DE JONGE ONDERZOEKERS
Kijk, doe, ontdek !

Figure 1: Boek 8: botsen

#	Omschrijving
29	Muis binnen vierkant
30	Muis binnen cirkel
31	Cirkels botsen
32	Vierkanten botsen

Contents

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

Het is nog een beetje een slordig boek. Er zitten tiepvauten in en de opmaak is ni`et altijd` *even mo*oi.

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.

Muis binnen vierkant

In deze les gaan we leren hoe je kunt kijken of de muiscursor binnen een vierkant valt

Muis binnen vierkant: opdracht 1

Type deze code over:

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

void draw()
{
  fill(255, 255, 255);
  if (mouseX > 25)
  {
    fill(255, 0, 0);
  }
  rect(25, 50, 75, 100);
}
```

Wat zie je? Wanneer wordt het vierkant rood?

Muis binnen vierkant: oplossing 1

Het vierkant wordt rood als je de muiscursor meer dan 25 pixels naar rechts beweegt.

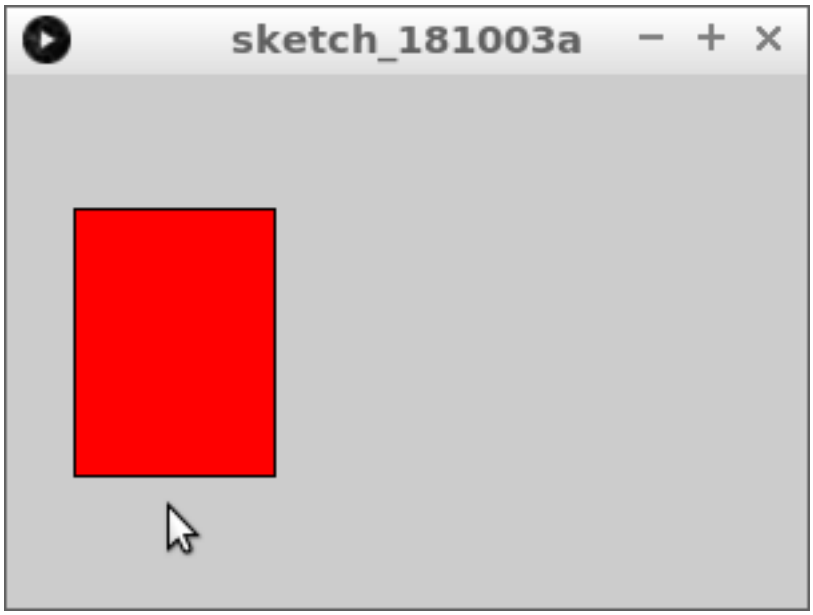


Figure 3:

Muis binnen vierkant: opdracht 2

Verander de code zo dat het vierkant rood wordt als je met de muis links bent van de rechterkant van het vierkant.

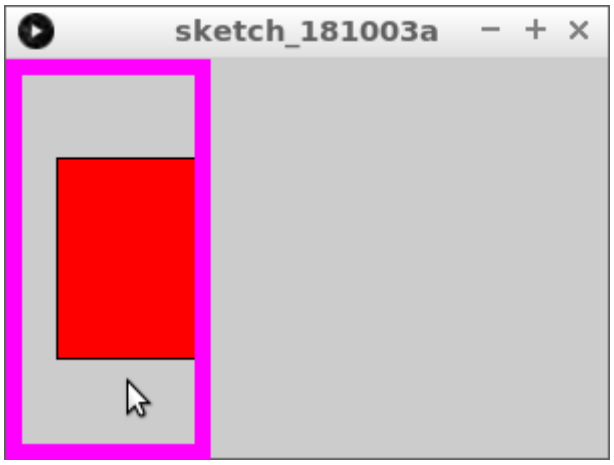


Figure 4:

	
`if (x < 200) { }`	'Lieve computer, als `x` kleiner is dan 200, doe dan wat tussen accolades staat.'

Muis binnen vierkant: oplossing 2

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

void draw()
{
  fill(255, 255, 255);
  if (mouseX < 100)
  {
    fill(255, 0, 0);
  }
  rect(25, 50, 75, 100);
}
```

Muis binnen vierkant: opdracht 3

We gaan nu de `if` statements combineren!

