

6.14 3D-doolhof

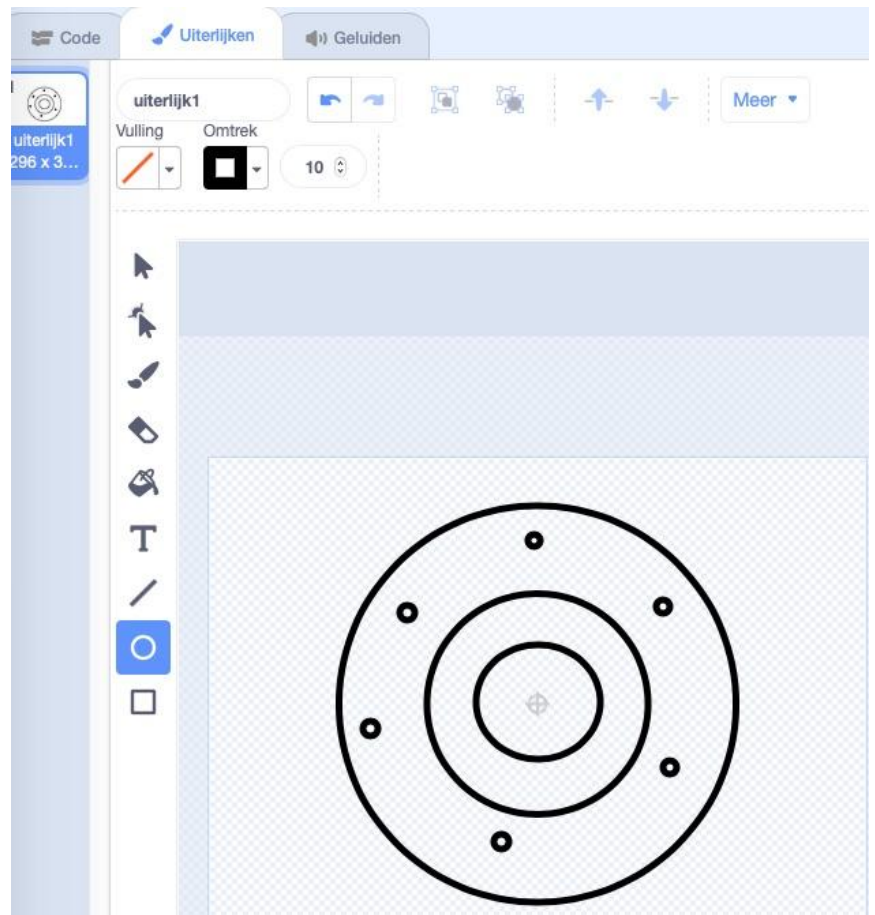
Link naar video met Engelstalige uitleg: https://youtu.be/8KvN_GxYR6k

We gaan een supergaaf 3D-spel maken. Het is een stuk gecompliceerder dan andere spelletjes. Maar als je doorzet, zal het wel lukken!

1. We tekenen een doolhof:

1.1 Verwijder de kat en teken zelf als nieuwe sprite een cirkelvormig doolhof. (**geen achtergrond maken** he!!!) Als je de shiftknop (= dat pijltje waarmee je hoofdletters maakt) ingedrukt houdt op je toetsenbord, kan je een perfecte cirkel maken.

- Teken nog twee kleinere cirkels zoals op de tekening hiernaast.
- En ook nog van die hele kleine cirkeltjes.
- Kijk ook op de tekening om de vulling, omtrek en teken dikte juist te krijgen.



1.2 Klik op 'Zet om naar bitmap'. Selecteer de Gum.

- Maak met de gum drie gaten in je doolhof.



- Geef je cirkelvormig doolhof de naam: Level 1

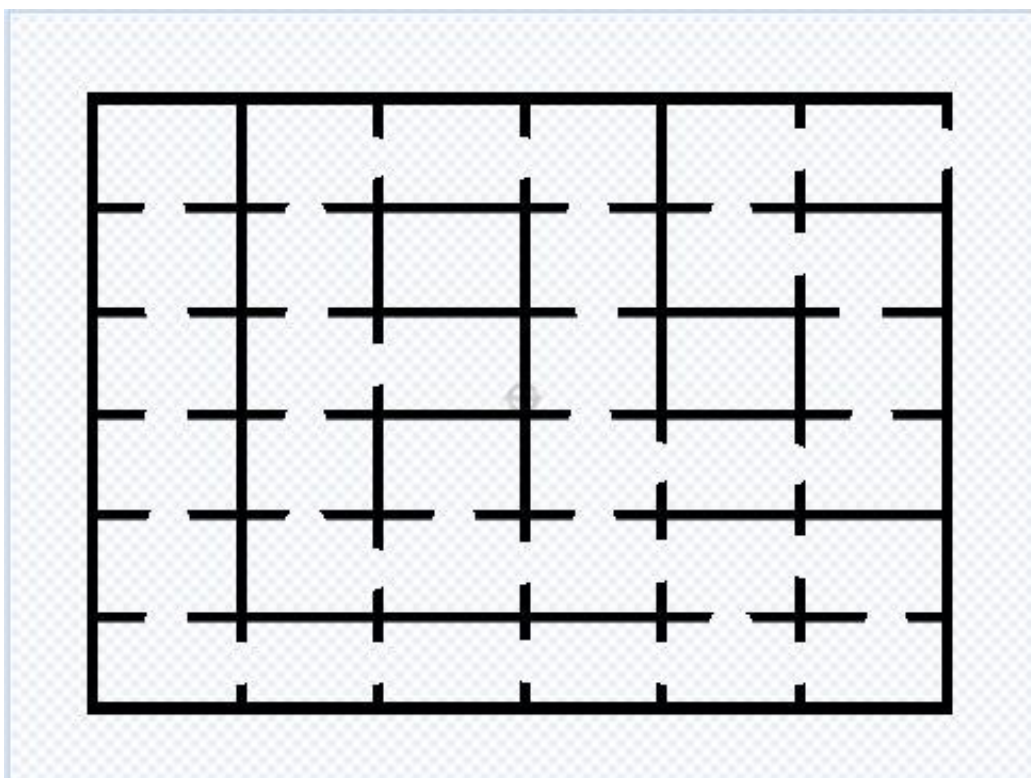


Nu wordt het wat moeilijker...maar hey, dat kan jij ook wel aan. We gaan namelijk een rechthoekig doolhof maken.

1.3 Zorg voor een Level 2. Of een nieuw uiterlijk van dezelfde sprite. (dus **geen** nieuwe sprite maken en al zeker **geen** achtergrond ofzo!)



- Teken het volgende rechthoekige doolhof. Zorg ervoor dat je twee Uiterlijken pal in het midden staan. Gebruik hiervoor de rechthoek, lijn en gum.



1.4 Als op de groene vlag wordt geklikt, gaat onze sprite 'doolhof' starten met uiterlijk 'level 1'.



Laat ons nu de speler programmeren. Ik moet jullie nog melden dat we het spel eerst in 2D gaan maken en nadien pas in 3D. Want 3D is een nogal complex gedeelte.

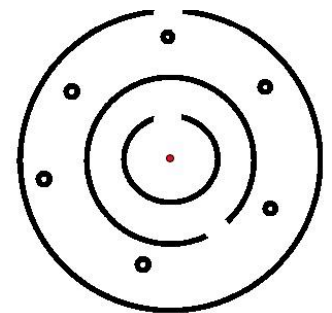
Onze speler gaat een kleine Dot worden. Dat is gewoon een punt..

1.5 Maak de speler. Ga naar 'Kies een sprite' en teken zelf het punt. Je mag je 'dot' iets groter maken, maar echt maar een klein beetje groter. Ik ga de speler ook een neus geven, zo weet ik in welke richting hij kijkt.



Code van de 'speler': Als op de vlag geklikt wordt, moet je 'speler' starten in het midden van je doolhof.

Plaats je 'speler' eerst in het midden van je doolhof op het speelveld. De coördinaten van de x- en y-as zetten zich dan vanzelf goed. Typ dus niet zomaar de cijfers van hiernaast over. Bij jou kunnen die anders zijn, omdat we elk ons eigen doolhof maken.



- We gaan er voor zorgen dat je 'speler' zich **richt naar** het zuidwesten, zodat hij niet direct de uitgang in het noorden kan zien. Dat maakt de start van het spel wat spannender.



