



Figure 1: Boek 6: 3D 1

#	Omschrijving
21	3D 1: bal bewegen
22	3D 2: bal besturen
23	3D 3: bal en kubus
24	3D 4: belichting

Contents

Voorwoord	1
3D les 1: bal naar rechts	2
3D les 2: bal besturen	9
3D les 3: bal en kubus bewegen	15
3D les 4: belichting	21

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.

3D les 1: bal naar rechts

In deze les gaan we 3D programmeren.

3D 1: opdracht 1

Type deze code over:

```
float x = 0;

void setup()
{
  size(300, 200, P3D);
  noStroke();
}

void draw()
{
  background(255, 255, 255);
  translate(x, 100, 0);
  lights();
  sphere(20);
  x = x + 1;
}
```

Wat zie je?

3D 1: oplossing 1



Figure 3: 3D 1: oplossing 1

Je ziet een bal van links naar rechts bewegen.

Als je goed kijkt zie je dat het witste gedeelte van de bal ook van plek verandert.

3D 1: opdracht 2

- Laat de bal nu naar rechtsonder gaan
- Maak een nieuwe variabele, bijvoorbeeld y

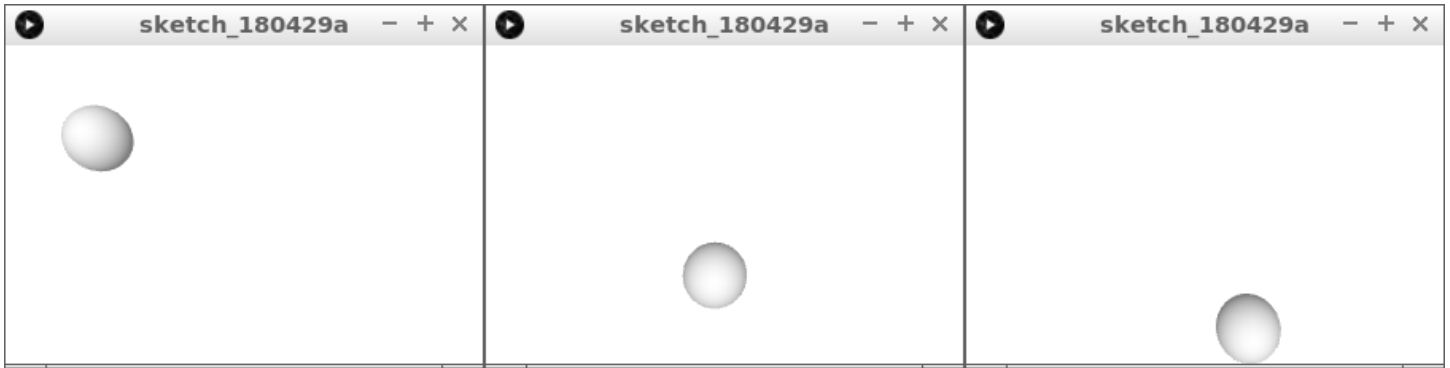


Figure 4: 3D 1: opdracht 2

3D 1: oplossing 2

```
float x = 0;
float y = 0;

void setup()
{
  size(300, 200, P3D);
  noStroke();
}

void draw()
{
  background(255, 255, 255);
  translate(x, y, 0);
  lights();
  sphere(20);
  x = x + 1;
  y = y + 1;
}
```

3D 1: opdracht 3

- Laat de bal nu naar voren gaan
- Maak een nieuwe variabele, bijvoorbeeld z

Dit ziet er misschien anders uit dan je verwacht!