Vervang de `if` die je nu hebt door dit:

```
  if (mouseX > 25 && mouseX < 100)
  {
    fill(255, 0, 0);
  }
```

De `&&` lees je als 'en'

Muis binnen vierkant: oplossing 3

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

void draw()
{
  fill(255, 255, 255);
  if (mouseX > 25 && mouseX < 100)
  {
    fill(255, 0, 0);
  }
  rect(25, 50, 75, 100);
}
```

Muis binnen vierkant: opdracht 4

Maak nu het vierkant rood als de muiscursor onder de bovenkant van het vierkant is.

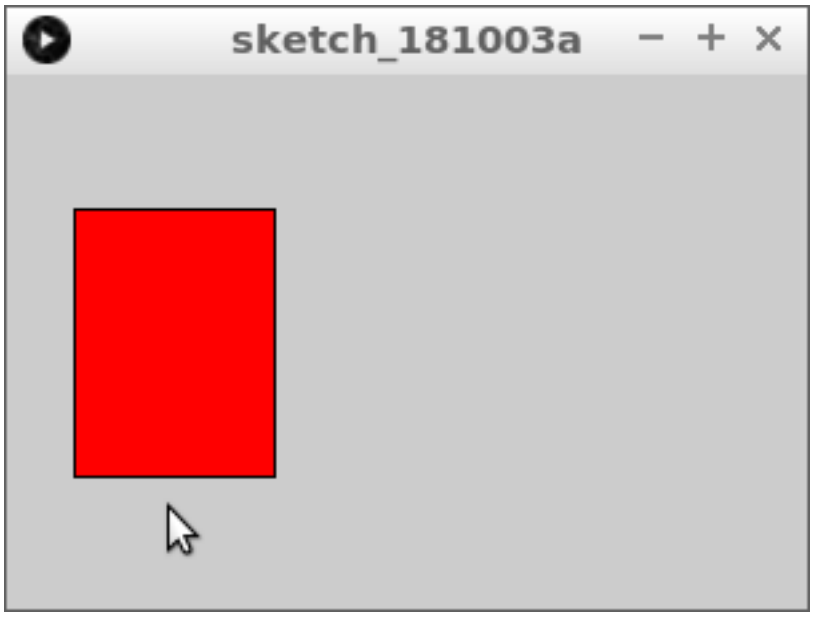


Figure 5:

Muis binnen vierkant: oplossing 4

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

void draw()
{
  fill(255, 255, 255);
  if (mouseY > 50)
  {
    fill(255, 0, 0);
  }
  rect(25, 50, 75, 100);
}
```

Muis binnen vierkant: Eindopdracht

Maak het vierkant rood als de muiscursor in het vierkant is.

Muis binnen cirkel

In deze les gaan we leren hoe je kunt kijken of de muiscursor binnen een cirkel valt

Muis binnen cirkel: opdracht 1

Type deze code over:

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

void draw()
{
  fill(255, 255, 255);
  if (dist(mouseX, mouseY, 150, 100) < 40)
  {
    fill(255, 0, 0);
  }
  ellipse(150, 100, 80, 80);
}
```

Wat zie je? Wanneer wordt de cirkel rood?

Muis binnen cirkel: oplossing 1

De cirkel wordt rood als je de muiscursor in de cirkel beweegt.

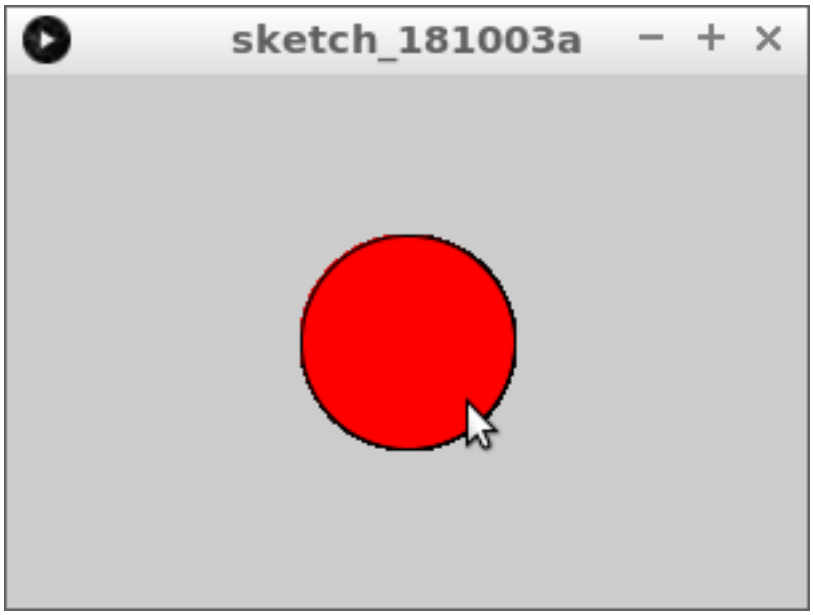


Figure 6:

Muis binnen cirkel: opdracht 2

Maak een variabele aan (bovenaan): `float cirkel_midden_x = 150;`. Vervang de andere `150`-en in de code door `cirkel_midden_x`.

Muis binnen cirkel: oplossing 2

```
float cirkel_midden_x = 150;

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

void draw()
{
  fill(255, 255, 255);
  if (dist(mouseX, mouseY, cirkel_midden_x, 100) < 40)
  {
    fill(255, 0, 0);
  }
  ellipse(cirkel_midden_x, 100, 80, 80);
}
```

Muis binnen cirkel: opdracht 3

Voeg de volgende regel toe aan de `setup` functie:

```
cirkel_midden_x = random(width);
```

Wat zie je? Start het programma meerdere keren!

Muis binnen cirkel: oplossing 3

```
float cirkel_midden_x = 150;

void setup()
{
  size(300, 200);
  cirkel_midden_x = random(width);
}

void draw()
{
  fill(255, 255, 255);
  if (dist(mouseX, mouseY, cirkel_midden_x, 100) < 40)
  {
    fill(255, 0, 0);
  }
  ellipse(cirkel_midden_x, 100, 80, 80);
}
```

Muis binnen cirkel: opdracht 4

Maak een nieuwe variabele aan: `cirkel_midden_y`. In `setup` krijgt deze een willekeurig getal tot `height`. `cirkel_midden_y` vervangt de `100`s.

Muis binnen cirkel: oplossing 4

```
float cirkel_midden_x = 150;
float cirkel_midden_y = 100;

void setup()
{
  size(300, 200);
  cirkel_midden_x = random(width);
  cirkel_midden_y = random(height);
}

void draw()
{
  fill(255, 255, 255);
  if (dist(mouseX, mouseY, cirkel_midden_x, cirkel_midden_y) < 40)
  {
    fill(255, 0, 0);
  }
  ellipse(cirkel_midden_x, cirkel_midden_y, 80, 80);
}
```

Muis binnen cirkel: opdracht 5

Maak een nieuwe variabele aan: `cirkel_doorsnede`. In `setup` krijgt deze een willekeurig getal tot `150`. `cirkel_doorsnede` vervangt de `80`s.

Wat gaat er mis?

Muis binnen cirkel: oplossing 5

```
float cirkel_midden_x = 150;
float cirkel_midden_y = 100;
float cirkel_doorsnede = 100;

void setup()
{
  size(300, 200);
  cirkel_midden_x = random(width);
  cirkel_midden_y = random(height);
  cirkel_doorsnede = random(150);
}

void draw()
{
  fill(255, 255, 255);
  if (dist(mouseX, mouseY, cirkel_midden_x, cirkel_midden_y) < 40)
  {
    fill(255, 0, 0);
  }
  ellipse(cirkel_midden_x, cirkel_midden_y, cirkel_doorsnede, cirkel_doorsnede);
}
```

Muis binnen cirkel: eindopdracht

Maak een nieuwe variabele aan: `cirkel_straal`. In `setup` wordt deze `cirkel_doorsnede / 2`. `cirkel_straal` vervangt de `40`s.

Cirkels botsen

In deze les gaan we leren hoe je kunt meten of twee cirkels botsen

Cirkels botsen: opdracht 1

Type deze code over:

```
float x1 = 150;
float y1 = 100;
float d1 = 180;
float r1 = d1 / 2;

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

void draw()
{
  fill(255, 255, 255);
  if (dist(mouseX, mouseY, x1, y1) < r1)
  {
    fill(255, 0, 0);
  }
  ellipse(x1, y1, d1, d1);
}
```

Wat zie je?

Cirkels botsen: oplossing 1

Een cirkel. Als je met je muiscursor erin gaat, dan wordt deze rood.

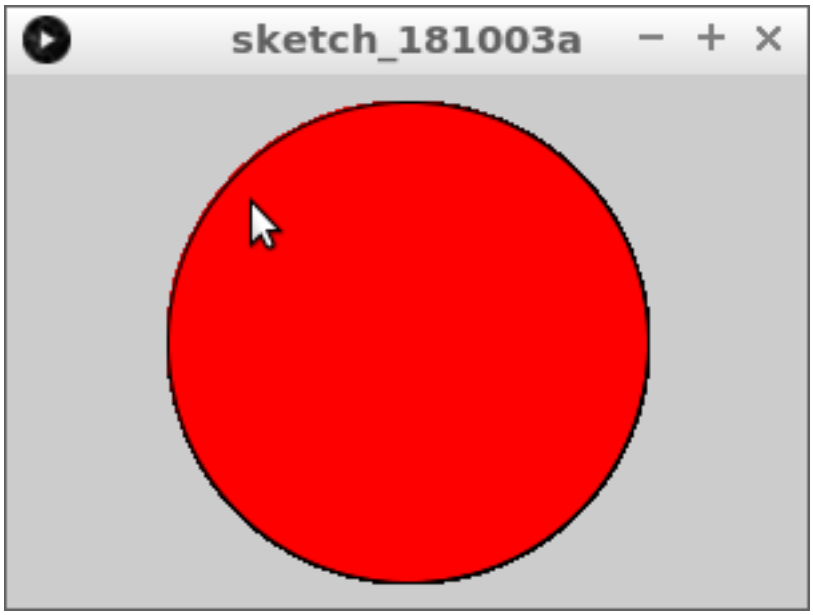


Figure 7:

Cirkels botsen: opdracht 2

Voeg een tweede cirkel toe. Maak vier nieuwe variabelen:

```
float x2 = 30;
float y2 = 100;
float d2 = 60;
float r2 = d1 / 2;
```

Teken een tweede cirkel met als middelpunt `(x2, y2)` en een breedte en hoogte van `d2`.

Cirkels botsen: oplossing 2

```
// ...
float x2 = 30;
float y2 = 100;
float d2 = 60;
float r2 = d2 / 2;

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

void draw()
{
  // ...
  ellipse(x2, y2, d2, d2);
}
```

Wat zie je?

Cirkels botsen: opdracht 3

Voeg toe aan de `draw` functie:

```
x2 = x2 + random(-1, 1);
y2 = y2 + random(-1, 1);
```

Wat zie je?

Cirkels botsen: oplossing 3

Je ziet de kleine cirkel bewegen.

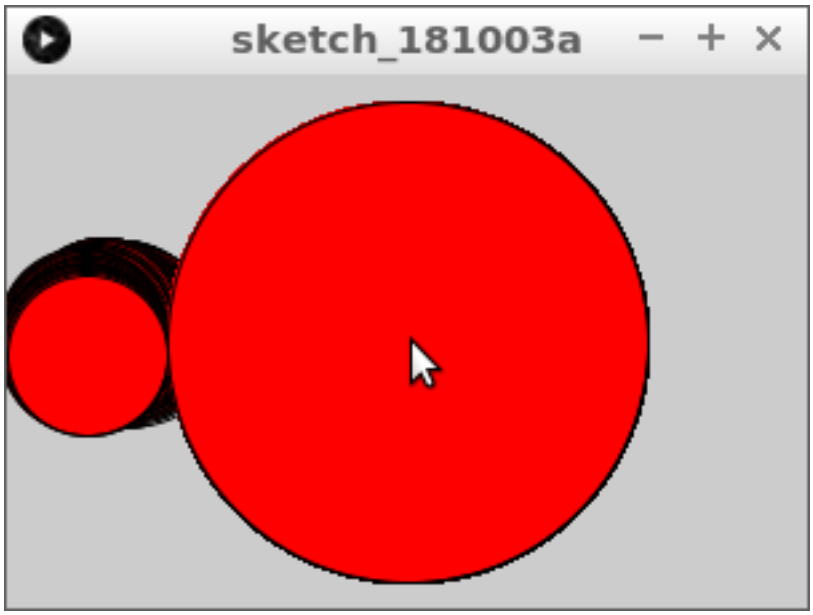


Figure 8:

Cirkels botsen: opdracht 4

Verander aan de `draw` functie het `if` statement naar:

```
if (dist(x1, y1, x2, y2) < r1 + r2)
{
  fill(255, 0, 0);
}
```

Wat zie je?

Cirkels botsen: oplossing 4

Je ziet de cirkels rood worden als ze botsen:

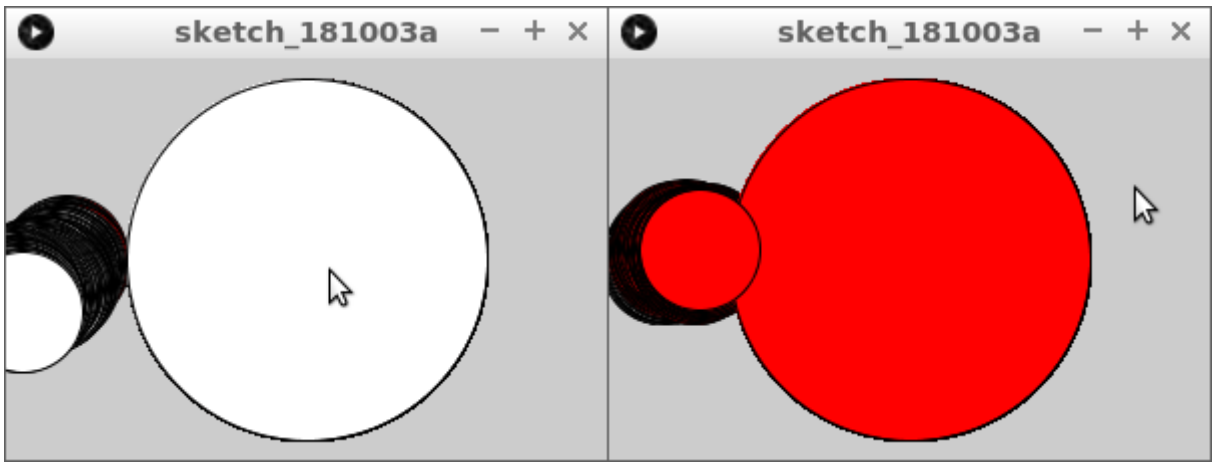
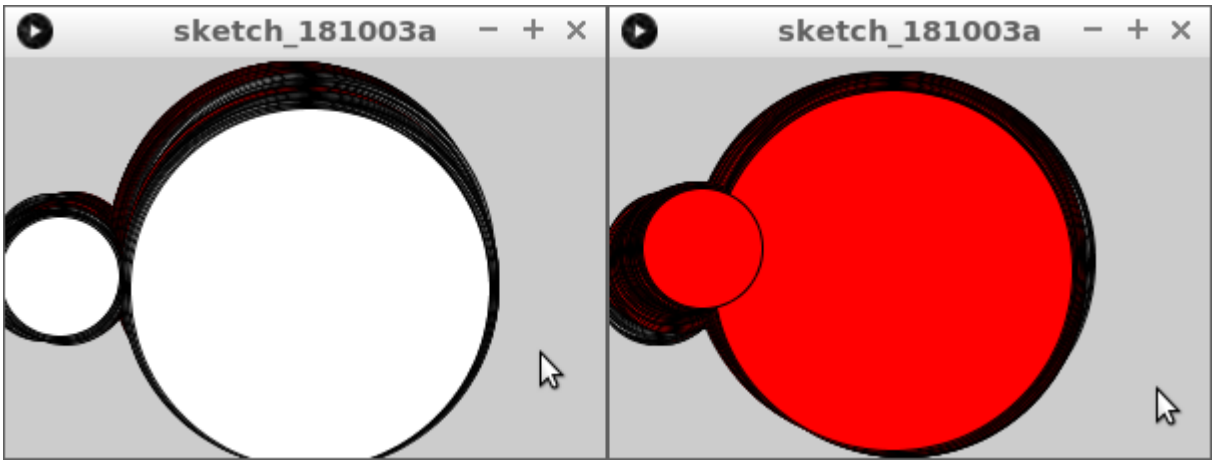


Figure 9:

Cirkels botsen: Eindopdracht

Laat ook de grote cirkel bewegen. Als ze botsen, moeten ze rood worden.



Vierkanten botsen

In deze les gaan we leren hoe je kunt meten of twee vierkanten botsen

Vierkanten botsen: opdracht 1

Type deze code over:

```
float x1 = 150;
float y1 = 100;
float w1 = 75;
float h1 = 50;
float x2 = 75;
float y2 = 50;
float w2 = 75;
float h2 = 50;

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

void draw()
{
  x2 = mouseX;
  y2 = mouseY;
  fill(255, 255, 255);
  if (x2 + w2 > x1)
  {
    fill(255, 0, 0);
  }
  rect(x1, y1, w1, h1);
  rect(x2, y2, w2, h2);
}
```

Wat zie je?

Vierkanten botsen: oplossing 1

Twee vierkanten. Een vierkant volgt de muis. Als het bewegende vierkant de stilstaande raakt, wordt deze soms rood.

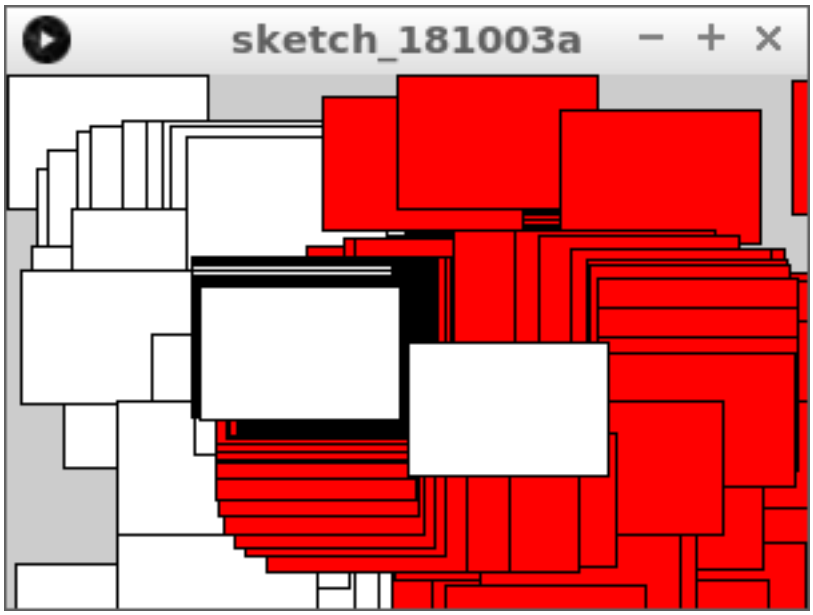


Figure 10:

Vierkanten botsen: opdracht 2

Verander de `if` naar:

```
if (x2 + w2 > x1 && x2 + w2 < x1 + w1)
```

Wat zie je?

Vierkanten botsen: oplossing 2

De vierkanten botsen nu horizontaal.

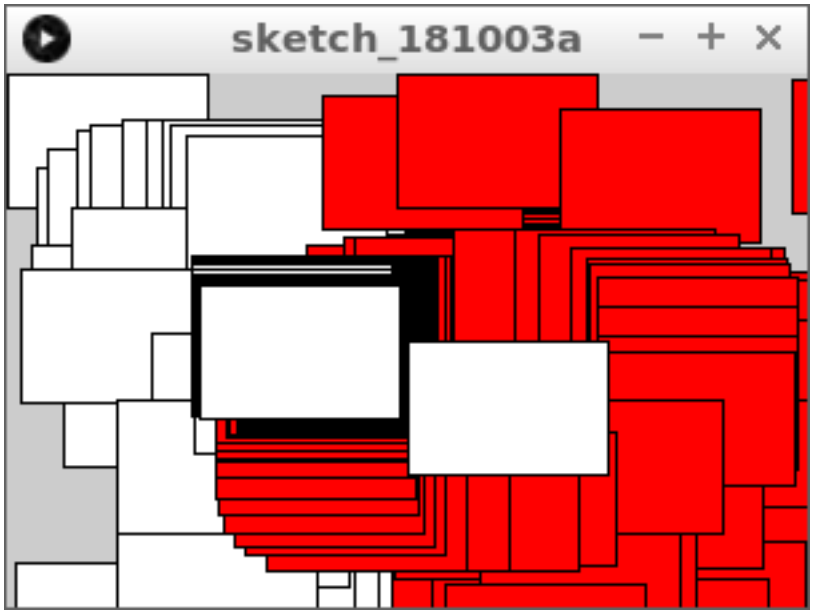


Figure 11:

```
// ...

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

void draw()
{
  // ...
  if (x2 + w2 > x1 && x2 < x1 + w1)
  {
    fill(255, 0, 0);
  }
  rect(x1, y1, w1, h1);
  rect(x2, y2, w2, h2);
}
```

Vierkanten botsen: opdracht 3

Verander de `if` naar:

```
if (x2 + w2 > x1 && x2 < x1 + w1 && y2 + h2 > y1)
```

Wat zie je?

Vierkanten botsen: oplossing 3

De vierkanten botsen nu juist behalve bovenin.

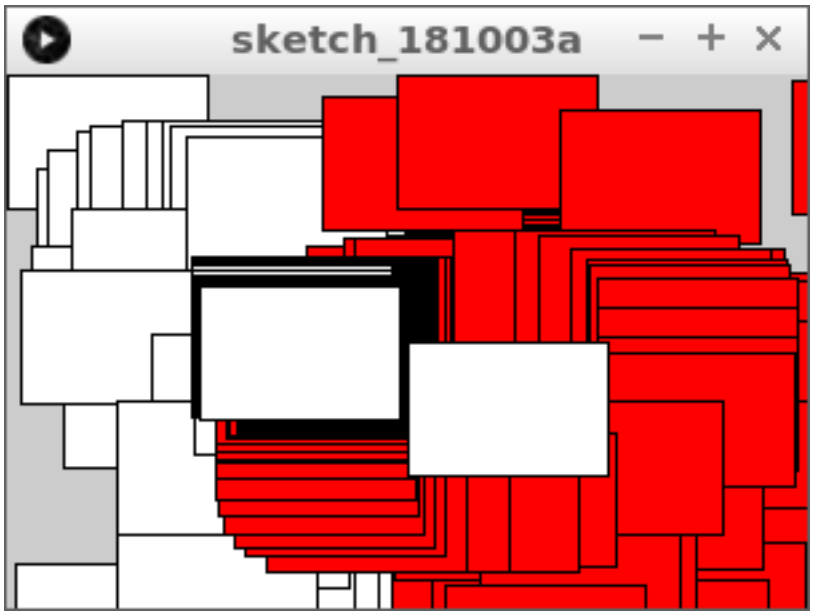


Figure 12:

Vierkanten botsen: eindopdracht

Maak de `if` nog langer, zodat de vierkanten juist botsen.