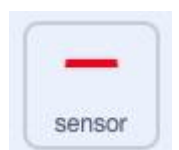
- Hoe gaan we onze speler laten bewegen? Als we de pijltjestoets naar omhoog in duwen, dan moet onze speler de hele tijd 1 stap zetten.
- Dupliceer deze blokken en doe hetzelfde voor de andere pijltjestoetsen.
- Let wel op: voor de pijltjestoetsen links en rechts moet onze speler 5 graden kunnen draaien!
- Op deze manier kan onze speler zich gemakkelijk bewegen. Er is echter nog een probleem. Hij kan gewoon door de muren van het doolhof heen. Daar gaan we nog iets aan moeten doen.



- We gaan dat probleem op een gemakkelijke manier oplossen. Als de speler het doolhof raakt dan moet hij een stap terug zetten. -1 wordt dan 1 en 1 wordt -1!

1.6 De sensor: als we het doolhof verlaten dan moeten we daar een sensor plaatsen, zodat ons spel weet dat we naar het volgende level gaan. Die sensor plaatsen we aan de uitgang van ons eerste level.

- Maak een nieuwe sprite 'sensor'. Teken een kleine rode lijn die net in de opening past van de uitgang van je doolhof. Deze sensor mag niet zichtbaar zijn tijdens je spel, maar nu effe wel.



- **Selecteer nu de sprite 'speler'.**

Programmeer: wanneer de speler de sensor raakt, mag hij doorgaan naar level 2. Plaats deze code onderaan de voorgaande code van de speler.



- Bij 'Waarnemen' vind je code die we nog nooit gebruikt hebben. Deze gaan we wat aanpassen naar 'uiterlijk#' van 'doolhof'.



- Steek dit laatste blok in het Functieblok 'is gelijk aan'. Zo krijgen we: als het uiterlijk van het doolhof gelijk is aan doolhof 1. Plaats dit blok in het lege vakje na de 'en'.



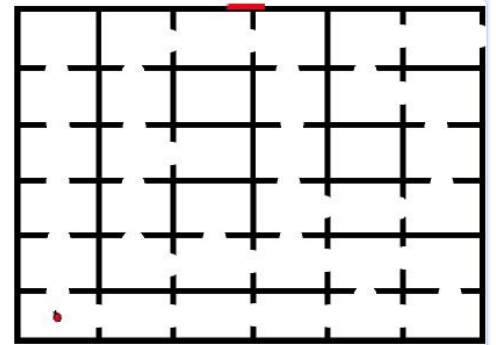
- Als volgt maken we een 'zend signaal' blok met als naam 'Level 1 is klaar'.



- **Selecteer je sprite 'doolhof'** en laat daar het signaal toekomen. Zorg er ook voor dat het 'uiterlijk' van je doolhof, overschakelt naar 'Level 2'.



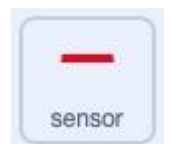
- **Selecteer de sprite 'speler'.** Je speler moet bij het doolhof met Uiterlijk 'level 2' links onderaan in het vakje starten.



- Wanneer de speler het signaal 'Level 1 is klaar' ontvangt, laat je hem starten in dat vakje linksonderaan. Je kan hem ook richten naar een kant waar geen uitgang te zien is.



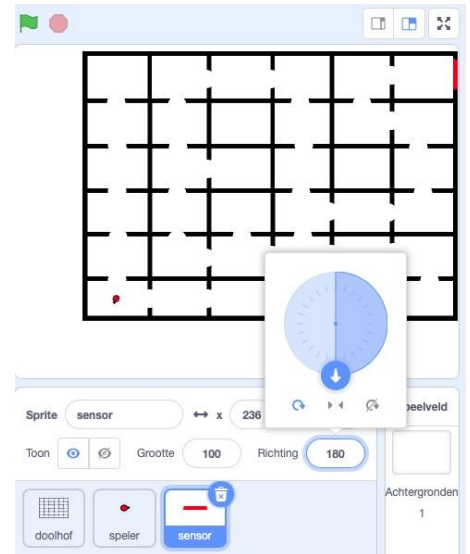
- **Selecteer de sprite 'sensor'.** Op dit moment staat de sensor nog aan de uitgang van 'level 1'. Dus....werk voor de boeg.



- Als het spel start in level 1, moet de sensor bovenaan op het doolhof staan aan de uitgang en hij moet zich richten naar 90 graden.



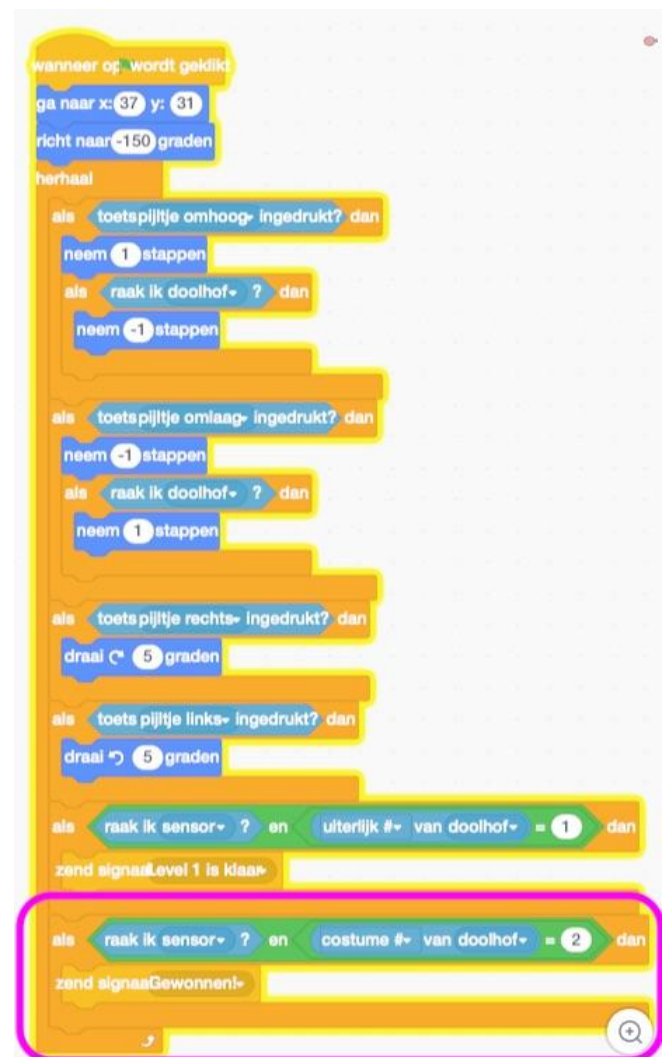
- Als het spel verandert naar level 2, moeten we de sensor ergens anders plaatsen, namelijk aan de rechterkant, aan de uitgang van Uiterlijk 'level 2'. Omdat die uitgang verticaal ligt, moeten we hem ook nog draaien en nadien tegen de uitgang plaatsen.



- De code hiervoor is:



- **Selecteer de sprite 'speler':** als we in 'level 2' zijn, en we raken de uitgang (sensor) van ons doolhof, dan zijn we gewonnen! Je kan nu natuurlijk zelf nog enkele extra levels toevoegen, maar wij houden het op deze twee.
- Maak een nieuw signaal met de naam 'Gewonnen!'



1.7 Maak zelf een nieuwe sprite 'Gewonnen!' en selecteer deze.



- Wanneer op de vlag wordt geklikt, moet deze sprite 'gewonnen!' op het midden van het speelveld staan en verdwijnen.
- Wanneer het signaal 'gewonnen!' wordt ontvangen, maken we de grootte van 'gewonnen' gelijk aan nul en tonen we hem. Maar je ziet hem nog niet ;-)
- En nadien gaan we hem 20 keer vergroten. Zo gaat hij zeker opvallen.



We zitten nog met een probleempje. Als de speler de sensor raakt in level 2, dan voert de code van de sprite 'gewonnen' zijn functie niet volledig uit. Het woord 'gewonnen!' groeit niet. De code wordt maar gedeeltelijk uitgevoerd. Als je speler de sensor raakt, dan wordt het signaal 'Gewonnen!', telkens opnieuw herhaald. 'Gewonnen' start deze code elke keer opnieuw.

OPLOSSING: De sensor moet **wanneer deze het signaal 'gewonnen' ontvangt, verdwijnen**. Zo kan de code eenmalig volledig uitgevoerd worden ipv telkens opnieuw half.

Dus...selecteer de sprite sensor en plaats de volgende code.



- De sensor verdwijnt als hij het signaal 'gewonnen' ontvangt, maar moet ook terug **verschijnen** wanneer we het spel starten natuurlijk.



