

# Scratch (12)

René Suiker

*In navolging van ongetwijfeld vele anderen wil ik beginnen met iedereen een gezond en gezellig 2021 te wensen. Zoals u ongetwijfeld ook hebt vernomen verkeert de vereniging in zwaar weer en hebben we moeite met het vinden van voldoende vrijwilligers, met name bestuursleden.*

## Mooie vereniging

Dat gaat mij echt aan het hart, ik heb zo'n twintig jaar deel uitgemaakt van het bestuur en draag de vereniging een warm hart toe. Ik hoop dan ook dat er onder de lezers van dit artikel nog mensen zijn, die bereid zijn voor het behoud van de vereniging de handen uit de mouwen te steken. Het is een mooie vereniging, met veel mensen met veel kennis en we zijn in staat mooie dingen te doen voor onze leden. Het zou toch eeuwig zonde zijn als we dat kwijt zouden raken. Dus als u het nog niet overwogen heeft, denk eens na over wat u voor de vereniging zou kunnen doen. En dan denk ik aan bestuurswerk, maar ook aan het schrijven van artikelen, zoals ik nu ook al jaren doe. Het is heel leuk en leerzaam.

## Zelf ook beter begrijpen

Toen ik aan deze reeks van artikelen over Scratch begon wist ik er nog niets van, maar door het aan anderen uit te leggen leer ik het zelf steeds beter begrijpen. Ik wil niet zeggen dat ik nu een expert ben, maar ik denk dat ik toch wel leuk mee kan komen.

En Scratch, daar gaan we het dus weer over hebben. De vorige keer had ik een heel uitdagende opgave verzonnen voor het huiswerk. Uitdagend, omdat u in feite een spel moest gaan maken. En dat had u 12 nummers geleden nog niet achter uzelf gezocht, maar ik denk dat ik alles heb aangereikt om dit mogelijk te maken en het zou toch leuk zijn als het u gelukt is.

Helaas heb ik geen meldingen gekregen, er is geen huiswerk ingediend, maar ik ga er toch vanuit, dat enkelen van jullie dat wel hebben afgerond. Waarom niet even via de Scratch-berichten een link verstuurd naar uw project? Nog even in de herinnering:

### Opgave 11.1:

- Zoek een plaatje van een doolhof
- Gebruik dat als achtergrond in je programma
- Zoek een leuk plaatje van een object om door het doolhof te loodsen
- Verzin een toetsenbordbesturing
- Maak een programma waarmee de gebruiker het object door het doolhof loodst
- Houd onderweg de tijd bij
- Telkens als de rand wordt geraakt gaat het object terug naar de startpositie
- Bij 5 keer herstart is het spel afgelopen
- Als het de gebruiker lukt om tot het eind te komen, volgt een muziekje en wordt de tijd in het groot getoond

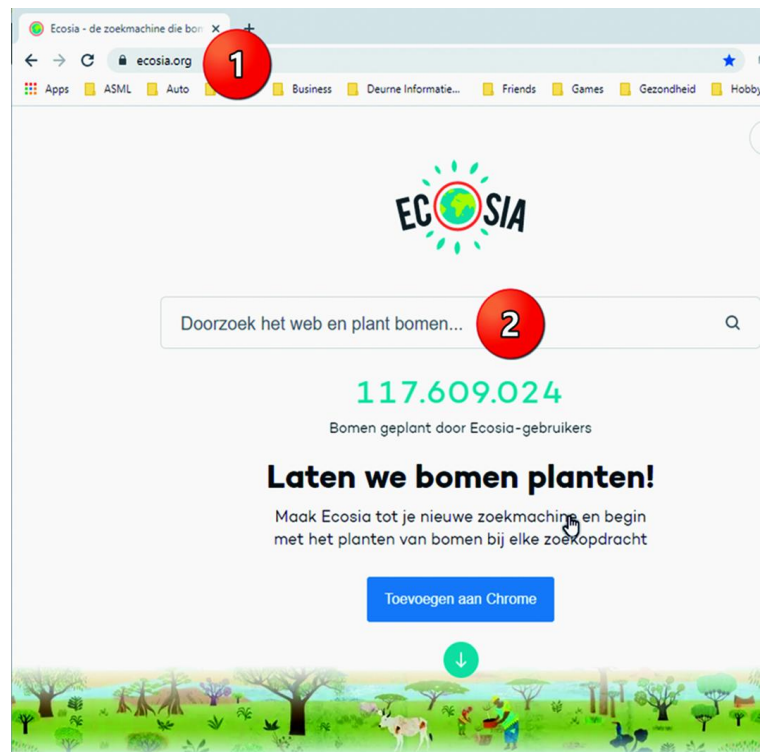
Hier komt het dus allemaal samen. Je mag op zoek naar een plaatje voor deel a, maar je mag natuurlijk ook zelf aan de slag met een doolhof tekenen. Je kunt dit zo complex of eenvoudig maken als je wilt en als je de smaak te pakken hebt, kun je zelfs nadenken over diverse levels in het spel.

