

## W4 - L1 Programma

### Nabespreking

### Opgaven

#### Constrain II

*Individueel / tweetallen*

Kijk eens naar de volgende code:

```
int x = constrain( 100, 0, 200);
```

Wat doet deze code en beredeneer of dat zinvol is?

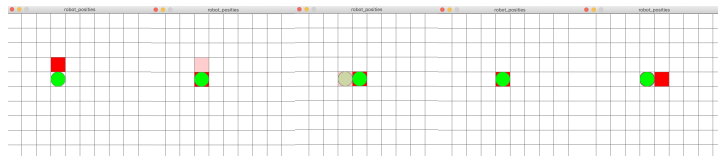
#### Industriële Robot II

*Individueel*

De industriële transportrobot zoals uit Industriële Robot I heeft niet echt waarde als je er alleen maar mee kan rondrijden. Daarom gaan we er een functionaliteit aan toevoegen: objecten verplaatsen.

Als de robot op een object staat, dan moet het object worden opgepakt als er op de ENTER toets gedrukt wordt. Als het object is opgepakt en de robot zich verplaatst, dan moet de robot het object meenemen. Het object moet weer losgelaten worden als weer op de ENTER toets gedrukt wordt. Natuurlijk kan de robot het object niet oppakken als hij er niet bovenop staat. Dit is weergegeven in onderstaande figuur.

*(Vlnr: Robot (vierkant) en object (cirkel) naast elkaar, robot en object op zelfde positie, object meegenomen door robot, robot en object blijven stilstaan om te lossen, robot na lossen gaat alleen verder)*



Schrijf het oplossingsalgoritme.

---

**Let op** In navolgende opgaven ga je gebruik maken van events. Ter herinnering: 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!");
}
```

---

### Industriële robot III

#### *Individueel*

In deze opgave gaan we opgaven Industriële Robot I en Industriële Robot II implementeren. Het resultaat is een eenvoudige 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. De stappen die de robot zet zijn net zo groot zijn als de robot zelf, zodat hij altijd goed uitgelijnd blijft op het speelveld. 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. De algoritmen hiervoor heb je al geschreven.

1. Beschrijf welke variabelen je nodig hebt en waarvoor. Geef per variabele de naam, datatype en waarvoor ze gebruikt wordt aan.
2. Kies de event(s) waarop je programma moet reageren en geef aan waarom juist deze.
3. Schrijf het programma in Processing en *laat de algoritmen als commentaar in de code staan*.

### Industriële robot IV

Maak Industriële robot III af en zorg ervoor dat de robot het werkveld niet kan verlaten (de robot mag niet buiten het beeld van de GUI terecht komen) .

### EXTRA: Industriële robot, meerdere objecten

Breidt jouw programma uit met meerdere cargo's die verdeelt staan over het speelveld. Natuurlijk gebruik je een array om de positie van deze objecten in bij te houden.