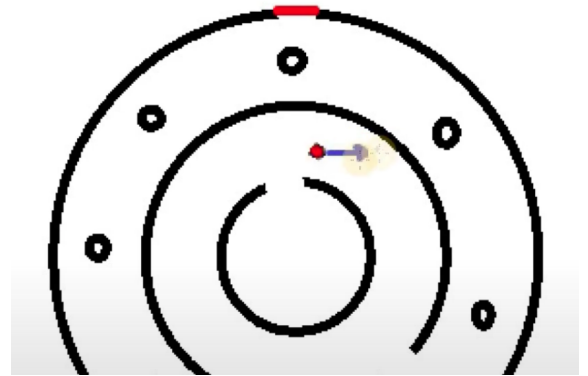
TEST JE SPEL UIT EN GA NA OF ALLES NAAR BEHOREN WERKT!

Zo... je 2D doolhof is klaar! Nu rest ons nog het moeilijkste gedeelte. We gaan dit 2D spel omzetten naar een 3D spel! Het meest complexe gedeelte van deze oefening.

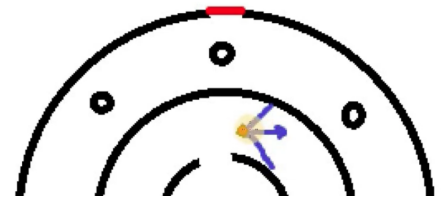
In de Engelstalige video wordt dit aanschouwelijk uitgelegd. Omdat beelden soms meer zeggen dan woorden, raad ik jullie dan ook aan om de video eens te bekijken. Het 3D gedeelte start op minuut 13:47. https://youtu.be/8KvN_GxYR6k

3D doolhof:

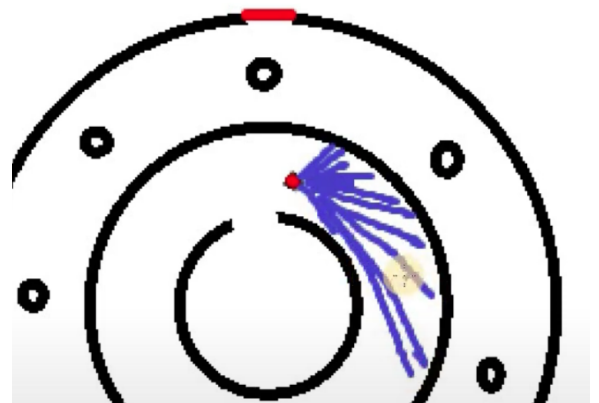
- Stel dat je speler kijkt in de richting van het blauwe pijltje. Wij als mens zien meer dan het punt waar we naar kijken.



- We zien ook de dingen die naast dat punt te zien zijn.

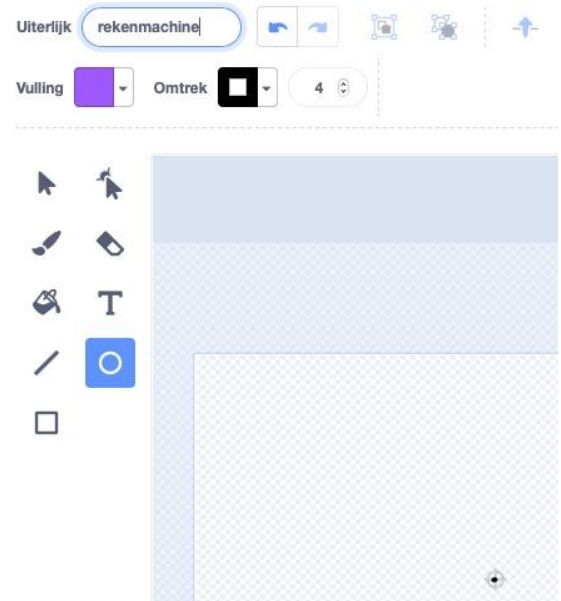


- Bij een 3D spel moeten we niet alleen meer zien, we moeten ook weten hoe ver elk gedeelte van de muur van het doolhof verwijderd is van onze speler. Hoe doen we dat? We gaan een andere sprite maken die we de 'rekenmachine' gaan noemen. Deze sprite zal de hele tijd alle punten die in een breed scala voor de speler te zien zijn, moeten berekenen. Hij gaat dat ook heel snel en automatisch moeten uitvoeren. Die informatie gaan we nadien gebruiken om de 3D beelden te tekenen. Al die info moet hij ook nog eens opslagen in een lijst. (Weet je nog? Bij **variabelen** kunnen we deze lijst maken)



- Op onze lijst gaan wel 100 nummers te zien zijn omdat we een dekking van wel 100 graden gaan maken. Met die informatie zal ons programma nadien de 3D muren kunnen tekenen.
- Laten we aan de code beginnen....klaar?

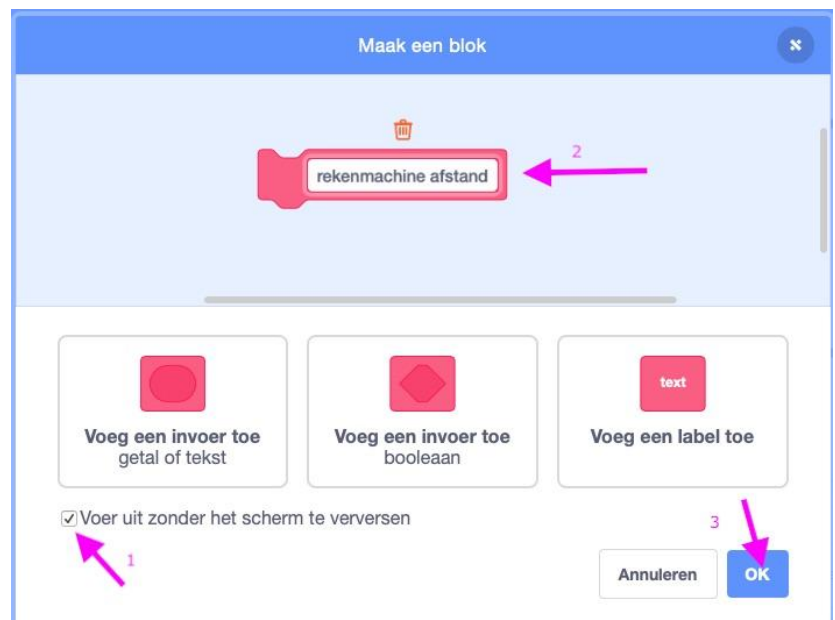
2.1 Maak een nieuwe sprite 'rekenmachine'. Het zal een super smal puntje (dot) worden. Plaats dat in het midden van je tekenblad.



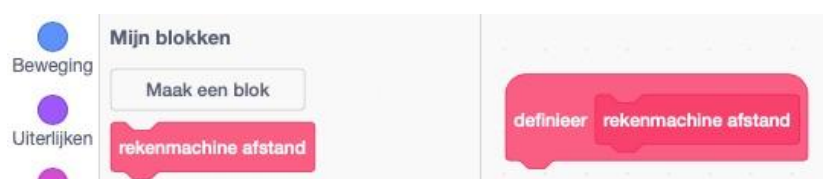
- **Selecteer de sprite 'rekenmachine'.** Als code gaan we een eigen '**blok**' maken. Selecteer '**Mijn blokken**' en klik nadien op 'Maak een blok'.



- Zet een vinkje bij 'Voer uit zonder het scherm te verversen'. (1) Zo kan je rekenmachine super snel werken.
- Geef je blok de naam (2): 'rekenmachine afstand'.
- Druk nadien op '**OK**' (3).



- De foto hiernaast laat het zelfgemaakt blok zien nadat je op 'OK' hebt geklikt.

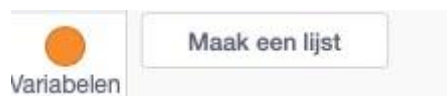


Wat gaat dit blok nu doen? Laten we het eerst definiëren. Want nu staat er nog geen code onder en werkt dit blok bijgevolg nog niet.

- **Maak een variabele en noem deze 'afstand'**



- **En we maken ook een 'lijst', die geven we de naam 'afstanden'.**



Wanneer we bewegen in de richting van de muur, gaan we deze informatie opslagen in deze lijst.