Je zou kunnen beginnen met een eenvoudig doolhof, maar als mensen die binnen een bepaalde tijd hebben uitgespeeld, kun je een moeilijker doolhof starten. In feite is dit de essentie van heel veel games. Je maakt het haalbaar, zodat mensen niet afhaken; maar je maakt het ook uitdagend, zodat er wat te doen is.

In het kader van Scratch is het verstandig om de wanden van het doolhof in één kleur te houden, zodat je met een eenvoudige afvraag kunt zien of de rand geraakt wordt. Ook is het handig om het eindpunt van een andere kleur te voorzien, zodat je eenvoudig kunt afvragen of het eindpunt bereikt is. Je kunt alles van coördinaten laten afhangen, maar dat maakt het lastiger.

## Plaatje zoeken

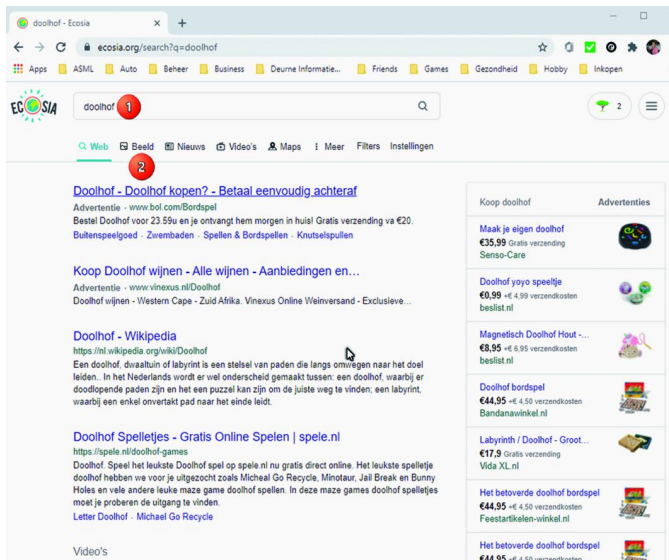
Goed, we beginnen. Een doolhof, om te beginnen één. We kunnen natuurlijk Google pakken, maar voor de verandering gaan we eens naar <https://www.ecosia.org/>; eens zien of dat ook een beetje werkt:



Figuur 1 - Ecosia - een andere zoekmachine

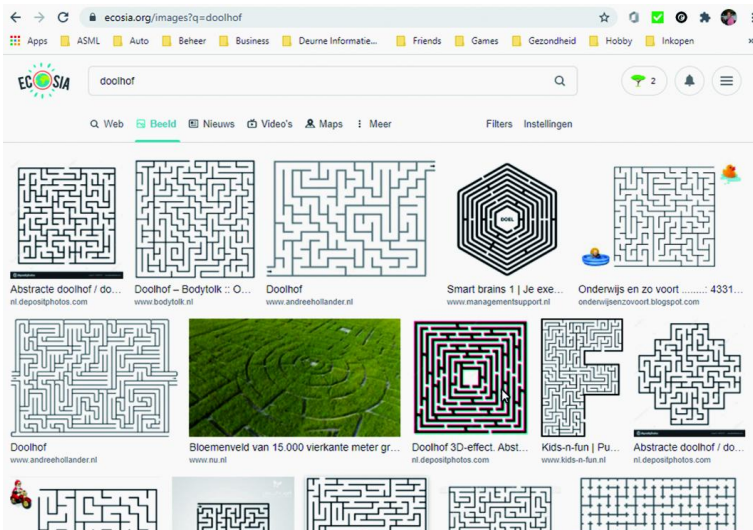
Bij (1) vul je dus [ecosia.org](https://www.ecosia.org/) in en bij (2) vul je doolhof in. Je krijgt dan de zoekresultaten te zien.