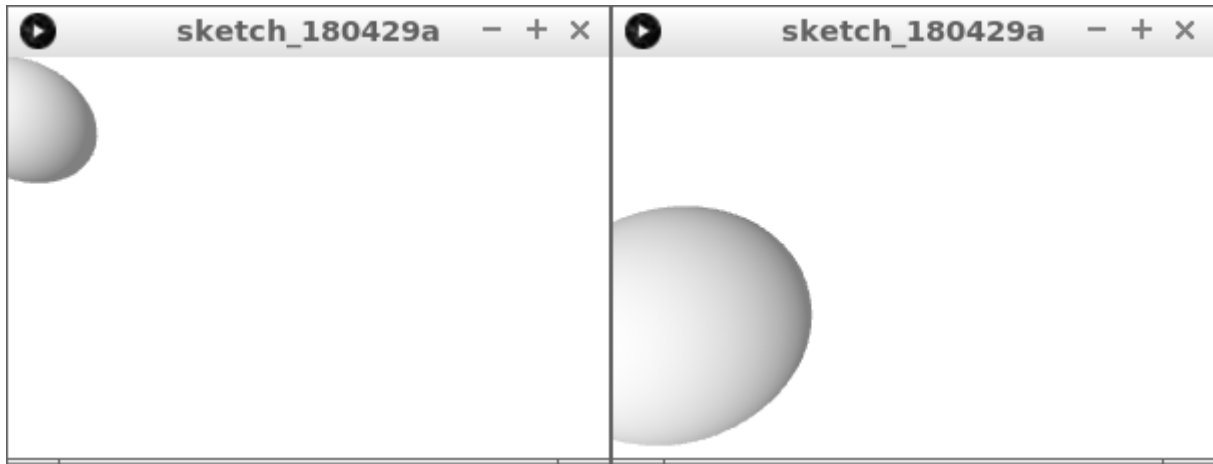


Figure 5: 3D 1: opdracht 3

3D 1: oplossing 3

```
float x = 0;
float y = 0;
float z = 0;

void setup()
{
  size(300, 200, P3D);
  noStroke();
}

void draw()
{
  background(255, 255, 255);
  translate(x, y, z);
  lights();
  sphere(20);
  x = x + 1;
  y = y + 1;
  z = z + 1;
}
```

3D 1: Eindopdracht

- Maak de grootte van de bol gelijk aan de x
- Haal noStroke weg

Dit ziet er misschien anders uit dan je verwacht!

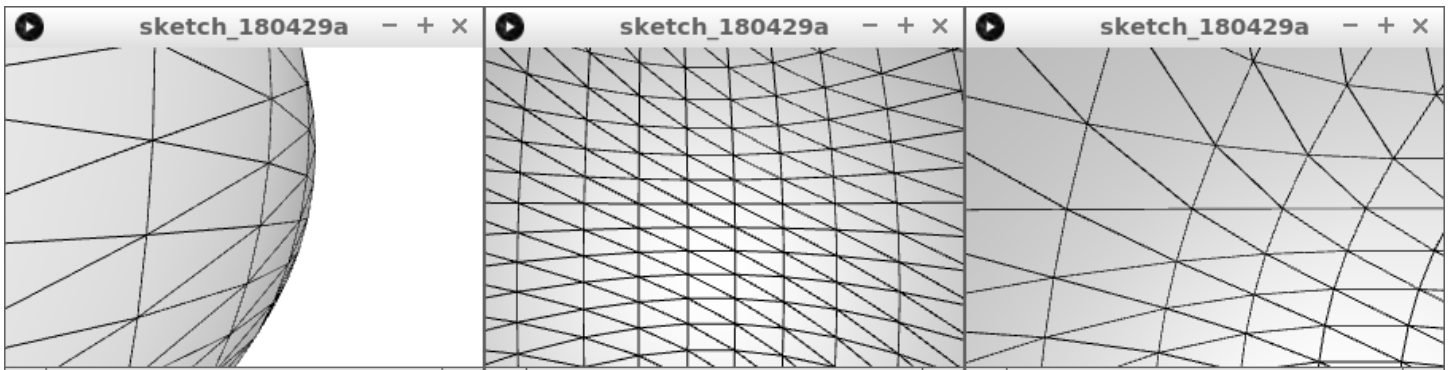


Figure 6: 3D 1: Eindopdracht

3D les 2: bal besturen

In deze les gaan we een 3D bal laten bewegen.