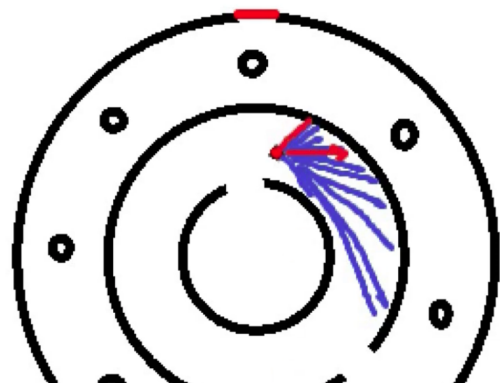
- We starten onder ons **zelfgemaakt blok** met het blok '**verwijder alle van afstanden**' en we gaan ook de **afstand 0 maken**. Zo maken we een propere start zonder gegevens op de variabele.



- Deze 'rekenmachine afstand' moet de positie van onze sprite 'speler' kunnen volgen. Hij moet zich richten naar dezelfde richting als de speler.
- Bij 'Waarnemen' vind je onderstaand blok. (achtergrond van speelveld). Plaats dat in het 'richt naar 90 graden' blok van hiernaast op de foto en verander waar nodig.



- Nadien moet 'rekenmachine afstand' 50 graden kunnen draaien. Waarom? Wel de speler richt zijn blik recht voor zich uit, maar de 'rekenmachine afstand' moet meer doen, die moet een hele hoek van 100 graden berekenen, waarvan 50 graden naar de linker kant toe. (zie de twee rode strepen op de foto hiernaast).



- Plaats de code 'draai naar links 50 graden' onderaan de vorige blokken.



- We zijn klaar voor het zwaardere werk. We gaan 100 keer de volgende code herhalen: 'neem 1 stappen' en verander de afstand met 1, zodat we onthouden hoeveel stappen we zetten om een bepaalde afstand te overbruggen. Dit gaan we herhalen tot we een muur in het doolhof raken of tot we een afstand moeten overbruggen van meer dan 400 pixels (als we bijvoorbeeld door een opening in het doolhof moeten).
- Nadien gaan we 'afstand' toevoegen aan 'afstanden'. Dat is die beruchte lijst waar we het eerder over hadden. Als onze speler dicht bij een muur staat, gaan de cijfers uit de lijst kleiner zijn dan wanneer de speler ver van een muur staat.
- Nadien kunnen we terug naar de 'speler' gaan en 1 graad naar rechts draaien.
- En de afstand 0 maken, zodat de informatie van de variabele wordt gehaald.
- En dit alles gaan we 100 maal herhalen.



Als je je speler dicht bij een muur plaatst en je klikt op dit blokje dan zie je dat de nummers in je lijst laag zijn. Als je nu je speler verder van een muur plaatst en je drukt op hetzelfde blokje, dan zullen de nummers in je lijst hoger zijn.



Wanneer willen we de afstand van de speler tot de muur berekenen? Elke keer als we bewegen! Of we nu vooruit gaan, achteruit gaan of draaien. Elke keer!

2.2 Selecteer de 'speler'.

- Maak een **nieuw bericht** bij 'Gebeurtenis' en geef het als naam: 'bereken de afstand'.
- We gaan de afstand berekenen wanneer we een stap achterwaarts of voorwaarts bewegen, naar rechts of naar links draaien.
- In al deze gevallen moeten we de afstand opnieuw berekenen omdat we in al deze gevallen kijken naar een andere plaats of omdat we bewogen hebben. En we moeten dat steeds opnieuw doen.
- Plaats een **'zend signaal 'bereken de afstand''** blok op de aangeduide plaatsen. (zie foto hiernaast).



2.3 Selecteer de sprite rekenmachine en plaats er de code die je op de foto omcirkeld ziet.

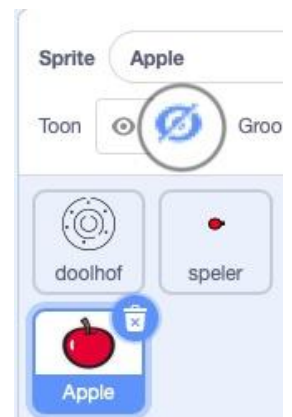
- De 'zend signaal **'bereken de afstand'** ontvang' blokken die we daarnet geplaatst hebben, gaan we bij de sprite rekenmachine laten toekomen. Elke keer als er gegevens ontvangen worden, zie je de data in je lijst veranderen. We hebben met deze een methode ontwikkeld waarmee we instant afstanden kunnen berekenen.



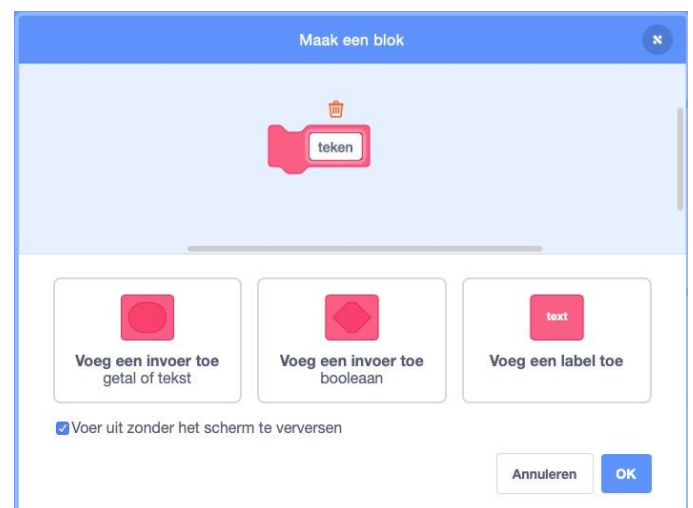
- Nu moeten we met deze methode de afstanden transformeren naar een 3D doolhof. Dat lijkt verschrikkelijk moeilijk, maar uiteindelijk zal je zien dat het niet zo gecompliceerd is. Het is zeker haalbaar. Laten we er aan beginnen.

2.4 Maak een nieuwe sprite die zal tekenen.

- Je mag hiervoor eender welke sprite uit de bib gebruiken, want we gaan hem toch laten verdwijnen. Wij kiezen de Appel. Vink hiervoor naast 'toon' het doorstreepte oog aan.



- **Selecteer de 'Apple'**, ga naar '**mijn blokken**' en klik op 'maak een blok'. (zie pag 11). Zet ook hier een vinkje bij 'Voer uit zonder het scherm te verversen' en noem het blok 'teken'.



Wat moet dit blok nu doen?

Als de speler dicht bij de muur staat, dan gaat het blok 'teken', grote lijnen tekenen die de muur uitbeelden. Deze lijnen zullen dicht bij elkaar staan zonder dat er ruimte tussen is. Als de speler verder weg staat van de muur, dan zullen de verticale lijnen kleiner zijn.



- Laten we het blok 'teken' definiëren.
- Maak een **variabele** en geef deze de naam 'teller'. We leggen later wel uit waar deze voor zal dienen.

- Om te kunnen tekenen, moeten we via 'Uitbreiding' de 'Pen' openen.

