

5.26 Maak je verhaal interactief

We maken een eenvoudig interactief verhaal, waarbij de speler keuzes moet maken en zo beslist hoe het avontuur verloopt. Jij als maker van het spel bepaalt en programmeert alle mogelijke wegen die de speler kan afleggen om zo ofwel het spel te winnen ofwel te verliezen.

1.1 Dee: Klik rechtsonder op het icoon van de poes om een nieuwe sprite te kiezen. Kies 'Dee' als nieuwe sprite.



- Sleep nu uit de linkerkolom de blokken naar het midden zodat je de constructie van hiernaast krijgt.
- Klik nu bovenaan op het groene vlaggetje. Hiermee start je de code die je net gemaakt hebt. Je zal zien dat 'Dee' een tekstballonnetje krijgt, een klein beetje naar rechts beweegt en er daarna een tweede tekstballonnetje komt.
- Wis nu de poes uit je sprites door op het vuilbakje te klikken, nadat je op het icoon van de poes hebt geklikt.



WAT HEB JE NU JUIST GEDAAN?

. De **gele blokken** zijn gebeurtenissen. Ze worden actief als er iets gebeurt. In dit geval: als er op de groene vlag geklikt wordt. Bij deze gebeurtenis worden alle blokken die eronder hangen direct na elkaar uitgevoerd.

. De **blauwe blokken** zijn bewegingen van je sprites (figuurtjes, poppetjes,...)

. De **paarse blokken** veranderen het uiterlijk. De twee blokken die hier gebruikt zijn, tonen een tekstballon en wachten dan 2 seconden. Deze wachttijd zorgt ervoor dat de ballonnen even zichtbaar blijven. Want het programma blijft natuurlijk gewoon lopen en zou anders direct het volgende blok uitvoeren.

. Als je een blok te veel in je programmeerveld hebt staan, kun je dat gewoon terugslipen naar de linkerkant, waar alle blokken staan. Zo wordt dit blok gewist.

1.2 Een leuke achtergrond

We hebben nu een sprite 'Dee', maar ook een hele saaie witte achtergrond.

- Klik helemaal rechtsonder op het icoontje in de kolom speelveld.
- Er opent een scherm met allemaal achtergronden. Kies er eentje, bv. 'Forest' (= bos).
- 'Dee' staat nog steeds helemaal in het midden. Je kan hem een beetje meer naar links en naar beneden zetten door hem vast te pakken en naar een andere positie te slepen. Dat wordt dan zijn beginpositie.



'Dee' staat nu in het bos en je programma ziet er al iets mooier uit!

Het verhaal begint.

Het wordt natuurlijk véél leuker als je vragen kan stellen en het antwoord van de spelers verder in je spel kan gebruiken. Dat zullen we nu doen!

1.3 Verwijder de reeds gemaakte blokken. We hebben deze niet meer nodig. Als je blokken te veel in je programmeer veld hebt staan, kun je deze gewoon terug slepen naar de linkerkant, waar alle blokken staan. Zo worden deze blokken gewist.

- Sleep de juiste blokken naar je programmeerveld om het volgende programma te maken. **Kijk goed naar de kleuren** van alle onderdelen.



- Lees zeker de volgende puntjes als je ergens vastzit.

- Sommige blokken hebben **opties** die je kan aanpassen door op het pijltje naar beneden te klikken



- Je kan **een vraag stellen** aan je spelers door dit blauwe blok te gebruiken. Het antwoord dat de speler geeft, wordt bewaard in een blok met als naam 'antwoord'.



- De **witte velden** in een blok kan je veranderen door erop te klikken en te typen, maar je kan er ook andere blokken in slepen. Je kan dus het blauwe blok 'antwoord' slepen in een wit veld van het groene blok. **Typ een spatie achter 'welkom '!!!** Anders kleven 'welkom' en het antwoord aan elkaar vast.



- Je kan zelfs blokken met andere blokken erin **nog een keer in een ander blok slepen**. Dit lukt als de vormen hetzelfde zijn: een blok in de vorm van een ovaal past in een wit ovaal.



Zo, je hebt de start van een verhaal. Klik zeker op het groene vlaggetje om het uit te proberen en te kijken of er niets fout loopt. Leuk hé?
 Heb je toch iets gevonden dat niet lukt? Lees dan de tips hierboven en kijk je code na!!

