# Test

*Deze pagina biedt een **stap-voor-stap handleiding** om te leren programmeren in Scratch door het bouwen van een **doolhofspel**. Je volgt de lessen door middel van **uitlegvideo's**, maakt **opdrachten direct in Scratch** en leert daarbij belangrijke programmeerconcepten zoals beweging, loops, if-then-else statements, variabelen en het resetten van de speltoestand. De cursus begeleidt je van de introductie en het opzetten van het startproject tot het afronden van je eigen unieke versie van het spel en het reflecteren op wat je hebt geleerd.*

# *Stap 0 - Introductie en start*

Welkom! We gaan je stap voor stap helpen om Scratch te begrijpen, zodat je zelf leert programmeren. Maar wat gaan we eigenlijk doen?

## ? Leerdoelen

- Je ontdekt de Scratch **interface**: waar je leert in blokken te prgrammeren .
- Je leert wat sprite is en je leert deze te laten **bewegen.**
- Je leert wat een herhaalblok (Engels: **loop**) is.
- Je leert wat een als-dan-anders blok (Engels: **if-then-else**) is.
- Je leert wat een **variabele** is en hoe je die kan in Scratch kan gebruiken

### Scratch

image.png
*Dit is een plaatje vand e Scratch interface*

Scratch is een grafische manier van programmeren. Het is goede eerste stap omdat je zo leert **nadenken als een programmeur**, zonder dat je meteen hoeft te worstelen met moeilijke code of foutmeldingen. In Scratch werk je met blokken die niet kunnen "crashen", waardoor je snel resultaat ziet. Pas als je begrijpt hoe logica, lussen en beslissingen werken, is het makkelijker om over te stappen naar een echte programmeertaal zoals Python.

### Wat kun je verwachten?

- Je bekijkt korte uitlegvideo's van Felienne, een docent aan de TU Delft.

- Je maakt opdrachten direct in Scratch.
- Je leert stap voor stap hoe je een eigen **doolhofspel** bouwt.

Als je nog meer wilt weten van deze module dan kan je hier meer informatie vinden.

## Spelregels

- Je bouwt het spel stap voor stap op. Daarna maak je je eigen variant.
- Het inleveren van andermans werk is **fraude** en zorgt er voor dat je een module wordt geblokkeerd. Er volgt een gesprek met je studiecoach. Samen met hem/haar bepaal je onder welke voorwaarden jij je studie nog kan voortzetten.
- Maar het is ook écht leuk om je eigen spel te maken! Kom je er even niet uit, dan mag je natuurlijk wel een medestudent om hulp vragen.
- Voor deze opdrachten gebruiken we (nog) **geen AI** — dat komt later. Je mag AI wel om extravuitleg vragen als je iets niet begrijpt,
- Bij elke opdracht lever je iets in: soms is dat je code (of een screenshot daarvan), soms een antwoord op één of meer vragen. De antwoorden schrijf je in je **eigen woorde**n. Door zelf na te denken over de vragen, begrijp je de stof beter  en word je uiteindelijk een goede softwaredeveloper.

**Voorbereiding:** Open dit project in Scratch:

https://scratch.mit.edu/projects/96709199/#editor

(of download het bestand hier: Doolhof Start.sb3)

## Taal aanpassen

Standaard staat de taal op English, deze module is in het **Nederlands** dus als je de taal wil aanpassen dan kan dat onder **settings** - **laguage**.

Je zou dit startscherm moeten zien:

image.png

## Bekijk de uitlegvideo

In deze video legt Felienne uit wat je gaat bouwen en hoe je aan de slag kunt. Let goed op de uitleg over hoe Scratch werkt.

https://www.youtube.com/embed/JcIVNv4VZv8?si=8FT4Hu648QqKqrCY

**Alternatieve link:** Klik hier als de video hierboven niet werkt

# ?? Opdracht

Vertel in een paar regels of je al ervaring hebt met programmeren. Wat heb je gedaan? Formuleer dit in minimaal drie zinnen.

Heb je geen ervaring? Beschrijf kort wat je met deze opdrachten hoop te leren. Wat zijn je verwachtingen? Formuleer dat in minimaal drie zinnen.