3D 2: opdracht 1

Type deze code over:

```
float x = 0;

void setup()
{
  size(300, 200, P3D);
  noStroke();
  x = width / 2;
}

void draw()
{
  background(255, 255, 255);
  lights();
  translate(x, height / 2, 0);
  sphere(20);
  if (keyPressed)
  {
    if (key == 'a') x = x - 1;
    if (key == 'd') x = x + 1;
  }
}
```

Wat zie je?

3D 2: oplossing 1

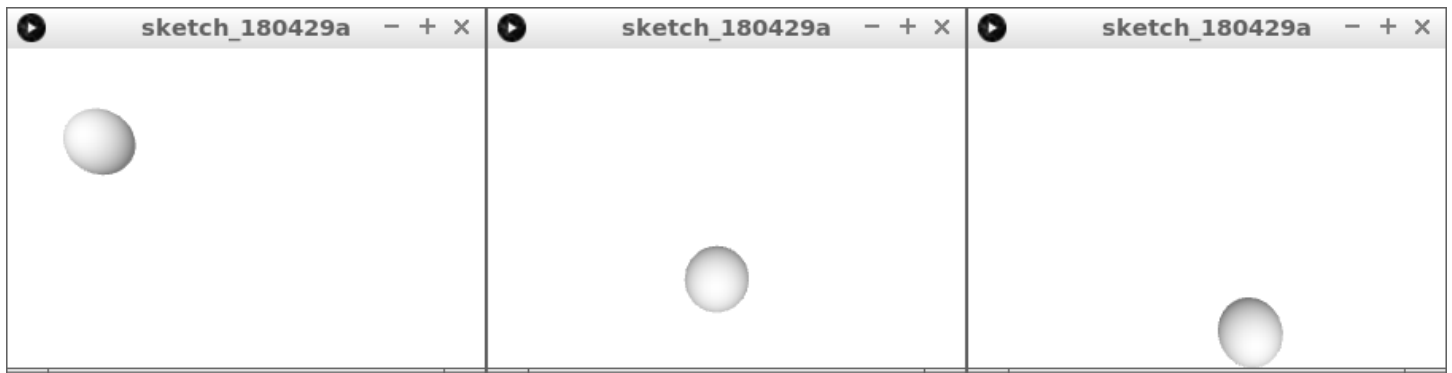


Figure 7: 3D 2: oplossing 1

Je ziet een bal die je met **a** en **d** naar links en rechts kunt bewegen.

Als je goed kijkt zie je dat het witste gedeelte van de bal ook van plek verandert.

3D 2: opdracht 2

- Zorg dat de bal nu ook naar onder en boven kan gaan, met de toetsen **w** en **s**
- Maak een nieuwe variabele, bijvoorbeeld **y**, met als beginwaarde **height / 2**