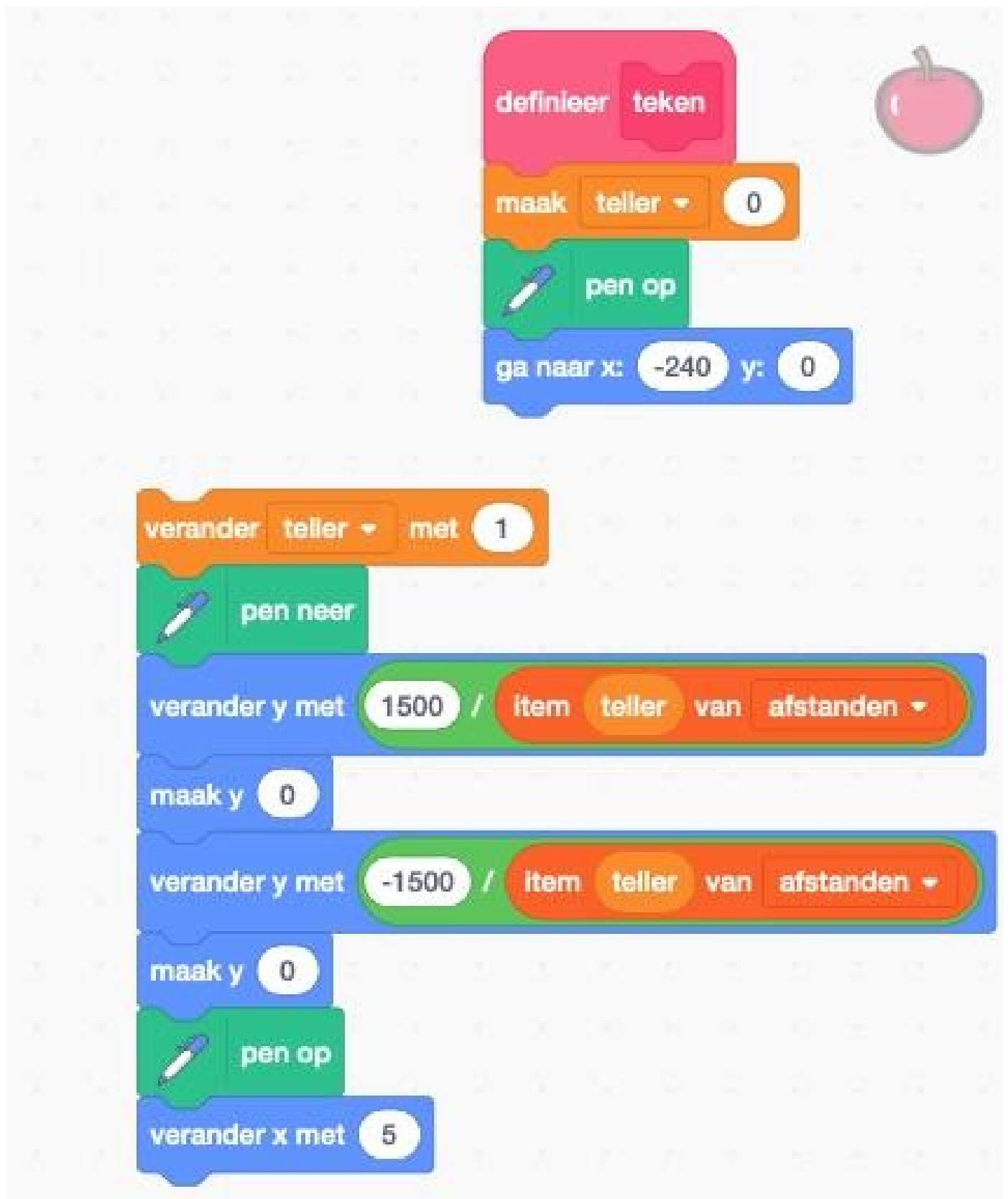


voeg een uitbreiding toe



De code die op deze pagina wordt besproken, vind je op pagina 19.

- Om te kunnen tekenen, starten we met 'pen op'.
- We gaan tekenen van links naar rechts. Hiervoor neem je het blok 'ga naar x:-240 y: 0'. *Deze cijfers juist overnemen!*
- We gaan opwaartse lijnen en neerwaartse lijnen tekenen.
- Plaats het blok 'maak 'teller' 0.
- Plaats het blok 'verander teller met 1'. Dit blok zal het eerste element uit je lijst tekenen.
- Plaats het blok 'pen neer'.
- Plaats het blok 'verander y met 10'. Dat is belangrijk. Als de muur dicht bij de speler staan is deze groot. Dat is ook zo in het echte leven. In dit geval moeten we bij **Functies** een **gedeeld door** blokje nemen. We gaan 1500 delen door het nummer dat op de lijst staat. Wat wil dit nu zeggen? Als de afstand bijvoorbeeld 100 is, dan betekent dit, dat de muur ver is. $1500 \text{ gedeeld door } 100 = 15$. De lijn zal dus klein zijn, omdat de muur ver is. de muur is 100 pixels verder. Als de afstand 1 is, dan raakt de speler bijna de muur. Dit nummer 1500 gedeeld door 1 is 1500. Deze lijn gaat zo groot zijn dat hij buiten je speelveld terecht komt.
- We gaan nu terug naar het centrum met het blok 'maak y 0', maar deze keer gaan we naar beneden ipv naar boven. Om die reden maken we van de 1500 deze keer -1500.
- Hierna moet de **pen terug op**.

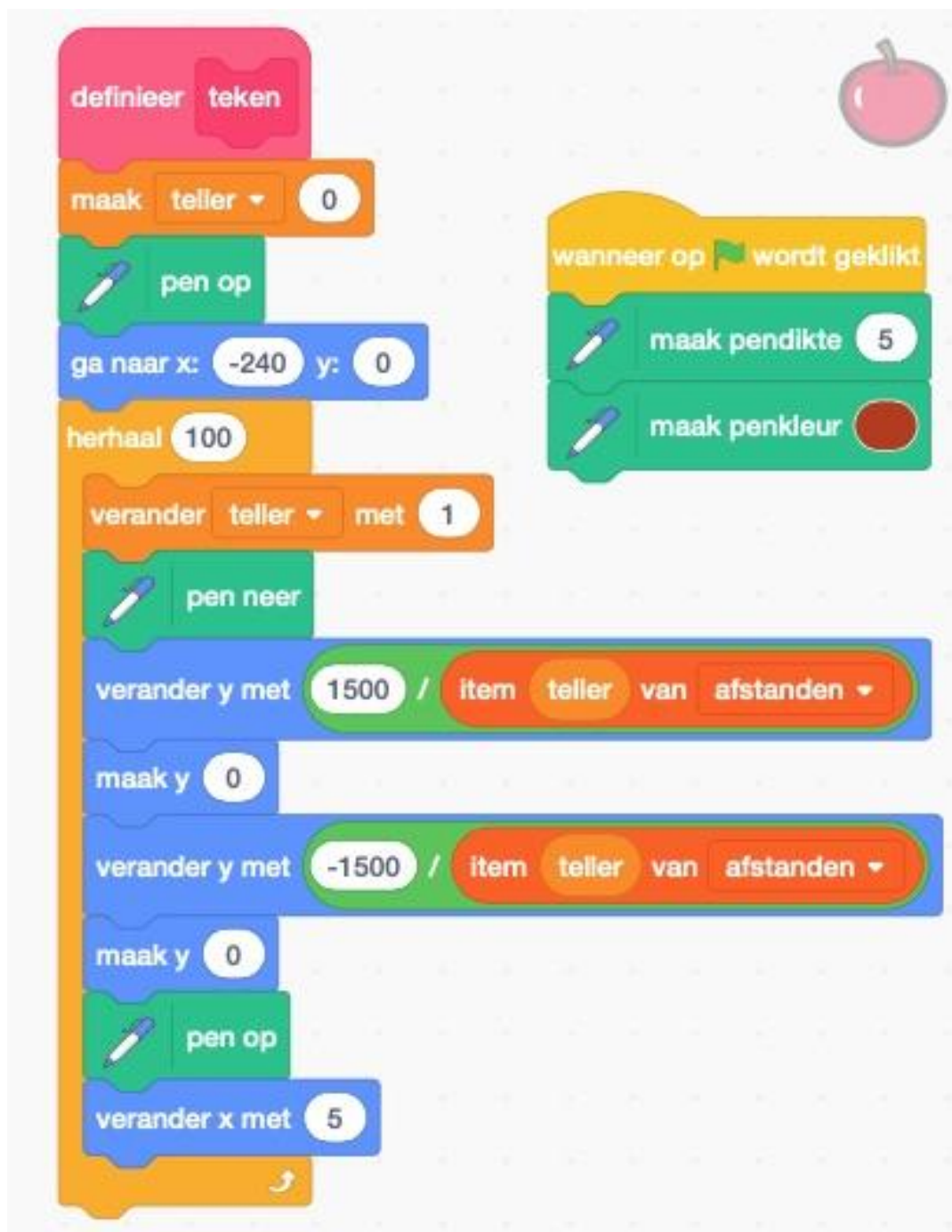


Nu gaan we naar rechts verhuizen om de volgende lijnen te tekenen:

- Hiervoor nemen we het blok 'verander x met 10'. We veranderen het getal 10 door het getal 5. De breedte van deze lijnen moeten vijf zijn. Dit is iets wat we bij de start van het spel moeten duidelijk maken. Dat doe je via de blokjes hiernaast:



- De penkleur mag je op een soort bruine kleur zetten.