2.1 DE EERSTE KEUZE

Nu zullen we het verhaal interactief maken. Dat wil zeggen dat er andere dingen gebeuren in het verhaal als de speler een andere keuze maakt. Eigenlijk moet je het verhaal zien als een soort avonturenpad met verschillende splitsingen, waar de spelers telkens een nieuw weg kiezen.

ALS...DAN...ANDERS

De eerste keuze die in dit verhaal gemaakt moet worden, is wat de speler gaat doen nu hij/zij weet dat de prinses ontvoerd is door de draak.

Er zijn drie opties:

- . **Optie 1:** de speler zegt dat de 'Dee' thuis moet blijven
- . **Optie 2:** de speler zegt dat de 'Dee' zo snel mogelijk moet vertrekken
- . **Optie 3:** de speler zegt dat de 'Dee' eerst zijn boterhammen moet opeten.

'Dee' vraagt dit aan de speler door een vraagblok waarbij de speler met A, B of C kan antwoorden. Maak een nieuw stuk code met als eerste blok dit vraagblok.



Het antwoord dat de speler geeft, wordt bewaard in het blok 'antwoord'.

antwoord

Er zijn nu vier mogelijkheden:

- . **Optie 1:** de speler geeft antwoord A
- . **Optie 2:** de speler geeft antwoord B
- . **Optie 3:** de speler geeft antwoord C
- . **Optie 4:** de speler heeft iets anders geantwoord

Deze vier mogelijkheden moeten geprogrammeerd worden via 'als...dan...anders'blokken. Dit soort programmastructuren heten in het Engels 'if...then...else' en zijn een heel belangrijk deel van computerprogramma's. In Scratch zitten deze structuren bij de **oranje blokken**.



Je zal zien dat deze blokken een diamant-achtig bruin vakje hebben en daaronder twee witte uiteinden. Dit zijn de verschillende stukken code die uitgevoerd worden afhankelijk van wat de uitkomst is van het diamanten blok bovenaan.

Stel dit stuk code samen: we plaatsen drie 'als...dan...anders'blokken in elkaar.



Bij de 'als' regel zie je dat er iets gecontroleerd wordt. Het groene blok checkt met het gelijkheidsteken of de **twee waarden gelijk** zijn. De waarde aan de linkerkant zit in het antwoordblok: dit is het antwoord dat de speler gaf.

De waarde aan de rechterkant is de waarde die het programma gaat controleren. Dus als de speler 'A' getypt heeft, is deze groene check correct en wordt het **bovenste stuk van dit blok** uitgevoerd, namelijk: 'Dee' zegt: 'Je wil dat ik thuis blijf?'

Als het antwoord van de speler niet gelijk is aan A, wordt het onderste stuk van dit blok uitgevoerd en zegt 'Dee': 'Je antwoord was iets anders'.

2.2 GENESTE BLOKKEN

Er waren vier mogelijkheden waarmee de speler kon antwoorden. Mogelijkheid A is al in orde. Nu moet de rest nog geprogrammeerd worden.