# ? Inleveren

Typ je antwoord in de tekstbox in.

Gebruik je **eigen woorden** en denk aan wat je in het spel gaat bouwen.

# *Stap 1 – Beweging*

We beginnen met de besturing van je hoofdpersonage, Giga. In deze stap leer je hoe je Giga kunt laten bewegen met de pijltjestoetsen.

## Wat leer je?

- Bewegen over de X-as (links en rechts).
- Bewegen over de Y-as (omhoog en omlaag).

## Bekijk de uitlegvideo

https://www.youtube.com/embed/IGtsy24vbiE?si=ZYnQwgoNBWrrlWw_

**Alternatieve link:** Bekijk de video via HiDrive

image.png

# ?? Opdracht

Hoeveel richtingen kan jouw poppetje bewegen aan het einde van deze stap?

# ? Communiceren met AI

Als je AI later vraagt om beweging te maken, is alleen "*laat mijn poppetje bewegen*" niet nauwkeurig genoeg. Beschrijf ten minste:

- **Welk object** beweegt: de sprite Giga.
- **Door welke gebeurtenis** de beweging begint: het indrukken van een pijltjestoets.
- **Over welke as** de sprite beweegt: de X-as of de Y-as.
- **In welke richting** de waarde verandert: groter of kleiner.

**Voorbeeld van een nauwkeurige opdracht:**

Wanneer ik de rechterpijl indruk, laat Giga naar rechts bewegen door zijn X-positie te verhogen. De Y-positie moet hetzelfde blijven.

**Denkvraag:** Hoe zou je op dezelfde manier de beweging bij de pijl omhoog beschrijven?

# ? Inleveren

Vul in het tekstvak in hoeveel richtingen jouw sprite beweegt.

# *Stap 2 – De lus (loop)*

In deze stap leren we hoe je een herhaling maakt met een lus. Hiermee kun je code telkens opnieuw uitvoeren zolang een bepaalde voorwaarde klopt.

## Wat leer je?

Je gebruikt een **herhaalblok (loop)** om acties te blijven herhalen.

## Bekijk de uitlegvideo

https://www.youtube.com/embed/OFzSUt23Lho?si=r63G16qrEFFOA2YY

**Alternatieve link:** Bekijk de video via HiDrive

image.png

image.png

# ?? Opdracht

Leg in 2 tot 4 zinnen in je eigen woorden uit wat een lus (loop) doet in een programma.

# ? Communiceren met AI

Bij een lus moet je niet alleen vertellen *wat* er wordt herhaald, maar ook *hoe lang* de herhaling doorgaat.

- **Actie:** welke instructies worden herhaald?
- **Herhaling:** hoe vaak of zolang welke voorwaarde waar is?
- **Stopmoment:** wanneer moet de lus eindigen?

**Voorbeeld:**

Blijf controleren welke pijltjestoets is ingedrukt zolang het spel actief is. Verplaats Giga bij iedere controle een klein stukje in de gekozen richting. Stop deze herhaling wanneer het spel is afgelopen.

**Denkvraag:** Welke actie wordt in jouw Scratch-code herhaald en wanneer stopt die herhaling?

# ? Inleveren

Typ je uitleg over de lus in het tekstvak.

# *Stap 3 – Als-dan-anders*

We voegen nu logica toe: wat moet er gebeuren als iets gebeurt?

## Wat leer je?

Je leert het ***als-dan-anders***-blok (if-then-else) gebruiken.

## Bekijk de uitlegvideo

https://www.youtube.com/embed/6f9TeuFZshM?si=XWdXdtFBhWh0XuD8

**Alternatieve link:** Bekijk de video via HiDrive

# ?? Opdracht

Wat doet een als-dan-anders (if-then-else)-blok in je code? Leg het uit in je eigen woorden.

# ? Communiceren met AI

Een opdracht met een beslissing wordt duidelijker als je drie onderdelen benoemt:

1. **Als:** de voorwaarde die wordt gecontroleerd.
2. **Dan:** wat er gebeurt als de voorwaarde waar is.
3. **Anders:** wat er gebeurt als de voorwaarde niet waar is.