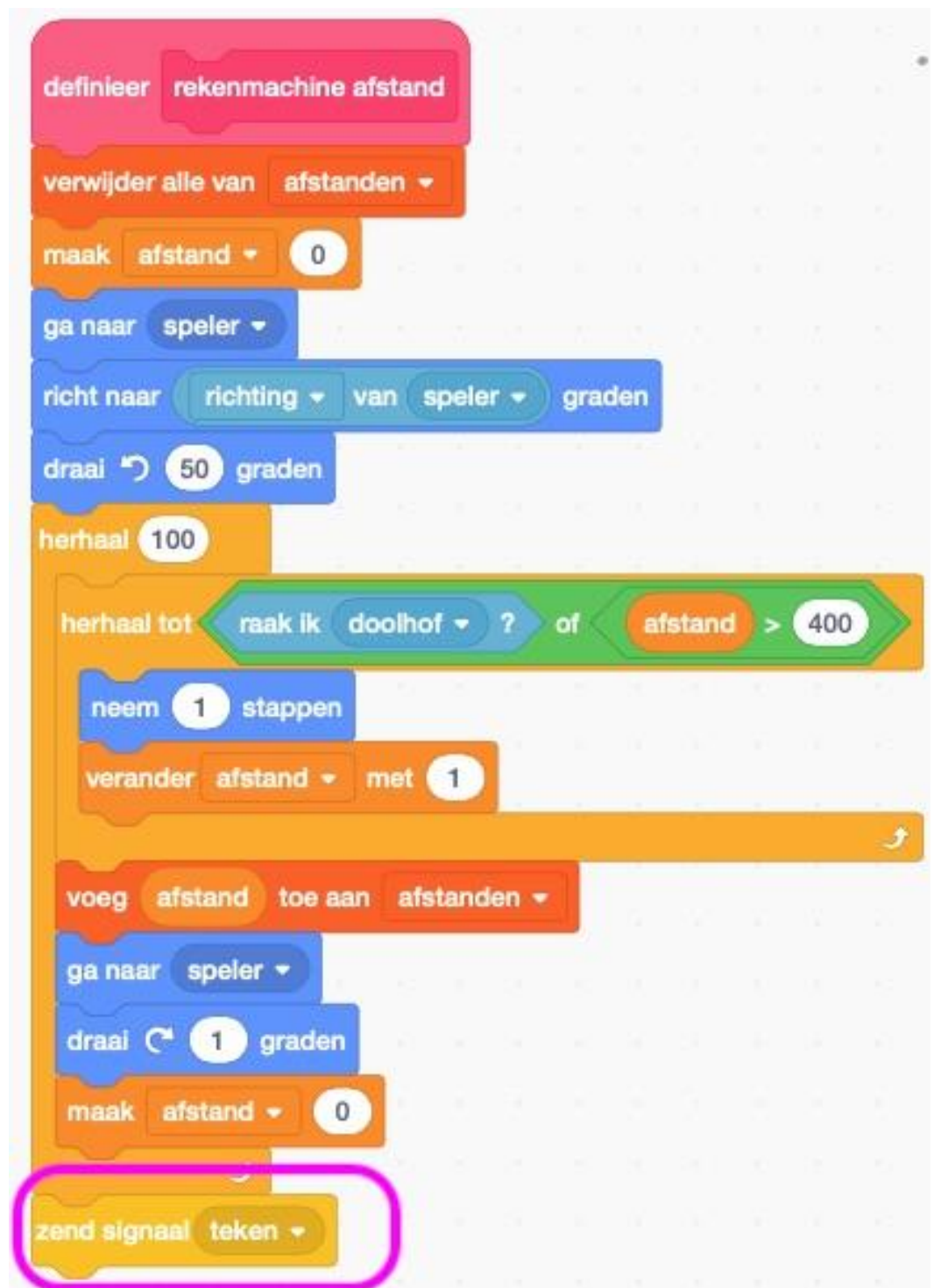


- We gaan ons vorige blok (zie pagina 19) 100 keer herhalen omdat 100 keer 5 de breedte van het hele scherm zal innemen en dat is 480. Nadien mag je de twee grote blokken aan elkaar klikken.

Zo 'teken' is gedefinieerd. Maar wanneer gaan we nu gaan tekenen? Elke keer als we de afstand berekenen gaan we tekenen.

2.5 Selecteer de sprite 'rekenmachine'.

- Plaats helemaal onderaan de reeds aanwezige blokken, het gebeurtenisblok 'zend signaal teken'. Dat laatste signaal moet je eerst nog zelf maken, want dat hebben we nog niet.

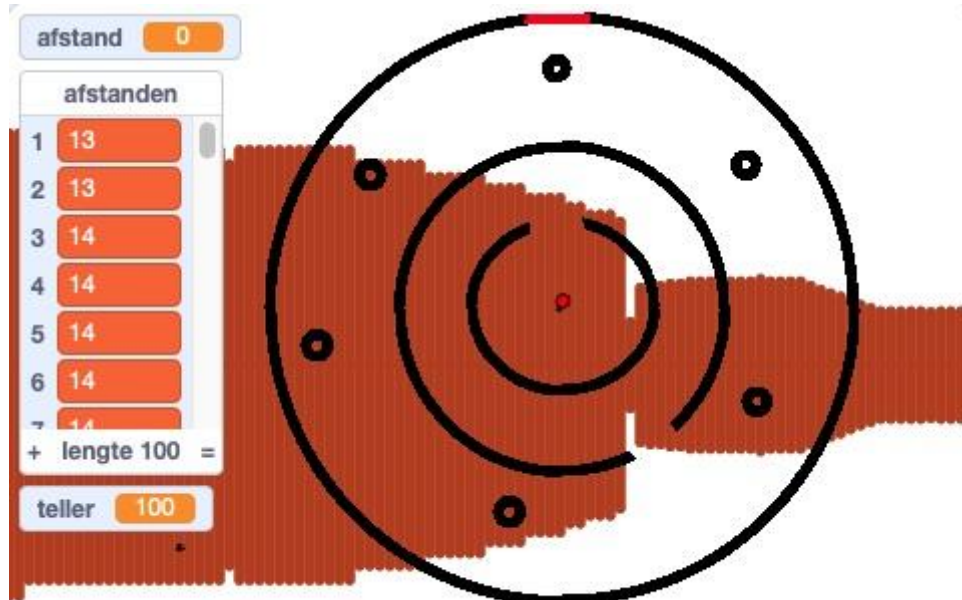


2.6 Selecteer de sprite 'Apple'

- Onze sprite Apple is de tekenaar. Als die het signaal 'teken' ontvangt, moet hij eerst **alles wissen**. Dat komt omdat wanneer we ons bewegen we telkens iets anders zien. Het beeld dat we net voor onze beweging zagen is dus niet langer geldig. Vandaar dat het verwijderd moet worden. Doordat de sprite 'Apple' heel snel **tekent** en oude tekeningen **wist**, zal het scherm nooit wit worden.

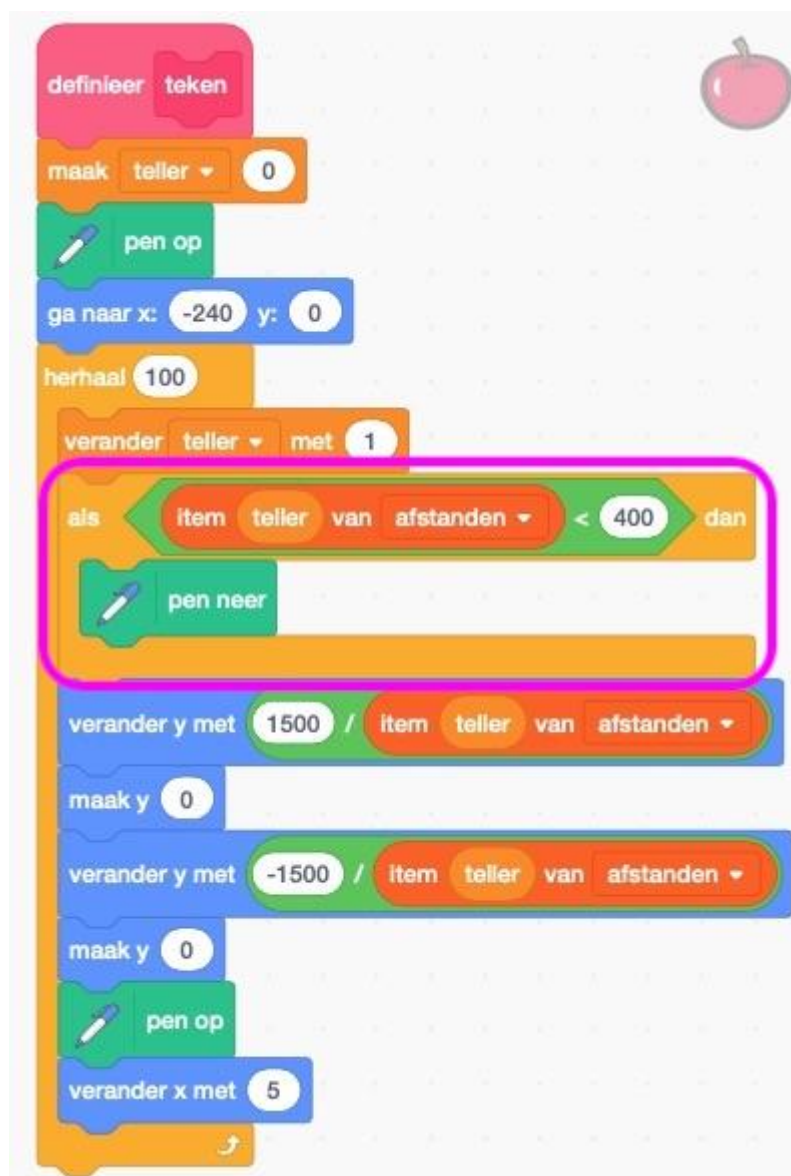
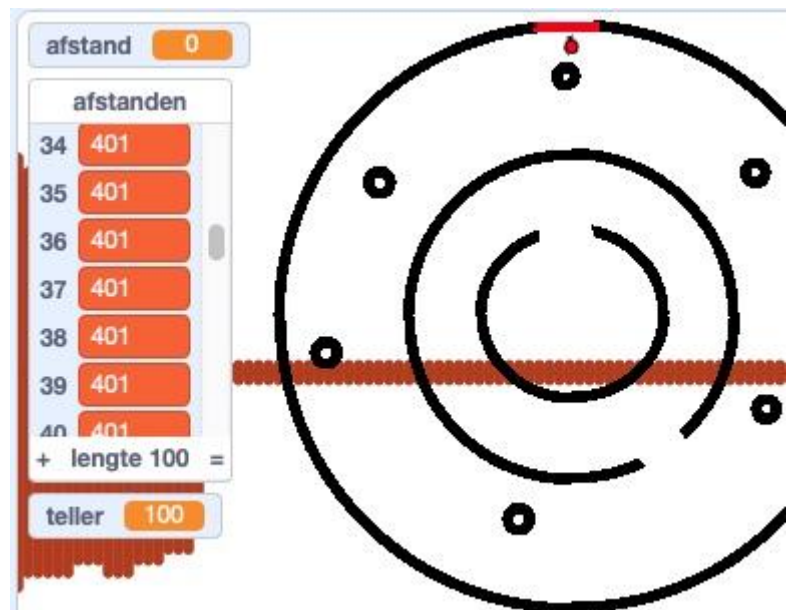


