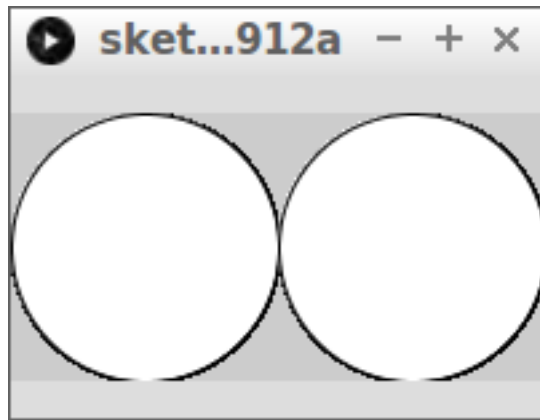


Figure 51: Ovaal 5

- 3. Je wilt bovenstaande plaatje namaken. Het venster is 200 pixels breed en 100 pixels hoog. Wat zijn de middelpunten van de cirkels? Hoe breed zijn de cirkels? En hoe hoog? Hoe maak je dit in Processing?

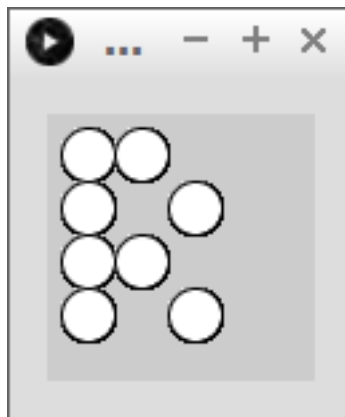


Figure 52: Ellipse

- 4. Hierboven staat een tekening. Maak deze tekening na in Processing

Oplossing

- 1. Zie hieronder:

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

```
void draw() {
  ellipse(50, 50, 100, 100);
}
```

- 2. Zie hieronder:

```
void setup() {
  size(200, 100);
}
```

```
void draw() {
  ellipse(100, 50, 200, 100);
}
```

- 3. Zie hieronder:

```
void setup() {
  size(200, 100);
}
```

```
void draw() {
  ellipse( 50, 50, 100, 100);
  ellipse(150, 50, 100, 100);
}
```

- 4. Zie hieronder:

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

```
void draw()
{
  ellipse(15,15, 20, 20);
  ellipse(15,35, 20, 20);
  ellipse(15,55, 20, 20);
  ellipse(15,75, 20, 20);

  ellipse(35, 15, 20, 20);
  ellipse(55, 35, 20, 20);
  ellipse(35, 55, 20, 20);
  ellipse(55, 75, 20, 20);
}
```

Eindopdracht

Maak een cirkel die door twaalf kleinere cirkels volledig omringd en afgesloten is.

fill

Een rechthoek of een ovaal kan ook ingekleurd worden.

Een van de beroemdste games ooit heeft bijvoorbeeld veel ingekleurde vierkanten:

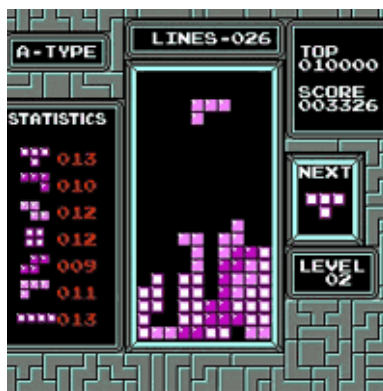


Figure 53: Tetris

In deze les gaan we leren

- hoe we de invulkleur van vierkanten en ovaal instellen

Zo gaat het eruit zien:

Weet je nog niet hoe je een lijn een kleur kan geven? Ga dan naar stroke

Kleuren

De lampjes hebben de kleuren rood, groen en blauw.

Omdat de lampjes zo klein zijn, ziet ons oog van afstand de menging van deze drie lampjes. Zo zien de lampjes rood en groen er samen uit als geel. Hier zie je hoe de kleuren mengen:

Om wit te krijgen, heb je alledrie de kleuren nodig.