**Voorbeeld:**

Als de rechterpijl is ingedrukt, verhoog dan de X-positie van Giga. Anders blijft zijn X-positie ongewijzigd.

**Denkvraag:** Wijs in jouw eigen Scratch-code de voorwaarde, de actie bij *waar* en de actie bij *onwaar* aan.

# ? Inleveren

Typ je uitleg in het tekstvak.

# *Stap 4 – Terug naar startpositie*

Als Giga een muur raakt, willen we dat hij teruggaat naar het begin.

## Wat leer je?

Je leert leert de sprite, Giga terug te zetten op zijn beginpositie.

## Bekijk de uitlegvideo

https://www.youtube.com/embed/lRX_n2S0Hwk?si=YNeZtufi6l11jnJX

**Alternatieve link:** Bekijk de video via HiDrive

image.png

image.png

image.png

# ?? Opdracht

Welke coördinaten (X en Y) gebruikt jouw spel om Giga terug te zetten naar de start?

# ? Communiceren met AI

De opdracht was:

"Als Giga een muur raakt, willen we dat hij teruggaat naar het begin."

Maar hoe zou je deze opdracht nauwkeuriger kunnen formuleren als je (straks) met AI gaat werken?

Wanneer Giga de muur raakt, zet zijn X- en Y-positie terug naar de startpositie. Laat de score ongewijzigd.

In deze versie staan vier belangrijke onderdelen:

- **Trigger:** Giga raakt de muur.
- **Actie:** verander zijn X- en Y-positie.
- **Gewenste toestand:** gebruik de coördinaten van de startpositie.
- **Beperking:** verander de score niet.

**Denkvraag:** Welke exacte X- en Y-waarde zou je aan de opdracht toevoegen, zodat er geen twijfel bestaat over de startpositie?

# ? Inleveren

Typ de X- en Y-positie in het tekstvak.

# *Stap 5 – De sleutel*

Je spel is bijna klaar! In deze stap voeg je een sleutel toe. Als Giga deze aanraakt, is het spel gewonnen.

## Wat leer je?

Je leert controleren of Giga de Sleutel aanraakt en als dat gebeurt dan wordt het spel beëindigd.

## Bekijk de uitlegvideo

https://www.youtube.com/embed/UoqZhd4ggn0?si=j8a5GLqdgiNfATHc

**Alternatieve link:** Bekijk de video via HiDrive

# ?? Opdracht

Maak een screenshot van je hele browser waarin je de gemaakte code laat zien.

# ? Communiceren met AI

Het aanraken van de sleutel noemen we een **botsing** of **collision**. Bij het beschrijven van een botsing benoem je beide objecten en het gevolg.

Controleer voortdurend of de sprite Giga de sprite Sleutel raakt. Als ze elkaar raken, beëindig dan het spel en toon dat de speler heeft gewonnen.

**Denkvraag:** Wat mag er gebeuren zolang Giga de sleutel nog niet raakt?

# ? Inleveren

Upload de screenshot met jouw Scratch-code.

# *Stap 6 – Monster toevoegen*

We maken het spannend! Voeg een monster toe dat je moet ontwijken. Laat het monster bewegen.

## Bekijk de uitlegvideo

https://www.youtube.com/embed/6sYbPEse-jc?si=ugD4jr0WJ0vxC-8N

**Alternatieve link:** Bekijk de video via HiDrive

# ?Let op

De detectie van een botsing met het spook werk niet altijd erg goed. Dt komt door het gebruik van

image.png

Dit commando wordt eerst helemaal uitgevoerd voordat de rest van het programma iets doet.

Als je dit niet helemaal begrijpt dan is dat goed en je mag een alternatieve spook beweging bedenken zonder dat je dit command gebruikt.

Hier een link naar een voorbeeld van een alternatieve spook-beweging. Het spook maakt telkens een stapje en als het de rand raakt dan draait het iets en maakt het weer stapjes tot het de rand raakt enzovoort, enzovoort.

# ?? Opdracht

Laat het monster bewegen. Lever een screenshot in van de code die het monster laat bewegen.