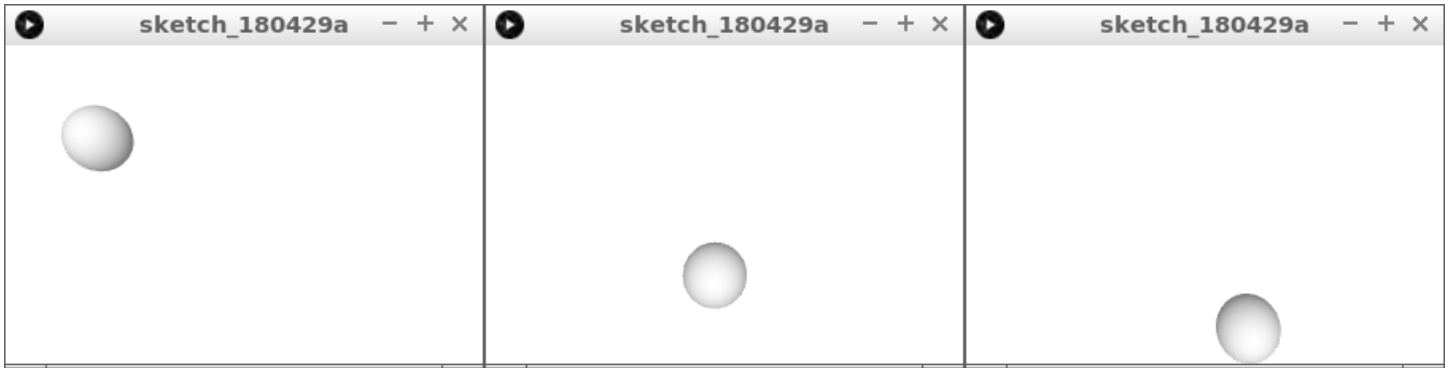


Figure 8: 3D 2: opdracht 2

3D 2: oplossing 2

```
float x = 0;
float y = 0;

void setup()
{
  size(300, 200, P3D);
  noStroke();
}

void draw()
{
  background(255, 255, 255);
  lights();
  translate(x, y, 0);
  sphere(20);
  x = x + 1;
  y = y + 1;
}
```

3D 2: Eindopdracht

- Laat de bal nu naar voren en achter gaan, met de toetsen **q** en **e**
- Maak een nieuwe variabele, bijvoorbeeld **z**, met als beginwaarde nul

Dit ziet er misschien anders uit dan je verwacht!

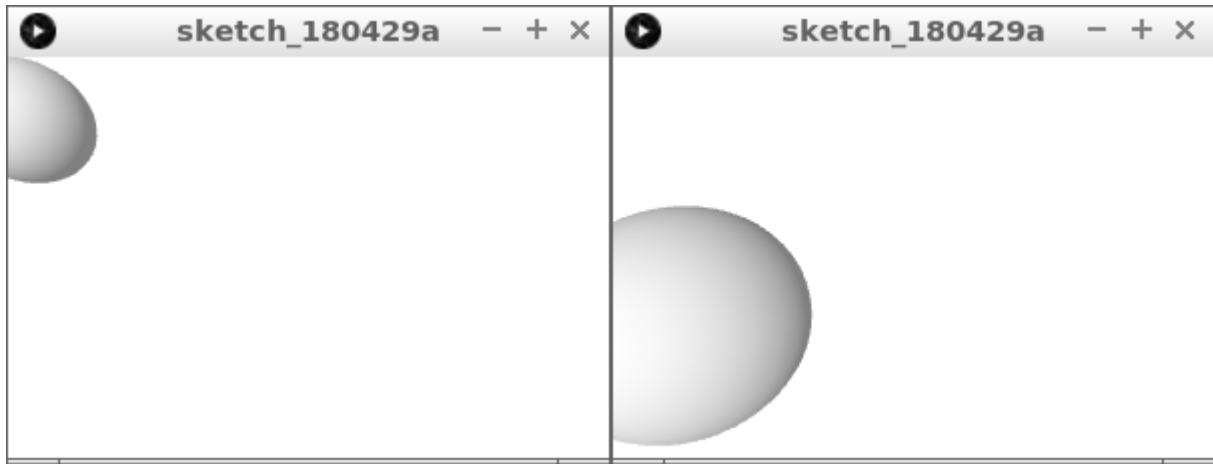


Figure 9: 3D 2: opdracht 3

3D les 3: bal en kubus bewegen

In deze les gaan we een 3D bal en kubus laten bewegen.

