

Sneeuwpop

Steven Bronsveld

April 17, 2019

1 Gegevens

1.1 Algemene bronnen

- GitHub Pagina
<http://www.github.com/StevenBrons/Processing>
- Nature of Code boek
<https://natureofcode.com/book>
- Processing Basic tutorial
https://www.youtube.com/user/shiffman/playlists?view=50\&sort=dd\&shelf_id=2

2 Introductie

In deze module Processing ga je coderen in het Processing framework. Er wordt veel gebruik gemaakt de video's en de website van **Daniel Shiffman**. Het is aan jezelf de bronnen die aangeleverd worden ook te gebruiken. Voor elke assignment is het de bedoeling dat je alle opdrachten maakt en inlevert voor de door de begeleider gegeven inlever datum.

Verder kan het voorkomen dat er voor bepaalde opdrachten of bronnen [Optioneel] staat. Dit geeft aan dat je dit materiaal niet hoeft in te leveren. Als je de stof moeilijk vindt (of meer wil weten) worden deze opdrachten sterk aangeraden. Voor sommige opdrachten staat [Bonus], doe deze opdrachten als je extra verdieping en uitdaging wil.

3 Leerdoelen

Deze week leer je omgaan met de volgende concepten:

- Variabelen: `int`
- Standaard functions:
 - `size(int width,int height)`
 - `circle(int x, int y, int r)`
 - `random(float bound)`
 - `stroke(float r,float g,float b)`
 - `fill(float r,float g,float b)`
 - `background(float r,float g, float b)`
 - `strokeWeight(int w)`
- Globale variabelen: `width`, `height`, `mouseX`, `mouseY`
- Het aanroepen van functions
- Het maken van eigen functions
- Het gebruiken van for-loops
- Standaard operatoren: `a + b`, `a - b`, `a * b`, `a / b`, `a % b`, `a ^ b`

4 Uitleg

- Interactive Processing Editor
<https://hello.processing.org/editor/>
- [Optioneel] Introduction - Processing Tutorial
<https://www.youtube.com/watch?v=2VLaiR5Ckbs\&list=PLRqWX-V7Uu6ZYJC7L-r6rX6utt6wwJCyi>

- Pixels - Processing Tutorial
https://www.youtube.com/watch?v=a562vsSI2Po&list=PLRqWX-V7Uu6bsRnSEJ9tRn4V_XCGXovs4
- Processing Environment - Processing Tutorial
https://www.youtube.com/watch?v=5N31KNg000g&list=PLRqWX-V7Uu6Yo4VdQ4ZTtqRQ1AE4t_Ep9
- Interaction - Processing Tutorial
<https://www.youtube.com/watch?v=o8dfrrZ86gs&list=PLRqWX-V7Uu6by61pbhdvyEpIeymlmnXzD>
Alleen **3.1**
- Variables - Processing Tutorial
https://www.youtube.com/watch?v=B_-ycSR3ntik&list=PLRqWX-V7Uu6aFN0goIMSbSYOkKNT089uf
Alleen **4.1**

5 [optioneel] Voorbeelden

Bekijk de volgende voorbeelden. Ga per regel na of je snapt wat er gebeurt. Pas eventueel wat dingen aan om te kijken wat het effect is. Als je codeert mag je het internet en de voorbeelden altijd gebruiken, maak hier dus gebruik van!

1. `houses`
2. `chess`

6 Opdrachten

Je gaat de eerste code schrijven! Zorg ervoor je een nieuwe *sketch* aanmaakt.

6.1 Een cirkel

Plak de volgende code in je lege sketch. Run de sketch om te kijken wat dit doet!

```
1 void setup() {
2     size(500,500);
3     circle(100,100,70);
4 }
5 void draw() {
6 }
```

Pas de code aan zodat de cirkel in het midden van het scherm staat.

Opmerking

Oudere versies van Processing hebben geen `circle(x,y,radius)` functie. Gebruik dan `ellipse(x,y,width,height)`

6.2 Het midden

Als je de grote van het scherm aanpast (pas de arguments van `size` aan), zul je zien dat de cirkel niet meer in het midden staat. Gebruik de variabelen `width` en `height` om het midden van het scherm te berekenen. Zorg ervoor dat de cirkel in het midden van het scherm blijft staan, onafhankelijk van de grote van het scherm.

6.3 Een sneeuwpop

Maak een sneeuwpop door twee cirkels half-boven de eerste cirkel te tekenen. Zorg ervoor dat deze cirkels ook op de juiste plek blijven staan als je de schermgrootte verandert!

6.4 Een function

Het tekenen van de sneeuwpop is een simpel **algoritme**. Als je een programma wil maken is het enorm belangrijk je het op te delen. Dit opdelen gebeurt onder meer door het maken van **functions**. Plaats de volgende function **helemaal onderaan** je de sketch.

```
1 void drawSnowman(int x, int y) {  
2 }
```

Plaats de volgende regel code onderaan in `setup`.

```
1 drawSnowman(int(random(width)),int(random(height)));
```

Verplaats het tekenen van je sneeuwpop naar de `drawSnowman` function. Het is belangrijk dat je de sneeuwman op de coördinaten `x,y` plaatst.

6.5 Meer sneeuwpoppen

Maak een `for` loop in de `setup` die 10 sneeuwpoppen op willekeurige plekken op het scherm plaatst.

6.6 Gekleurde sneeuwpoppen

In de wereld van code kan alles! Waarom zouden we ons limiteren tot witte sneeuwpoppen? Zorg ervoor dat alle sneeuwpoppen verschillende kleuren hebben!

6.7 Nog meer sneeuwpoppen

Voeg de volgende function toe:

```
1 void mouseClicked() {  
2 }
```

Deze function wordt uitgevoerd op het moment dat je met de muis op de sketch klikt. Zorg ervoor dat er een sneeuwpop op de plek van de muis verschijnt als je klikt.

Tip

Zoek op het internet naar een manier om de muis coördinaten te krijgen

6.8 [Bonus] Hogere sneeuwmannen

Pas de function `drawSnowman` aan zodat er nog een derde variable is:

```
1 void drawSnowman(int x, int y, int n) {  
2 }
```

Deze derde variable geeft de hoogte van de sneeuwpop aan. Zorg ervoor dat ook de hoogte (het aantal cirkels) variabel is! Let op! De cirkels moeten zowel kleiner worden, als boven op elkaar gestapeld worden! Zorg dat je ook 100-ballen-hoge sneeuwpoppen kunt maken! Pas vervolgens

```
1 drawSnowman(int(random(width)),int(random(height)));
```

aan zodat de sneeuwpoppen tussen de 2 en 6 hoog zijn.

6.9 [Bonus] Sneeuwmannen dans

Maak de function `makeCircle`, met als parameters een `int n`

```
1 void makeCircle(int n) {  
2 }
```

Deze function moet `n` sneeuwmannen in een cirkel om het midden van het scherm tekenen. De sneeuwmannen moeten allemaal op gelijke afstand van elkaar staan.

7 Inleveren

Lever je gemaakte sketch in.