# ? Communiceren met AI

Als code niet goed werkt, heeft AI meer aan een nauwkeurige **foutbeschrijving** dan aan de vraag "*Waarom werkt het niet?*" Beschrijf:

- **Verwacht gedrag:** wat zou er moeten gebeuren?
- **Werkelijk gedrag:** wat gebeurt er nu?
- **Moment:** wanneer ontstaat het probleem?
- **Relevante code:** welke Scratch-blokken horen bij dit gedrag?

**Voorbeeld:**

Ik verwacht dat het monster steeds een klein stukje beweegt en dat Scratch tussendoor controleert of het Giga raakt. Nu voert het monster eerst zijn hele beweging uit en wordt de botsing pas daarna gecontroleerd. Leg uit waardoor dit gebeurt. Geef nog geen oplossing.

**Let op:** in deze les gebruik je AI alleen voor extra uitleg. Je maakt en controleert de code zelf.

## ? Inleveren

Upload de screenshot van de code van het monster.

# *Stap 7 – Game over*

Als het monster Giga raakt, is het spel afgelopen. Je leert nu hoe je het spel opnieuw kunt laten starten.

### Bekijk de uitlegvideo

https://www.youtube.com/embed/LR5vDCgcLMM?si=Kj7SCUr9BADcQYGp

**Alternatieve link:** Bekijk de video via HiDrive

## ?? Opdracht

Laat zien dat Giga teruggezet wordt naar het begin wanneer hij de muur of het monster raakt.

## ? Inleveren

Lever een screenshot in van de code die dit laat zien.

# *Stap 8 – Alles terugzetten*

Als het spel opnieuw begint, moeten zowel het monster als Giga teruggezet worden naar hun beginpositie. Pas ook de snelheid aan als het spel te moeilijk is.

### Bekijk de uitlegvideo

https://www.youtube.com/embed/N76K2ZfCP3o?si=Yf6DBrKI71vHcx2V

**Alternatieve link:** Bekijk de video via HiDrive

## ?? Opdracht

Pas de snelheid van het monster aan en laat zien hoe je alles terugzet.

# ? Communiceren met AI

De posities, snelheid en score vormen samen de **toestand** van het spel. Bij een herstart moet je precies aangeven welke delen van die toestand worden teruggezet.

Wanneer het spel opnieuw start, zet Giga en het monster terug naar hun eigen startpositie. Zet de snelheid van het monster terug naar de beginsnelheid en zet de score op nul. Start daarna pas de beweging opnieuw.

**Denkvraag:** Welke waarden vormen in jouw spel de begintoestand?

# ? Inleveren

Lever een screenshot in van de code waarin het monster wordt teruggezet en je de snelheid hebt aangepast.

# *Stap 9 – Punten bijhouden*

We voegen nu een **score** toe met behulp van een variabele.

> Een variabele is een waarde die je kunt opslaan en aanpassen tijdens het spel.

## Bekijk de uitlegvideo

https://www.youtube.com/embed/1GTrDEd1ECY?si=_ALBcbZ-dacTo0dq

**Alternatieve link:** Bekijk de video via HiDrive

Om de score bij te houden gebruiken we een variabele.

| | |
|---|---|
| image.png | image.png |

# ?? Opdracht

Laat zien hoe je de score bijhoudt.

# ? Communiceren met AI

Wanneer je AI vraagt een variabele te gebruiken, beschrijf dan vier eigenschappen:

- **Naam:** bijvoorbeeld `score`.
- **Beginwaarde:** bijvoorbeeld 0.
- **Verandering:** bijvoorbeeld verhogen met 1.
- **Trigger:** bijvoorbeeld wanneer Giga de sleutel raakt.

Maak een variabele met de naam score en geef deze bij de start de waarde 0. Verhoog de score met 1 wanneer Giga de sleutel raakt. Toon de actuele score tijdens het spel.

**Denkvraag:** Moet de score na een botsing met de muur veranderen? Voeg dit als beperking toe aan de opdracht.

# ? Inleveren

Lever een screenshot in van je code waarin je de score bijhoudt met een variabele.