Vragen

1. Welke drie kleuren lampjes heeft een pixel?

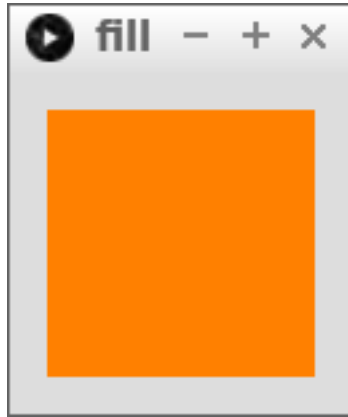


Figure 54: Fill

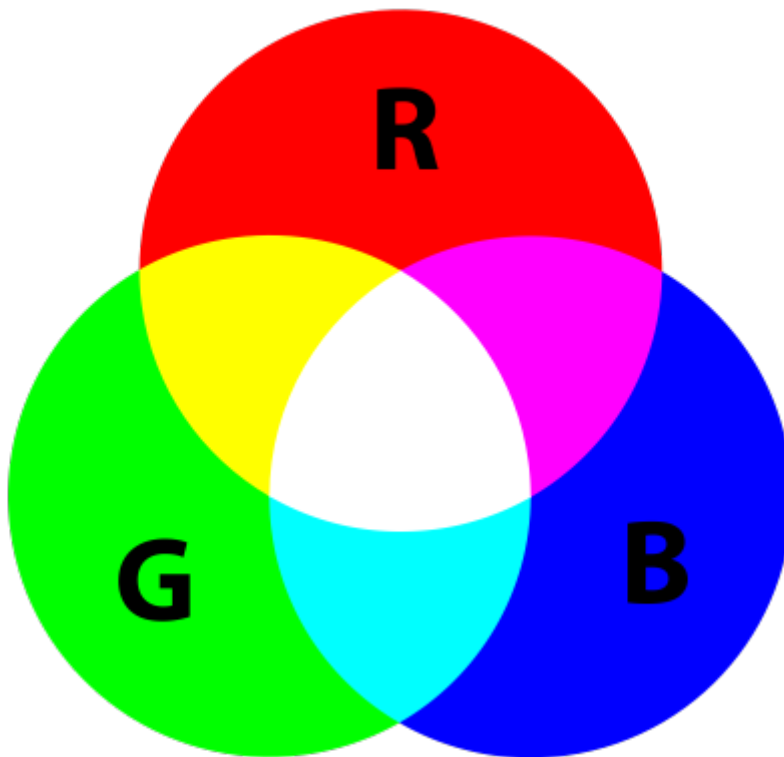


Figure 55: Additieve kleuren

- 2. Met welke lampjes samen maak je geel?
- 3. Met welke lampjes samen maak je cyaan/lichtblauw?
- 4. Met welke lampjes samen maak je magenta/paars?
- 5. Met welke lampjes samen maak je wit?
- 6. Met welke lampjes samen maak je zwart?
- 7. Met welke lampjes samen maak je grijs?
- 8. Met welke lampjes samen maak je oranje?

Antwoorden

- 1. Rood, groen en blauw
- 2. Rood en groen
- 3. Groen en blauw
- 4. Rood en blauw
- 5. Rood en groen en blauw
- 6. Met geen lampjes: als alle lampjes uit zijn, is het zwart
- 7. Met rood en groen en blauw, maar dan moeten ze niet op hun hardst branden
- 8. Met rood op z'n hardst en groen op halve kracht

background

In Processing is er een functie om de achtergrond een kleur te geven. Deze functie heet **background**. **background** is Engels voor 'achtergrond'. **background** is een functie die drie getallen nodig heeft. Deze drie getallen bepalen hoe hard de rode, groene en blauwe lampjes gaan schijnen. Deze drie getallen noemen we de RGB waarde. Met het getal nul zeg je dat een lampje uit staat. Met het getal 255 zeg je dat een lampje aan staat. Met getallen tussen nul en 255 kun je het lampje ertussenin laten branden.

Met deze Processing code krijg je een rode achtergrond:

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

```
void draw()
```

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

Opdracht

- 1. Kopieer deze code in Processing en start de code
- 2. Wat is een RGB waarde?
- 3. Wat is de RGB waarde van groen? Maak in Processing een groene achtergrond
- 4. Wat is de RGB waarde van blauw? Maak in Processing een blauwe achtergrond
- 5. Wat is de RGB waarde van geel? Maak in Processing een gele achtergrond
- 6. Wat is de RGB waarde van cyaan/lichtblauw? Maak in Processing een cyane/lichtblauwe achtergrond
- 7. Wat is de RGB waarde van magenta/paars? Maak in Processing een magenta/paarse achtergrond
- 8. Wat is de RGB waarde van wit? Maak in Processing een witte achtergrond
- 9. Wat is de RGB waarde van zwart? Maak in Processing een zwart achtergrond
- 10. Wat is de RGB waarde van grijs? Maak in Processing een grijze achtergrond
- 11. Wat is de RGB waarde van donkerrood? Maak in Processing een donkerrode achtergrond
- 12. Wat is de RGB waarde van oranje? Maak in Processing een oranje achtergrond

Oplossingen

- 1. OK
- 2. De Rood-Groen-Blauw waarde. Dit zijn drie getallen van nul tot en met 255 die bepalen hoe hard de drie kleurenlampjes branden
- 3. background(0,255,0)
- 4. background(0,0,255)