3D 3: opdracht 1

Type deze code over:

```
float x1 = 0;
float x2 = 0;

void setup()
{
  size(300, 200, P3D);
  noStroke();
  x1 = width * 1 / 4;
  x2 = width * 3 / 4;
}

void draw()
{
  background(196, 196, 196);
  lights();
  translate( x1,  height / 2, 0);
  box(20);
  translate(-x1, -height / 2, 0);

  translate( x2,  height / 2, 0);
  sphere(20);
  translate(-x2, -height / 2, 0);
}
```

Wat zie je?

3D 3: oplossing 1

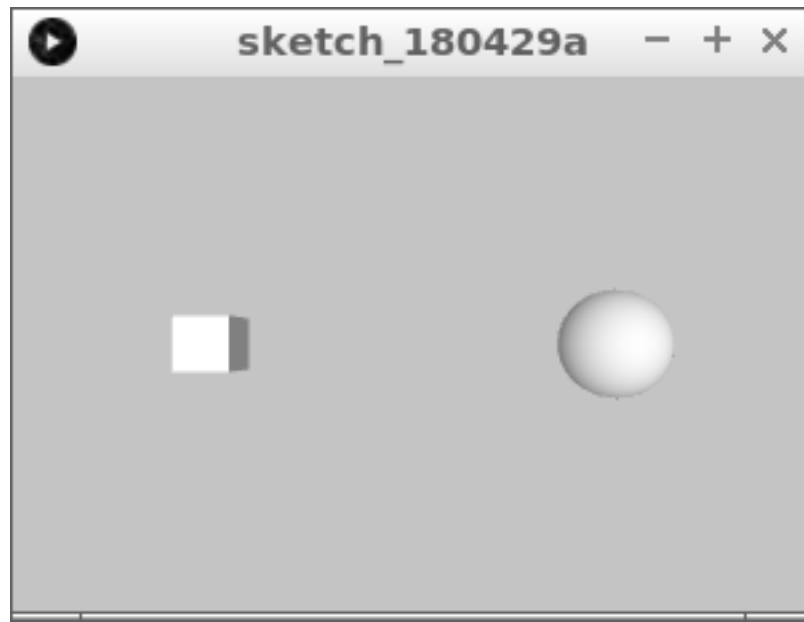


Figure 10: 3D 3: oplossing 1

Je ziet een kubus en een bal die stilstaan

3D 3: opdracht 2

- Zorg dat de kubus nu ook naar onder en boven kan gaan, met de toetsen **w** en **s**
- Maak een nieuwe variabele, bijvoorbeeld **y1**, met als beginwaarde **height / 2**