# *Stap 10 – Afronden*

We maken het spel af en zorgen dat het er goed uitziet. Je kunt je spel opslaan op je laptop of online delen.

## Bekijk de uitlegvideo

https://www.youtube.com/embed/cHWgQNulLjU?si=5iQpiZdUwrxSB8UU

**Alternatieve link:** Bekijk de video via HiDrive

# ?? Opdracht

Maak het spel af, voeg iets unieks toe en sla het op.

image.png

# ? Communiceren met AI

"*Voeg iets unieks toe*" is bewust een open opdracht. Voor AI moet je jouw idee eerst omzetten in duidelijke **eisen**. Eisen beschrijven wat het resultaat moet kunnen.

**Voorbeeld met een tweede sleutel:**

- Er staat een tweede sleutel op een andere positie in het doolhof.
- Iedere sleutel verhoogt de score maar één keer.
- De speler wint pas nadat beide sleutels zijn geraakt.
- Een botsing met de muur zet Giga terug, maar verandert de sleutels en score niet.

Deze controleerbare punten noemen we **acceptatiecriteria**: hiermee kun je achteraf testen of de opdracht goed is uitgevoerd.

**Denkvraag:** Schrijf voor jouw unieke toevoeging minimaal drie acceptatiecriteria. Je hoeft AI niet te gebruiken om deze te bedenken.

## ? Inleveren

Lever jouw eigen unieke versie van het spel in.

# *Stap 11 – Wat heb je geleerd?*

image.png

Je hebt veel geleerd over programmeren. Nu kijk je terug op wat je hebt gemaakt, herhaal je de belangrijkste begrippen en oefen je hoe je jouw kennis gebruikt om later duidelijke opdrachten aan AI te geven.

image.png

## ? Begrippen herhalen

Als je met een programmeur of met AI communiceert, helpen de juiste begrippen je om precies te zeggen wat je bedoelt. Herhaal daarom deze begrippen uit de les:

- **Sprite:** een object of personage in Scratch met een uiterlijk, positie en eigen code.
- **Coördinaten:** de X- en Y-waarde die samen de positie van een sprite bepalen.
- **Gebeurtenis of trigger:** het moment waarop gedrag begint, bijvoorbeeld het indrukken van een toets of het raken van een muur.

- **Instructie:** één opdracht die het programma uitvoert.
- **Volgorde:** de volgorde waarin instructies worden uitgevoerd. Een andere volgorde kan een ander resultaat geven.
- **Lus of loop:** een constructie waarmee instructies worden herhaald.
- **Voorwaarde of conditie:** een controle die waar of niet waar kan zijn.
- **Als-dan-anders of if-then-else:** een beslissing waarbij de uitgevoerde actie afhangt van een voorwaarde.
- **Botsing of collision:** het moment waarop twee sprites of andere spelobjecten elkaar raken.
- **Variabele:** een benoemde waarde die het programma kan bewaren en veranderen, zoals een score.
- **Toestand of state:** alle waarden die samen beschrijven hoe het spel er op een bepaald moment voor staat, zoals posities, score en gewonnen of verloren.
- **Initialiseren:** een sprite of variabele bij de start een beginwaarde geven.
- **Resetten:** onderdelen van het spel terugbrengen naar hun afgesproken begintoestand.
- **Bug:** gedrag van een programma dat afwijkt van wat je verwachtte.
- **Acceptatiecriterium:** een controleerbare eis waarmee je bepaalt of een opdracht goed is uitgevoerd.

# ?? Opdracht 1 – Begrijpen en herkennen

Beantwoord de volgende vragen in je eigen woorden. Gebruik bij ieder antwoord een voorbeeld uit jouw eigen doolhofspel.

1. Wat is een sprite? Noem twee sprites uit jouw spel.
2. Wat zijn coördinaten? Geef de X- en Y-coördinaten van de startpositie van Giga.
3. Waarvoor heb je een lus gebruikt? Benoem ook welke instructies worden herhaald.
4. Welke voorwaarde controleert jouw spel bij een botsing met de muur en wat gebeurt er daarna?
5. Waarvoor heb je een als-dan-anders-blok gebruikt? Beschrijf wat er bij *waar* en bij *niet waar* gebeurt.
6. Welke variabele heb je gebruikt? Beschrijf de naam, beginwaarde en het moment waarop de waarde verandert.
7. Welke waarden of onderdelen moet jouw spel resetten als je opnieuw begint?