(Zie figuur 2 op de volgende pagina)



Figuur 2 - Zoekresultaten

Bij (1) zie je nog waar je op gezocht hebt; klik dan bij (2) op beeld en je krijgt een reeks van afbeeldingen te zien:



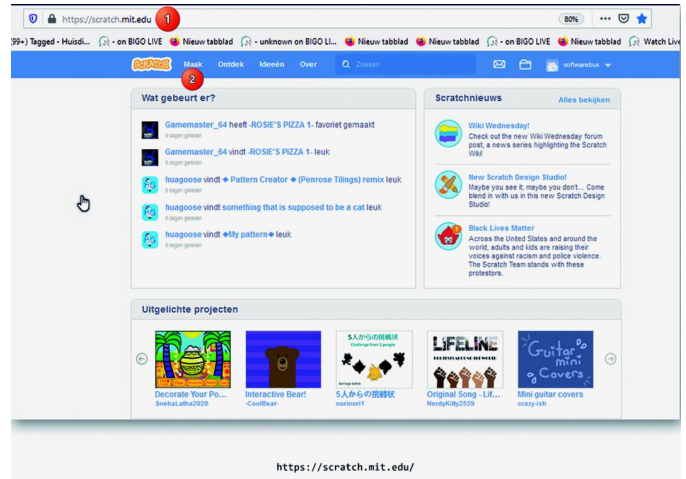
Figuur 3 - Afbeeldingen

Als je een mooie gevonden hebt, dan kies je die uit en sla je hem lokaal op. Dan ga je naar Scratch: <https://scratch.mit.edu/>

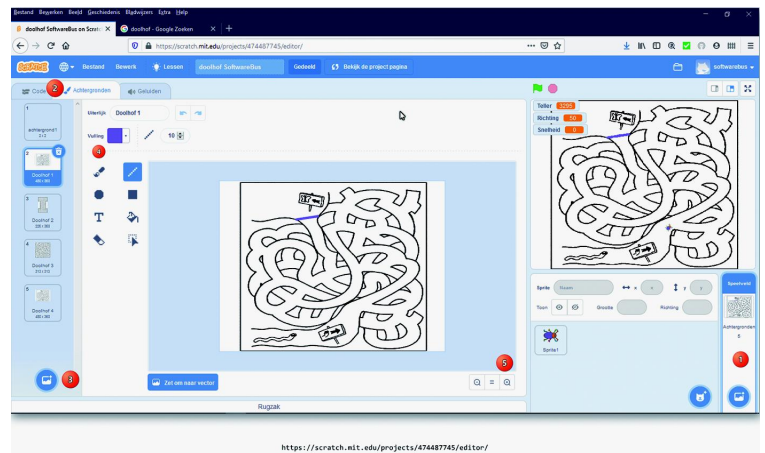
## Plaatje gebruiken

Log eventueel in als je nog niet ingelogd bent, maar wel een account hebt. Als je niet weet hoe dat moet, verwijst ik graag nog naar artikel 1 in deze reeks, een eeuwigheid geleden. Maar op website van CompUsers (<https://www.compusers.nl/>) is het artikel wel terug te vinden. Is zoeken te veel werk, dan is hier de juiste link naar de inhoud: <https://www.compusers.nl/inhoud-softwarebus-2018-5> Eenmaal binnen Scratch klik je op 'Maak' (2), bovenin de menubalk van Scratch. Je krijgt dan weer een nieuw, leeg project voorgeschoteld. (Zie figuur 4 'Maak')