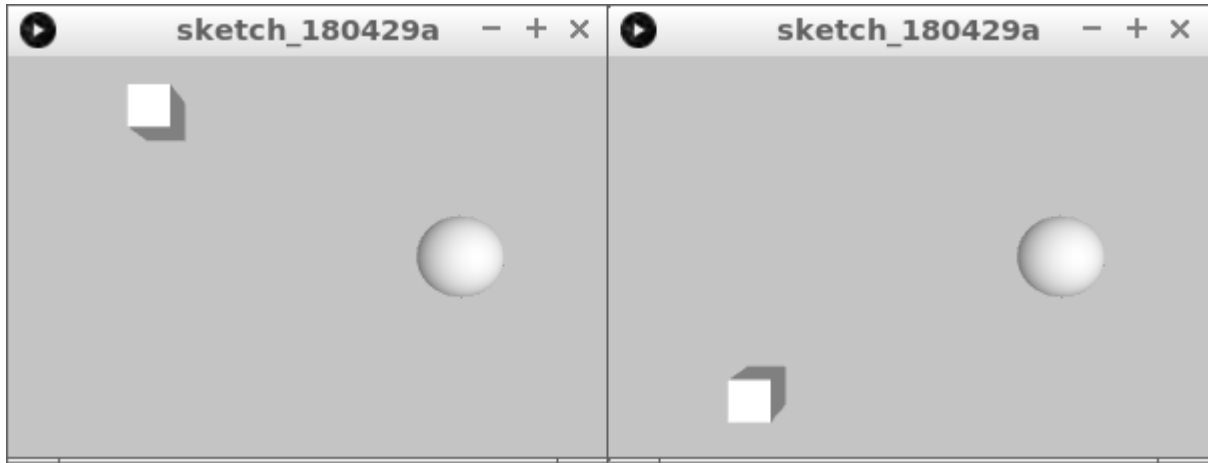


Figure 11: 3D 3: opdracht 2

3D 3: oplossing 2

```
float x1 = 0;
float x2 = 0;
float y1 = 0;

void setup()
{
  size(300, 200, P3D);
  noStroke();
  x1 = width * 1 / 4;
  x2 = width * 3 / 4;
  y1 = height / 2;
}

void draw()
{
  background(196, 196, 196);
  lights();
  translate( x1,  y1, 0);
  box(20);
  translate(-x1, -y1, 0);

  translate( x2,  height / 2, 0);
  sphere(20);
  translate(-x2, -height / 2, 0);

  if (keyPressed)
  {
    if (key == 'w') y1 = y1 - 1;
    if (key == 's') y1 = y1 + 1;
  }
}
```

3D 3: Eindopdracht

- Laat de bal nu ook naar voren en achter gaan, met de toetsen i en k
- Maak een nieuwe variabele, bijvoorbeeld y2, met als beginwaarde `height / 2`

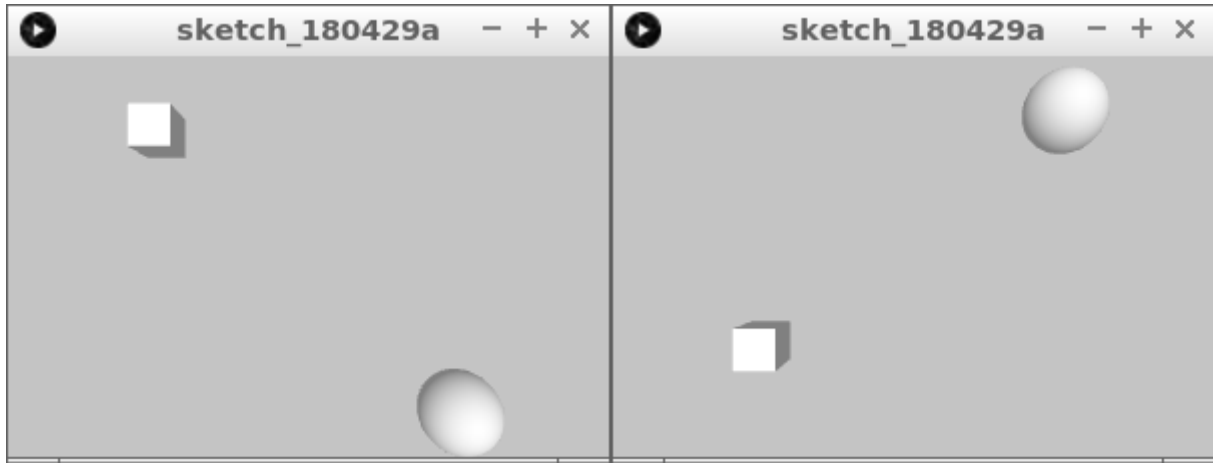


Figure 12: 3D 3: opdracht 3

3D les 4: belichting

In deze les gaan we een 3D bal belichten.

3D 4: opdracht 1

Type deze code over:

```
float licht_richting_x = 1;

void setup()
{
  size(300, 200, P3D);
  noStroke();
}

void draw()
{
  background(255, 255, 255);
  directionalLight(255, 255, 255, licht_richting_x, 0, 0);
  translate(width / 2, height / 2, 0);
  sphere(50);
}
```

Wat zie je?