- Klik eens op de code hierboven. Je zou onderstaande beeld moeten zien. Cool he. Het werkt! Beweeg je speler eens en kijk wat er gebeurt. Magisch gewoon!



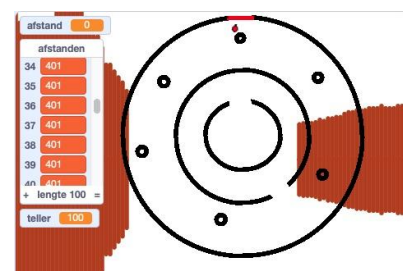
Maar nu moeten we nog enkele probleempjes oplossen. Geen paniek, je kan dit ook wel.

- Als je aan de uitgang van doolhof 'level 1' gaat staan, dan zie je een dunne bruine lijn. Waarom? De cijfers in je lijst geven aan dat er veel plaats is, en toch zien we een lijn. We moeten aan onze sprite 'Apple' zeggen dat hij deze lijn niet mag tekenen wanneer de afstand groter is dan 400. En dat is gelukkig heel simpel.



- We moeten hiervoor **code bij plaatsen, bij de sprite 'Apple'.**

- De **pen moet neer** als het **item teller van afstanden, kleiner is dan 400**. Op deze manier verdwijnt de bruine lijn van ons scherm.



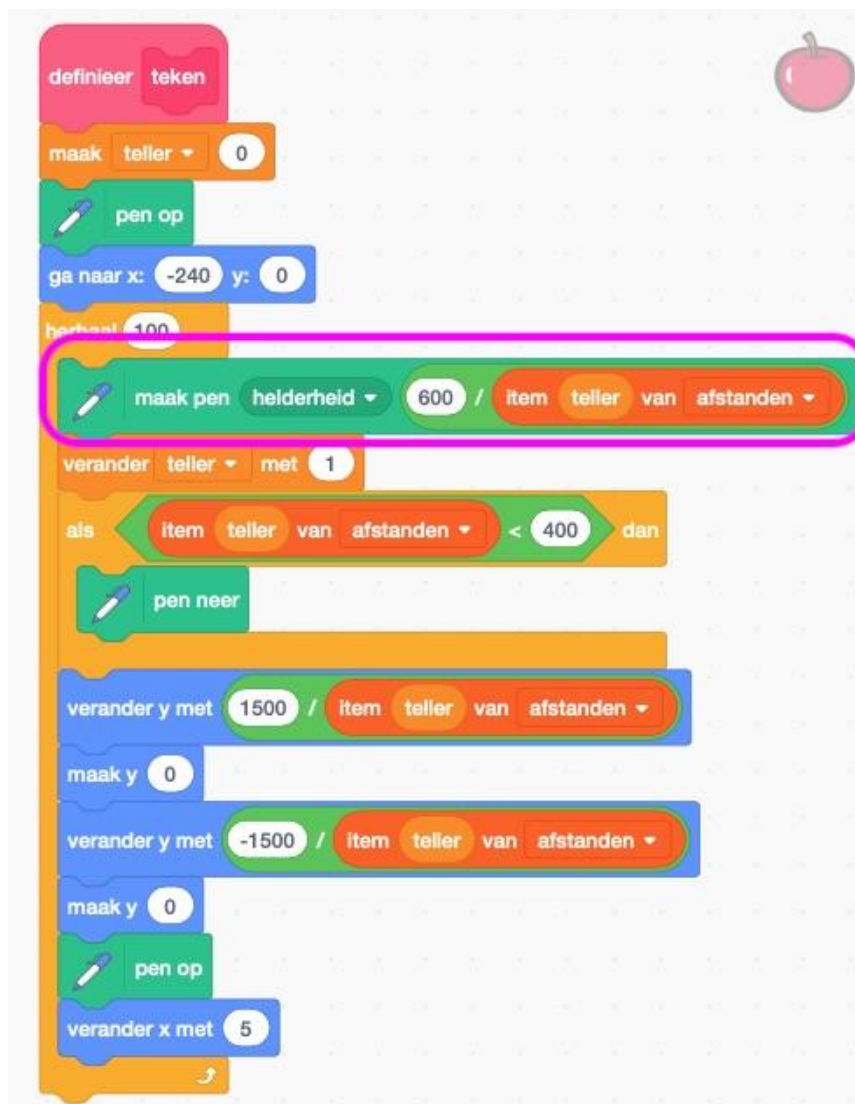
GELUKT!

2.7 Het volgende dat we gaan doen is de helderheid van de bruine muren aanpassen aan de afstand van de sprite 'speler' tegenover die muur.

- Selecteer de sprite 'Apple' en neem bij 'Pen' het blok 'maak penkleur 50'.

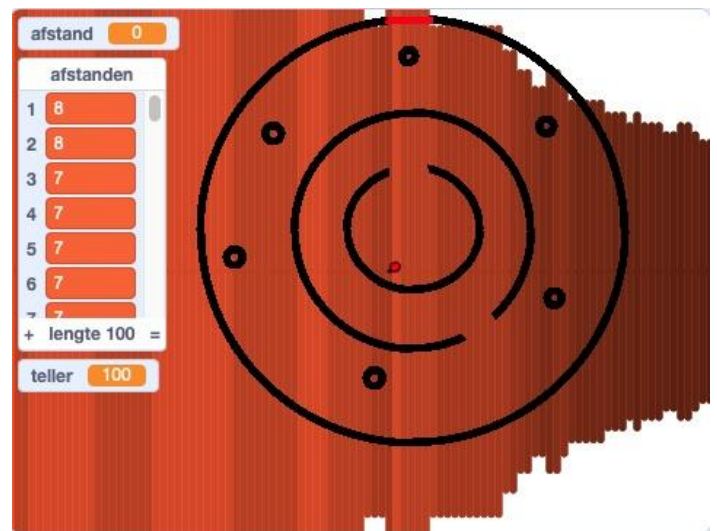


- Verander het woordje 'kleur' in 'helderheid' en in de plaats van het cijfer 50 zet je de blokken '600 / item teller van afstanden'. Dat zijn drie verschillende blokken!