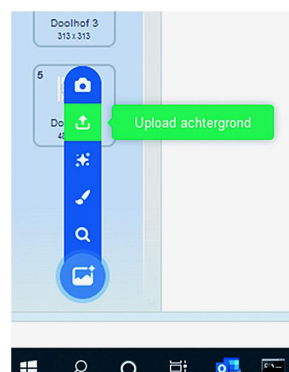
Je krijgt dan het standaard startscherm te zien, dat je intussen wel bekend voor moet komen. Maar we gaan een keer iets anders doen, want we gaan een keer met een achtergrond aan het werk die we actief moeten gaan gebruiken. In het Scratch scherm zie je weer de vertrouwde blokken, maar we gaan nu naar een nog minder vertrouwd onderdeel. Klik bij (1) hieronder op 'achtergronden': (Zie figuur 5 Achtergrond bewerken)



Figuur 4 - Klik op 'Maak'



Figuur 5 - Achtergrond bewerken



Figuur 6 - Upload achtergrond

Bij 2 zie je dan niet meer 'uiterlijken' staan, maar 'achtergronden' bij het middelste tabblad. Je ziet dan onder (2) de mogelijke achtergronden met naam voor dit programma. Als je er op één klikt, kom je in het scherm om de achtergronden te bewerken. Hier gaan we ons zojuist gevonden plaatje inzetten. Daartoe druk je op het knopje bij (3). Dan verschijnt er een uitschuifmenu zoals hier afgebeeld:

Daar kies je de tweede optie, 'upload achtergrond'. Dan selecteer je het plaatje vanaf de harde schijf en dan wordt de achtergrond beschikbaar in de balk onder (2) van het vorige plaatje (waar we weer naar teruggaan).

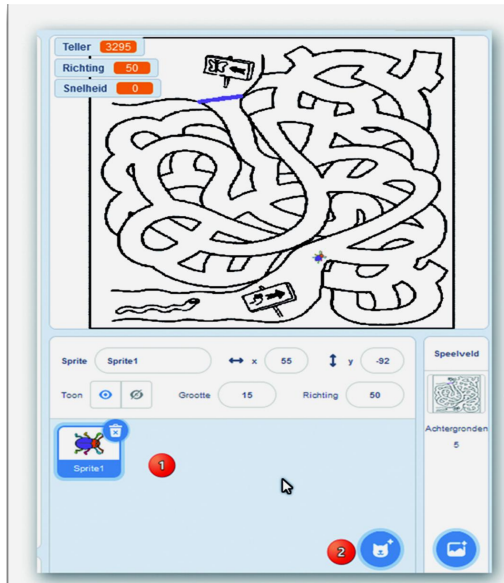
Vervolgens selecteer je het juiste plaatje in die balk en dan zie je het plaatje in het werkgebied. Bij (4) kies je een pen of een kwast en een kleur en dan teken je de eindstreep.

Om dat een klein beetje netjes te doen, kun je bij (5) het beeld een beetje opblazen, zodat je beter ziet wat je doet. In het plaatje zie ik hoe ik het redelijk rudimentair heb gedaan. Wil je een mooi spel maken, dan kun je de achtergrond natuurlijk zo mooi maken als je wilt.

Zoals gezegd, je kunt er ook voor kiezen om helemaal zelf je doolhof te ontwerpen, maar beseft wel, dat is een kunst op zich en daar kan ik je ook niet bij helpen.

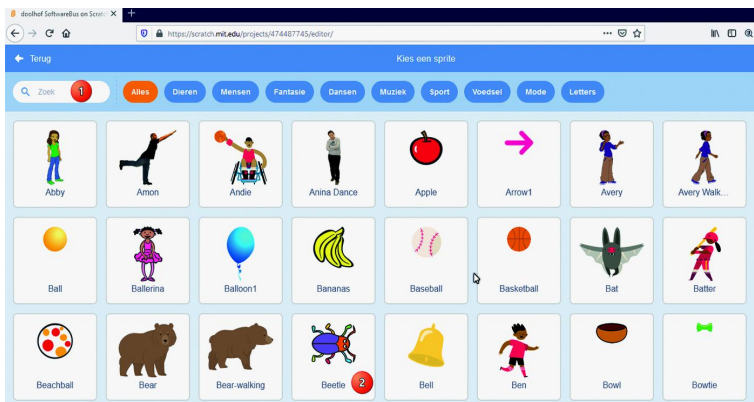
## Spelfiguur kiezen

De volgende stap in het proces (stap c van het huiswerk) is een leuk figuurtje te kiezen dat door het doolhof wandelt:



Figuur 7 - Spelfiguur kiezen

Je kiest dan rechts onder in niet meer voor achtergronden, maar klikt via (1) op het knopje (2). Hier kun je gewoon op klikken, als je er met de muis heen gaat ontvouwt zich een menu als in figuur 6 maar dat hebben we niet nodig, hoewel je natuurlijk ook een leuk spelfiguur kunt zoeken en downloaden. Het idee is hetzelfde, dus voor een spelfiguur wil ik uit de reeds beschikbare figuren kiezen, anders wordt mijn artikel weer te groot.



Figuur 8 - Spelfiguur selecteren

We kunnen dan heel veel figuren kiezen, op de balk bij (1) kun je de selectie beperken tot dieren, mensen, fantasie, enz. Je kunt hier ook een zoekterm invullen en dan beperk je het aantal spelfiguren nog meer. In mijn voorbeeld heb ik gekozen voor de Beetle (2). Maar je zou bijvoorbeeld net zo goed voor een raceauto kunnen kiezen. Heb je je keuze gemaakt, dan kun je onder 'uiterlijkheden' nog bewerkingen uitvoeren, of zelfs meerdere uiterlijken maken, teneinde het bewegen ook een visueel uiterlijk te geven, zoals we dat bij de mieren hebben gedaan. Ik ga dat nu niet herhalen, maar je kunt natuurlijk altijd teruglezen. Nu heb ik een achtergrond en een spelfiguur in mijn project geladen. Als je dat nog niet gedaan hebt, is het misschien handig je project op te slaan. Dat kan, zoals eerder aangegeven, alleen als je een account hebt.

## Toetsenbordbesturing kiezen

Hier valt niet echt iets te downloaden of zo, hoewel je natuurlijk vakliteratuur kunt raadplegen over hoe je zo iets zo goed mogelijk doet, op basis van de moderne gebruiker interactie (UX - User Experience) inzichten. Ik houd het in het kader van introductie in Scratch graag simpel en kom met de volgende besturing:

- Pijltje omhoog is snelheid verhogen
- Pijltje omlaag is snelheid verlagen
- Pijltje rechts is rechtsom draaien
- Pijltje links is linksom draaien
- Spatie is stilstaan

Ik zeg dus niet dat dit de weg, de waarheid en het leven is, ik ben nu uitgegaan van de speelfiguur, maar je kunt natuurlijk ook je oriëntatie op je speelveld als uitgangspunt nemen, of juist een volledig afwijkende besturing verzinnen. Je zou deze ook als extra moeilijkheidsgraad in een latere fase van je spel kunnen veranderen.

## De tijd bijhouden

Ja, ik weet het, het programma schrijven kwam hier nog voor, maar ik wil even wat beschouwing delen, voordat ik het programma uitleg. Even voor de duidelijkheid, ik ben ook geen Scratch-expert, ik heb het geleerd bij het schrijven van deze artikelen. En het hele huiswerk dat ik hier beschrijf kost veel meer tijd om het te beschrijven dan om het te doen, want het hele programma, inclusief werk aan achtergronden e.d. kostte me ongeveer een half uurtje. Het schrijven van dit artikel een veelvoud daarvan. Wat wil ik hiermee zeggen: dat je niet bang moet zijn om te beginnen, het lijkt moeilijker dan het is.

In het begin zag ik de mogelijkheid om Scratch de tijd bij te laten houden, op een heel hoog tempo. Daar zaten wel wat beperkingen in en het gaat eigenlijk om het idee, dat we het sneller doorlopen van het doolhof in een score tot uiterlijk willen zien komen. Ik heb er uiteindelijk voor gekozen om dit met een soort stappenteller te doen. Hoe lager de score, des te beter. Nogmaals, je kunt de klok gebruiken, maar het nadeel is dat deze niet stilstaat als je klaar bent, dus daar moet je allerlei voorzieningen gaan treffen die buiten het kader van dit artikel vallen. Ik kies dus even voor een simpele oplossing, maar misschien is het wel leuk om hier in een vervolg eens naar te kijken.

