SPD MODULE 3: STRUCTUUR, INTERACTIE, VOORWAARDEN EN EVENTS

LESSTOF

Screencasts: https://www.youtube.com/playlist?list=PL2652C3D0F4D07BAF

Reader: Hoofdstuk 3

OEFENOPGAVEN

Let op

In deze oefenopgaven ga je gebruik maken van zogenaamde events (gebeurtenissen), bijv. als je de muis beweegt treden er 'events' op waar je in je programma op kunt reageren. Processing is zo gemaakt dat dit alleen maar werkt als je ook een draw-methode in je sketch hebt staan.

M.a.w. de volgende code werkt niet:

```
void setup() {
  size(500, 300);
}

void mouseClicked() {
  println("KLIK!");
}
```

Maar de volgende code wel, ondanks dat er in de draw niets gebeurt (je moet natuurlijk wel *in* het scherm klikken):

```
void setup() {
  size(500, 300);
}

void draw() {
}

void mouseClicked() {
  println("KLIK!");
}
```

OPGAVE 3.1

Maak een secondenteller. Het programma begint op 0 en toont midden in een window het aantal verstreken seconden. Elke seconde moet het de geschreven waarde dus worden overschreven. Met de methode *millis()* kun je het aantal milliseconden (1/1000^{e} seconde) opvragen.

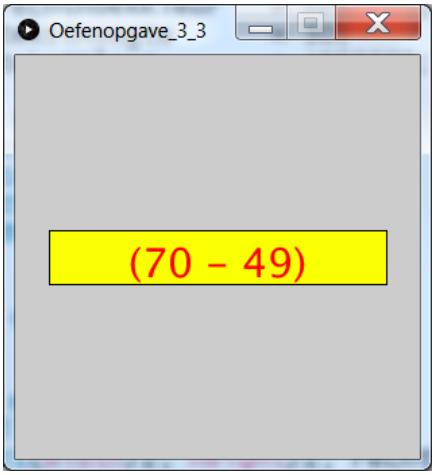
OPGAVE 3.2

Zoek in de processing reference naar de mouse-methoden. Eén van die methoden wordt aangeroepen als er met de muis bewogen is. Maak een programma dat in het console de X- en Y- positie van de muis afdrukt als je met de muis beweegt.

Beweeg met de muis. Wat gebeurt er in het console? Maakt het verschil waar je beweegt met je muis?

OPGAVE 3.3

Schrijf een programma dat in het windows de actuele X- en Y-positie van de muis afdrukt wanneer je op de muis klikt. Geef deze waarde midden in je window in rood op een gele achtergrond weer. Let erop dat het ook in het midden moet staan wanneer de grootte van het window wordt aangepast!



Valt je op dat de weergegeven positie de positie is binnen je eigen window? Muisbewegingen en -clicks buiten jouw window worden door het programma genegeerd!

OPGAVE 3.4

Neem als basis opgave 3.3, De coördinaten van de muis moeten nu echter alleen worden weergegeven wanneer de x-coördinaat *even* is. Wanneer de x-coördinaat oneven is, gebeurt er niets.

NB: gebruik de modulo operator (%) om te bepalen of een getal even is.

OPGAVE 3.5

Neem als basis het resultaat van opgave 3.4. Nu moet je echter de coördinaten alleen weergeven wanneer zowel de x-coördinaat als de y-coördinaat even is.

OPGAVE 3.6

Druk de coördinaten uit opgave 3.5 nu alleen af wanneer één van beide coördinaten even is.

OPGAVE 3.7

Het programma uit opgave 3.6 kunnen we natuurlijk ook omzetten zodat er een spel van te maken is. Dit gebeurt als volgt:

De speler begint met 25 credits. Hij beweegt met de muis en klikt. Wanneer beide coördinaten even zijn, verdient hij een credit. Wanneer een coördinaat even is en de ander oneven, verandert er niets en wanneer beide coördinaten oneven zijn, verliest hij een credit. Wanneer de credits op zijn, begint hij opnieuw.

OPGAVE 3.8

De winstsituatie uit opgave 3.7 moet kunnen verbeteren. Dat wil zeggen: Wanneer je twee keer achter elkaar “dubbel even” hebt geklikt, krijg je voor de tweede keer niet één maar twee credits. De derde keer achter elkaar levert 3 credits op. De vierde keer opeenvolgend levert 5 credits op en bij 5 keer of vaker levert het 10 credits op. Pas je programma hier op aan.

Extra uitdaging:

Controleer of de muis wel verplaatst is voordat er geklikt is. Dit kan op 2 manieren. Welke?

MODULEOPGAVE 3: ROBOTJE

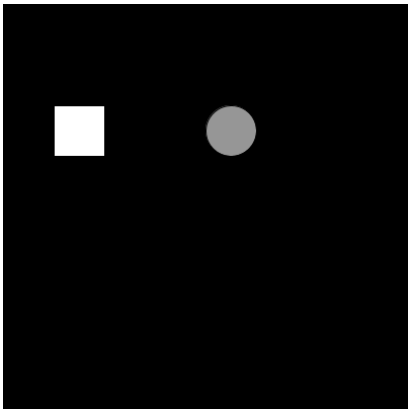
In deze opgave maak je een simpele simulatie van een robot die een vracht kan dragen. De robot is een vierkantje dat je met de pijltjestoetsen over het scherm kunt bewegen en de vracht ziet eruit als cirkel.

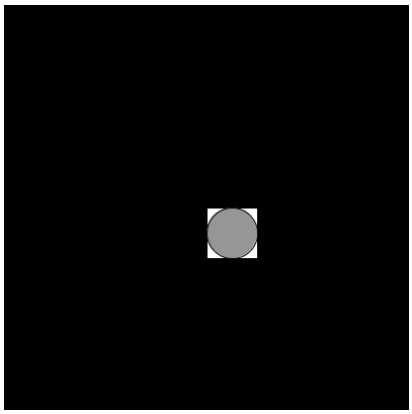
Zodra de robot op de vracht staat moet je met de “enter”-toets de vracht op kunnen tillen en mee kunnen nemen. Ook moet de vracht weer neergelegd kunnen worden met dezelfde toets.

Extra voorwaarden/hints:

- De robot moet het speelveld niet kunnen verlaten
- Zorg ervoor dat de stappen die de robot zet net zo groot zijn als de robot zelf, zodat hij altijd goed uitgelijnd blijft op het speelveld.

Hieronder zijn twee screenshots te zien van de robot en de vracht.





DEEL 1 (ANALYSE/ONTWERP)

Beschrijf welke variabelen je nodig hebt en waarvoor. Geef per variabele de naam, datatype en waarvoor ze gebruikt wordt aan.

Beschrijf verder op welke event/s je programma moet reageren. Geef per event zo nauwkeurig mogelijk (in je eigen woorden) aan wat er na het aanroepen moet gebeuren. Gebruik voor deze beschrijving misschien ook de eerder beschreven variabelen.

DEEL 2 (IMPLEMENTATIE)

Implementeer je programma in Processing en maak daarbij zoveel mogelijk gebruik van de resultaten van deel 1.