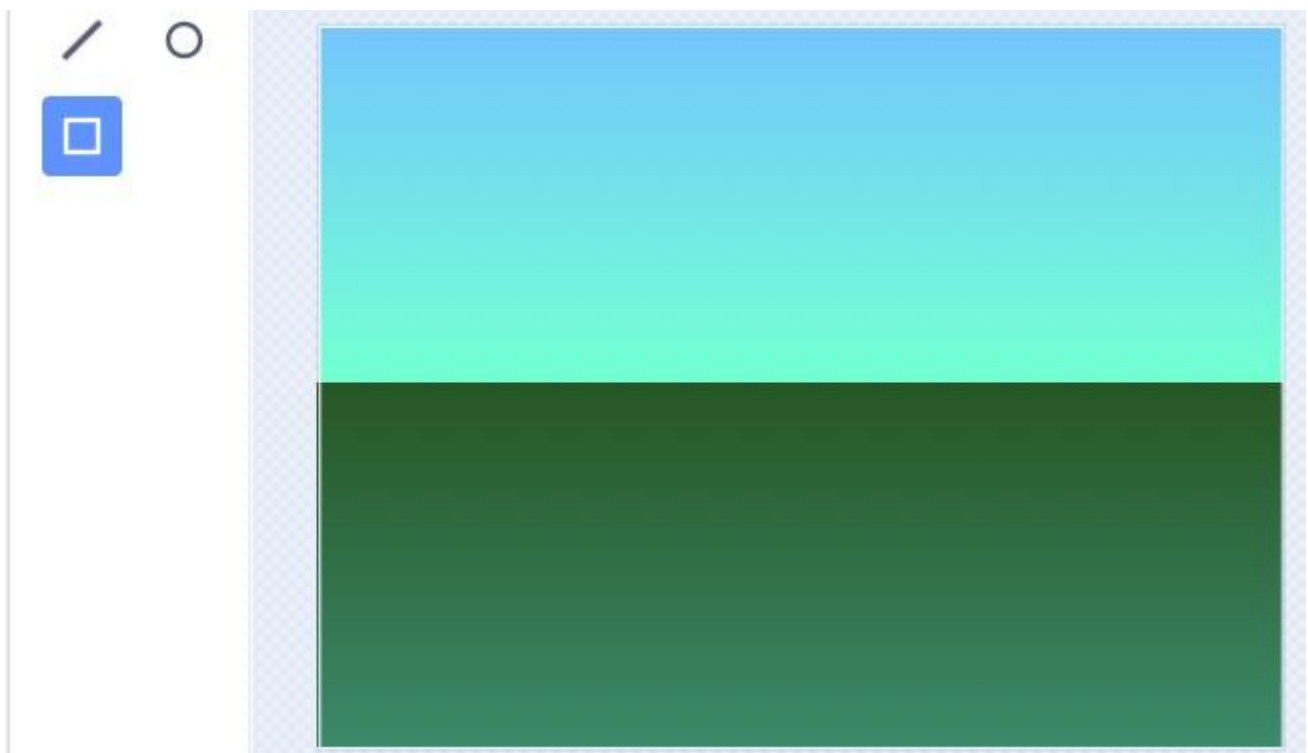
- Het blok hierboven plaats je dan weer in het reeds gemaakte codeblok dat je hiernaast ziet.

- Test je spel en je zal zien dat hoe dichterbij een muur komt, hoe helderder deze wordt. Hoe verder je van een muur staat, hoe donkerder de muur wordt. Dat geeft een meer realistischer effect.

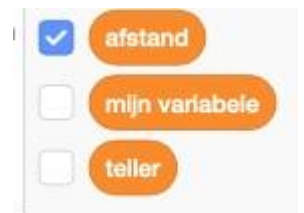


2.8 Om het spel nog realistischer te maken gaan we ook zorgen voor de lucht en de grond. Hiervoor ga je een achtergrond maken.

- **Maak een achtergrond en teken deze zelf.**



2.9 Doe bij al je **variabelen** de vinkjes weg zodat er geen enkel meer in je speelveld staat.



2.10 Je 2D doolhof level 1 en level 2 mag je niet verstoppen. Als je dat wel zou doen, dan zou het spel niet langer werken. Maar we kunnen er wel iets aan doen.

- **Selecteer je sprite 'doolhof'.**
Neem bij Uiterlijken het blok 'zet kleureffect op 0' en pas dat aan naar 'zet transparant effect op 100'.



- Plaats het blok 'zet transparant effect op 100' ook bij je sprite 'speler'.



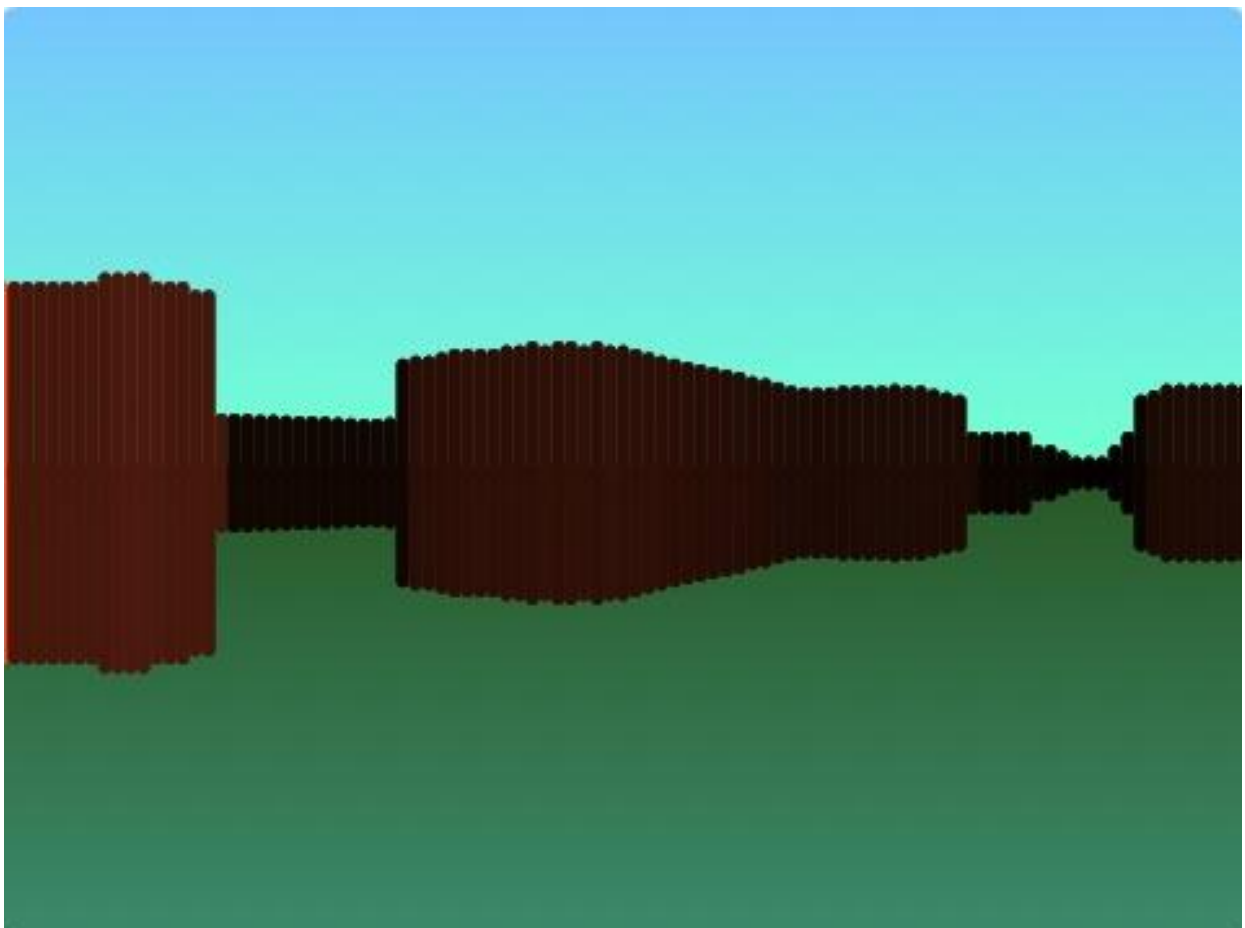
- Plaats het blok 'zet transparant effect op 100' ook bij je sprite 'sensor'.



- Plaats het blok 'zet transparant effect op 100' ook bij je sprite 'rekenmachine'. Plaats eerst het blok 'als op de groene vlag wordt geklikt'. Want dat staat er nog niet.



En nu zou je spel volledig moeten werken! Proficiat als je tot hier bent geraakt, want dit was een hele boterham. Veel plezier!



Link naar dit spel op Scratch: <https://scratch.mit.edu/projects/486828954>