## Het programma schrijven

Dat is dus dankzij de manier waarop Scratch werkt echt een fluitje van een cent. Als ik kijk wat ik hiervoor vroeger in Basic had moeten doen, dan had ik het echt niet in een half uurtje afgerond, in Pascal ook niet. Misschien in een moderne taal wel, maar daar ben ik nog niet in thuis. Maar dit programma kan in mijn eenvoudige versie opgedeeld worden in 6 blokjes: één hoofdprogramma en verder 5 blokjes voor de afwikkeling van de besturing. Klinkt misschien nog steeds ingewikkeld, maar dat is het dus echt niet.

Het hoofdprogramma bestaat weer uit de bekende twee delen:

- Initialisatie
- Besturingslus

De initialisatie is rechttoe rechtaan, zoals jullie van me gewend zijn.

We beginnen (1), zoals gebruikelijk, met de 'gebeurtenis' dat de groene vlag wordt ingedrukt. Dan (2) laten we de sprite verdwijnen, zetten alle effecten uit (we weten niet of er nog iets loopt), zetten de achtergrond zoals we die willen hebben (je weet maar nooit wat er staat) en we passen de grootte van de speelfiguur (de 'sprite') aan het speelveld aan. In dit geval, op een groot scherm, is 15% van de grootte prima, dan raak je niet gelijk al de rand en kun je nog wel een beetje zien wat de besturing doet. Vervolgens (4) verplaatsen we de sprite (daar hoort deze code bij, weet u het nog?) naar de startpositie. Tijdens het programmeren zet je eerst de sprite op de startpositie, gewoon door 'm op te tillen en te verplaatsen. Dan pak je het blokje vanuit 'beweging' en dan staan de juiste x en y al ingevuld. Zo makkelijk kan het zijn.



Figuur 9 - Initialisatie

Bij (5) zetten we de variabelen op een beginwaarde. In dit geval begin ik met een snelheid 0, een stappenteller start op 0 en de richting is 60 graden, gebaseerd op de richting van het doolhof en de gewenste startpositie. Vervolgens (6) richten we de sprite naar die richting, met behulp van de variabele. Dit, omdat we in de loop van het spel de richting steeds willen veranderen, op basis van de besturing. Dan laten we de sprite verschijnen en zijn we klaar voor de actie.

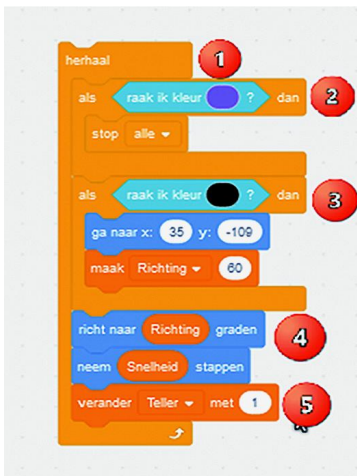
Achteraf denk je ‘Waarom combineer ik blokje (4) en (6) niet?’ Dat is een logische gedachte en dat heb ik dan ook gedaan, in de uiteindelijke versie. Beseft wel dat je dan eerst de variabelen moet initialiseren en dan pas de sprite moet positioneren, anders weet je niet in welke richting de sprite start. Kan ook weer een extra moeilijkheid zijn in een hoger level, maar voor nu: houd het even simpel.

Ook de besturing is heel eenvoudig. Ten aanzien van programmabesturing bestaat het programma uit één herhaling (1). Dat is alles.

We hebben geen conditionele herhaling, alleen, we kunnen wat dingen doen binnen deze lus en één ervan is om het programma te beëindigen. Daar beginnen we gelijk mee, in stap (2).

Daar vragen we ons af of we de eindstreep bereikt hebben, en zo ja, dan stopt het programma. In deze eenvoudige variant is dat het ook.

Als we nog niet bij de eindstreep zijn, kijken we of we een zwarte lijn raken (3). Zo ja, dan gaan we weer naar de startpositie en startrichting. De snelheid blijft zoals die was.