3D 4: oplossing 1

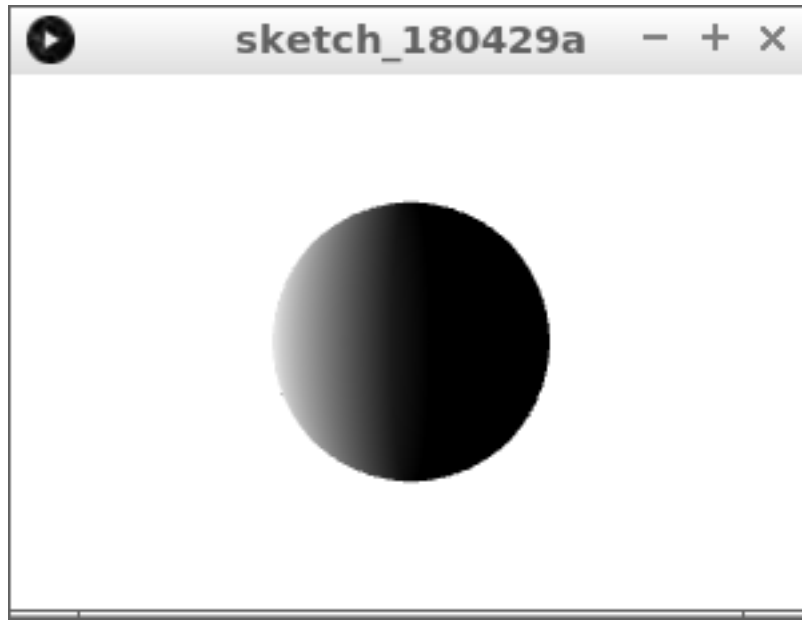


Figure 13: 3D 4: oplossing 1

Je ziet een bal met een wit licht erop.

3D 4: opdracht 2

- Zorg dat de `licht_richting_x` lager en hoger kan worden, met de toetsen `w` en `s`

Let op: er zijn maar drie standen mogelijk!

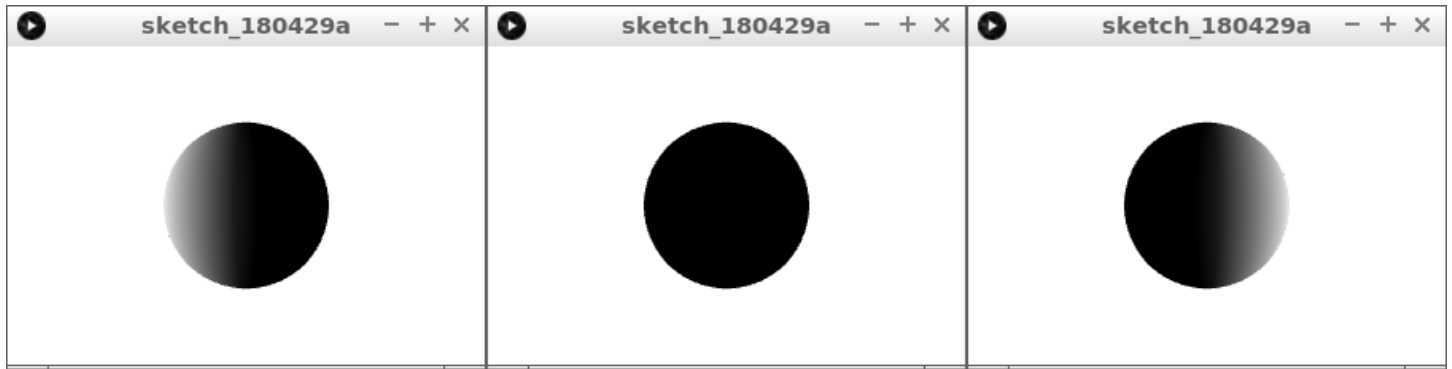


Figure 14: 3D 4: opdracht 2

3D 4: oplossing 2

```
float licht_richting_x = 1;

void setup()
{
  size(300, 200, P3D);
  noStroke();
}

void draw()
{
  background(255, 255, 255);
  directionalLight(255, 255, 255, licht_richting_x, 0, 0);
  translate(width / 2, height / 2, 0);
  sphere(50);
  if (keyPressed)
  {
    if (key == 'a') licht_richting_x = licht_richting_x - 1;
    if (key == 'd') licht_richting_x = licht_richting_x + 1;
  }
}
```

3D 3: opdracht 3

- Zet het licht y richting op een.
- Maak de kleur van het licht rood

Let op: er zijn maar vijf standen mogelijk!