8. Beschrijf één bug die je tijdens het maken tegenkwam. Wat verwachtte je en wat gebeurde er werkelijk?

# ? Opdracht 2 – Communiceren met AI

Een goede opdracht voor AI wordt ook wel een **prompt** genoemd. Een goede programmeerprompt bevat voldoende informatie om het gewenste gedrag te begrijpen en achteraf te controleren.

## Onderdelen van een duidelijke prompt

1. **Context:** vertel dat je in Scratch een doolhofspel maakt en welke sprites belangrijk zijn.
2. **Begintoestand:** noem relevante startposities, beginwaarden en andere instellingen.
3. **Trigger:** beschrijf wanneer het gedrag moet beginnen.
4. **Conditie:** beschrijf welke controle waar of niet waar kan zijn.
5. **Actie:** beschrijf precies wat er daarna moet gebeuren.
6. **Toestandsverandering:** benoem welke posities of variabelen veranderen.
7. **Beperkingen:** benoem wat niet mag veranderen of gebeuren.
8. **Acceptatiecriteria:** geef controleerbare punten waarmee je kunt testen of het resultaat klopt.

**Te vage opdracht:**

Maak mijn doolhofspel moeilijker.

**Duidelijkere opdracht:**

Ik maak in Scratch een doolhofspel met de sprites Giga, Sleutel en Monster. Laat het Monster voortdurend horizontaal bewegen tussen de linker- en rechterrand van het speelveld. Wanneer het Monster een rand raakt, moet het omkeren. Wanneer Giga het Monster raakt, zet je Giga terug naar zijn startcoördinaten. De score en de positie van de Sleutel mogen daarbij niet veranderen. Het resultaat is goed als het Monster blijft bewegen, bij beide randen omkeert en iedere botsing Giga correct terugzet.

**Jouw opdracht:** Kies één onderdeel of uitbreiding van jouw spel en schrijf daarvoor zelf een duidelijke prompt. Gebruik alle acht onderdelen uit de lijst hierboven. Je hoeft deze prompt niet aan AI te geven.

## Een antwoord van AI controleren

AI kan een overtuigend antwoord geven dat toch niet goed werkt. Jij blijft daarom verantwoordelijk voor het begrijpen en testen van het resultaat.

- Controleer of je ieder voorgesteld Scratch-blok kunt uitleggen.
- Controleer of het antwoord aan al jouw acceptatiecriteria voldoet.
- Test normaal gedrag, bijvoorbeeld Giga die naar de sleutel beweegt.
- Test een grensgeval, bijvoorbeeld twee snelle botsingen achter elkaar.
- Beschrijf bij een fout opnieuw het verwachte en het werkelijke gedrag.

**Schrijf drie testgevallen bij jouw prompt:** een normale situatie, een situatie waarin de conditie niet waar is en een grensgeval.

# ? Reflectie

- Wat heb jij geleerd?
- Wat vond je leuk?
- Wat vond je lastig?
- Welk programmeerbegrip kun je nu het beste uitleggen?
- Welk begrip wil je nog beter leren begrijpen?
- Wat hoop je in de volgende module te leren?

# ? Inleveren

Beantwoord de vragen in je eigen woorden. Gebruik bij de begrippen concrete voorbeelden uit jouw eigen spel.

1. Lever je antwoorden op de acht begripsvragen uit opdracht 1 in.
2. Lever de duidelijke prompt in die je bij opdracht 2 hebt geschreven.
3. Lever de drie testgevallen bij jouw prompt in.
4. Lever je antwoorden op de zes reflectievragen in. Gebruik 2 tot 4 zinnen per antwoord.

--

---

Revision #2
Created 2026-08-18 11:12:55 UTC by Max
Updated 2026-08-18 11:16:22 UTC by Max