Figuur 10 - Besturingslus

Elke keer dat de lus doorlopen wordt regelt blokje (4) de verplaatsing. Initieel hebben we snelheid 0, maar die verandert in de loop van het spel. Bij blokje (5) wordt de teller verhoogt, zoals gezegd geen klok, maar hij telt het aantal keren dat de lus doorlopen wordt. Als je de sprite sneller laat bewegen heb je dus minder stappen door het doolhof nodig (maar wel meer kans om de rand te raken).

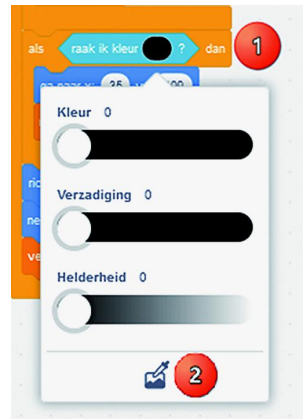
## Zijstapje

Je zit in blokje (2) en (3) een specifieke kleur. Om die goed te zetten kun je de kleurkiezer gebruiken:

Als je op de kleur klikt (1), dan komt een pop-up tevoorschijn, waar je de kleur kunt aanpassen. Daarbij zit ook een knopje, om een kleur te zoeken op je achtergrond. Als je daarop klikt, krijg je een cursor te zien, die je over een vergroting van de achtergrond kunt bewegen. Dan kun je vervolgens de juiste plek selecteren en daar de kleur ophalen.

## Besturing

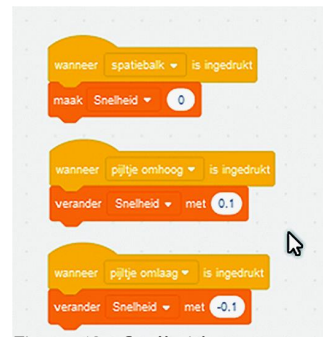
Tot dusver dus één lus, waarbij de beweging wordt gestuurd op basis van twee variabelen: snelheid en richting. Die worden door het toetsenbord bestuurd; die besturing maakt geen onderdeel uit van de hoofdlus, maar vindt plaats met behulp van extra blokjes, die getriggerd worden door het indrukken van een toets.



Figuur 11 - Kleurkiezer

Hier kom je tijdens het proefdraaien achter.

Voor de richting is het bijna net zo simpel, we bewegen hier in stapjes van 5 graden. Gaan we daarbij door de 0/360, dan kiezen we weer 0 respectievelijk 360 graden. Niet dat dit echt moet, maar ik weet nog niet hoe Scratch reageert op bv. 36005 graden.



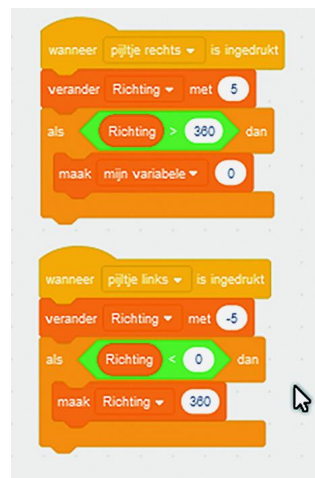
Figuur 12 - Snelheid

We hebben stapjes H en I van het huiswerk nog niet gedaan. Dat geef ik nu mee als nieuw huiswerk. Daarnaast wil ik dat jullie kijken wat er nog geoptimaliseerd kan worden én wat een risico is van deze aanpak.

Je kunt het hele project nu bekijken op:

<https://scratch.mit.edu/projects/474487745/fullscreen/> en

je kunt er dan ook een kopie van maken om er zelf aanpassingen op te doen.



Figuur 13 - Richting

Je kunt met het groene vlaggetje het spel starten, met de pijlen het beestje besturen en probeer dan maar eens de eindstreep te halen.

In een speelbare variant, die uitontwikkeld is, zou ik de variabelen ‘Richting’ en ‘Snelheid’ buiten beeld houden, want die leiden af, maar tijdens het debuggen is het handig om dit in de gaten te houden.

Ik denk dat jullie allemaal in staat zouden zijn om dit ook zelf te maken.

Misschien niet in een half uur, maar het lijkt allemaal moeilijker dan het feitelijk is.

**Veel Scratch-plezier!**