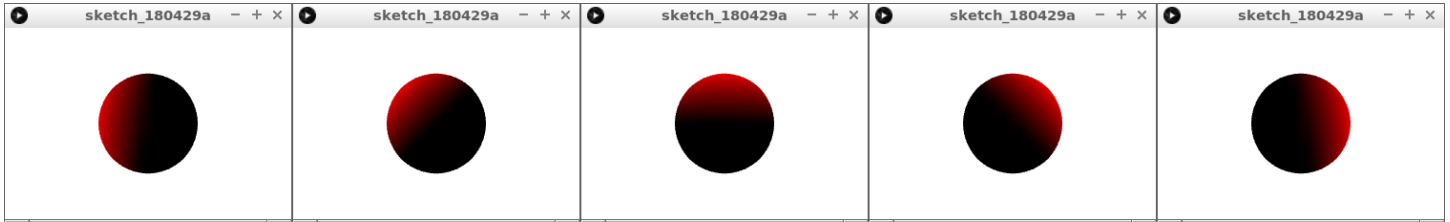


Figure 15: 3D 4: opdracht 3

3D 4: oplossing 3

```
float licht_richting_x = 1;

void setup()
{
  size(300, 200, P3D);
  noStroke();
}

void draw()
{
  background(255, 255, 255);
  directionalLight(255, 0, 0, licht_richting_x, 1, 0);
  translate(width / 2, height / 2, 0);
  sphere(50);
  if (keyPressed)
  {
    if (key == 'a') licht_richting_x = licht_richting_x - 1;
    if (key == 'd') licht_richting_x = licht_richting_x + 1;
  }
}
```

3D 4: Eindopdracht

- Zorg dat de `licht_richting_y` lager en hoger kan worden, met de toetsen `w` en `s`
- Maak een nieuwe variabele, bijvoorbeeld `licht_richting_x`, met als beginwaarde nul
- Maak de lichtkleur groen

Nu kan de lichtbundel mooi draaien.

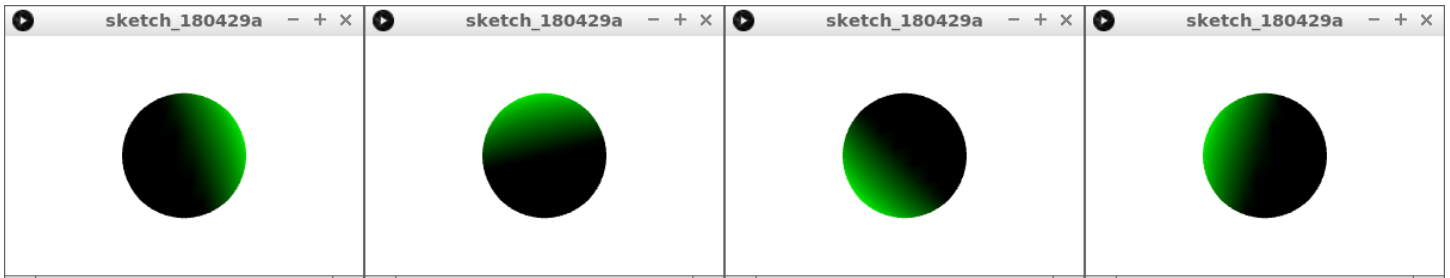


Figure 16: 3D 4: eindopdracht