- 5. `background(255,255,0)`
- 6. `background(0,255,255)`
- 7. `background(255,255,0)`
- 8. `background(255,255,255)`
- 9. `background(0,0,0)`
- 10. `background(128,128,128)`, maar andere getallen tussen 0 en 255 zijn ook goed. Als ze maar alledrie gelijk zijn
- 11. `background(128,0,0)`, maar het eerste getal mag ook een ander getal tussen de 0 en 255 zijn
- 12. `background(255,128,0)`, maar het tweede getal mag ook een ander getal in de buurt van 128 zijn

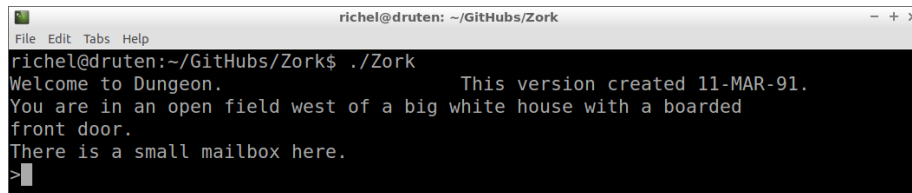
Eindopdracht

Gebruik `fill` om 5 vormen kleur te geven. Maar het moeten wel 5 verschillende kleuren zijn.

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:



```
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

Kun je nog geen puntjes tekenen? Ga dan naar de les waarin je puntjes leert tekenen

Kun je nog geen vlakken inkleuren? Ga dan naar de les ‘fill’

Tekst

Hier zie je de tekst ‘Hallo’ laat zetten op coördinaat (10,20):

```
text("Hallo", 10, 20);
```

Let op dat de tekst tussen dubbele apostroffen (") moet.

`text` kan ook rekenen!

Hier plus en min:

```
text(128 + 64, 10, 20);
```

```
text(128 - 64, 10, 20);
```

Hier een keersom:

```
text(16 * 16, 10, 20);
```

Hier een deelsom:

```
text(256 / 16, 10, 20);
```

Tekstgrootte kun je aanpassen met

```
textSize(32);
```

Tekstkleur kun je aanpassen met `fill`:

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

Opdracht

Zet de tekst `I love you 4 ever` op het scherm, waarbij:

- de coördinaat is (10, 30)
- de tekst heeft de normale kleur, dit is wit
- de tekst heeft de normale grootte, dit is 8 pixels

Oplossing

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

```
void draw()
{
  text("I love you 4 ever", 10, 30);
}
```

Opdracht

Zet de tekst `I love you 4 ever` op het scherm, waarbij:

- de coördinaat is (10, 30)
- de tekst zwart is (tip: gebruik `fill`)
- de tekst heeft de normale grootte, dit is 8 pixels

Oplossing

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

void draw()
{
  fill(0, 0, 0);
  text("I love you 4 ever", 10, 30);
}
```

Opdracht

Zet de tekst `I love you 4 ever` op het scherm, waarbij:

- de coördinaat is (10, 30)
- de tekst zwart is
- de tekst is 32 pixels groot (tip: gebruik `textSize`)

Oplossing

```
void setup()
{
  size(300,300);
  textSize(32);
}
```

```
void draw()
{
  fill(0, 0, 0);
  text("I love you 4 ever", 10, 30);
}
```

Opdracht

Zet de tekst I love you 4 ever op het scherm, waarbij:

- de tekst is 32 pixels groot (tip: gebruik `textSize`)
- alle woorden zwart zijn, behalve `love`, die rood is
- de 4 is de uitkomst van een berekening, bijvoorbeeld $2 + 2$ (maar hoe moeilijker de berekening, hoe stoerder)

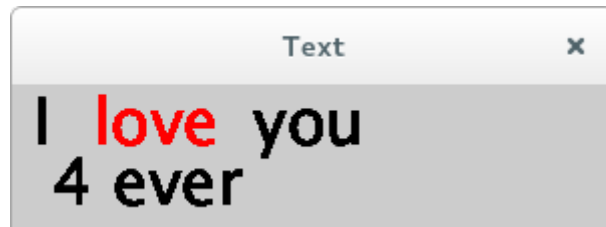


Figure 57: Text

Oplossing

```
void setup()
{
  size(300,300);
  textSize(32);
}

void draw()
{
  fill(0, 0, 0);
  text("I", 10, 30);
  fill(255, 0, 0);
  text("love", 40, 30);
  fill(0, 0, 0);
  text("you", 120, 30);
  text(256 / 64, 20, 60);
  text("ever", 50, 60);
}
```

Eindopdracht

Zet je eigen naam mooi op het scherm. Een voorbeeld:

R
I
C
H
E
L

Figure 58: Eindopdracht text