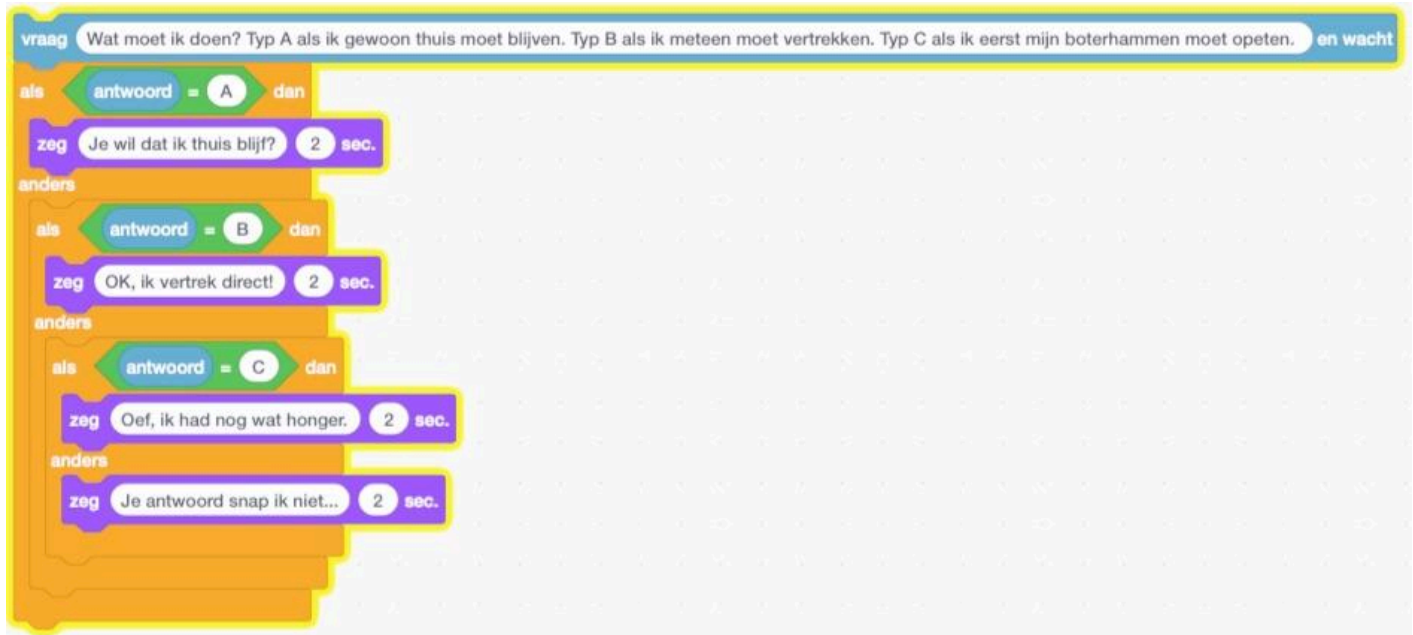
Als het antwoord niet gelijk was aan A, moet het programma opnieuw gaan checken wat het antwoord was. Er komt dus nog een 'als...dan...anders'blok bij.

Dit blok zit in het onderste stuk van het vorige blok. Deze constructie waarbij een keuze in het antwoord van een andere keuze zit, heet **een 'genest'blok**.



TIP: Je kan een losstaand stuk van je code testen door erop te klikken. Er komt dan een gele rand rond en je zal zien dat in het outputvenster dat ene stuk van je code uitgevoerd wordt.

En met nog een extra genest blok erbij, zijn alle vier de mogelijkheden geprogrammeerd!



2.3 ONBEKEND ANTWOORD

De speler kan ook een antwoord geven dat 'Dee' niet als optie had gegeven. In dat geval moet 'Dee' **de vraag opnieuw stellen**. Eigenlijk moet het programma de speler dus **de vraag opnieuw stellen**. Daarvoor moet de uitvoering van de code teruggestuurd worden naar het blok dat de vraag stelt. Dit kan in Scratch door 'signalen' te sturen.

Het ontvangen van een **signaal** is een gebeurtenis, net zoals het klikken op het groene vlaggetje. Het is een **geel blok** dat het begin is van een rij programmeerblokken.

Werk je vraagblok-ketting nu af op de volgende manier:

- bovenaan zet je een gebeurtenisblok
- bij de onderste 'anders' zet je een zend-signaalblok.

Als je op de opties van beide blokken klikt, kan je een nieuw bericht maken.

Maak een nieuw bericht 'vraag1'.

De laatste stap is nu om na de introductie van 'Dee' te zorgen dat het computerprogramma de vraag stelt wat 'Dee' moet doen.

Voeg dus ook daar op het einde een **zend-sigitaal blok** toe met als bericht 'vraag1'

De volledige code is nu:

```
when clicked
  change look to dee-b
  ask "Hoe heet je?" and wait
  say "voeg Welkom en antwoord samen" for 3 sec.
  say "Ik moet je iets vertellen..." for 2 sec.
  say "Mijn prinses is ontvoerd door de draak!" for 3 sec.
  send signal "vraag1"

when signal "vraag1" received
  ask "Wat moet ik doen? Typ A als ik gewoon thuis moet blijven. Typ B als ik meteen moet vertrekken. Typ C als ik eerst mijn boterhammen moet opeten." and wait
  if answer = A
    say "Je wil dat ik thuis blijf?" for 2 sec.
  else
    if answer = B
      say "OK, ik vertrek direct!" for 2 sec.
    else
      if answer = C
        say "Oef, ik had nog wat honger." for 2 sec.
      else
        say "Je antwoord snap ik niet..." for 2 sec.
  send signal "vraag1"
```

3.1 OP AVONTUUR

'Dee' heeft al elke keer geantwoord als de speler iets koos, maar nu moet er natuurlijk nog iets mee gebeuren!

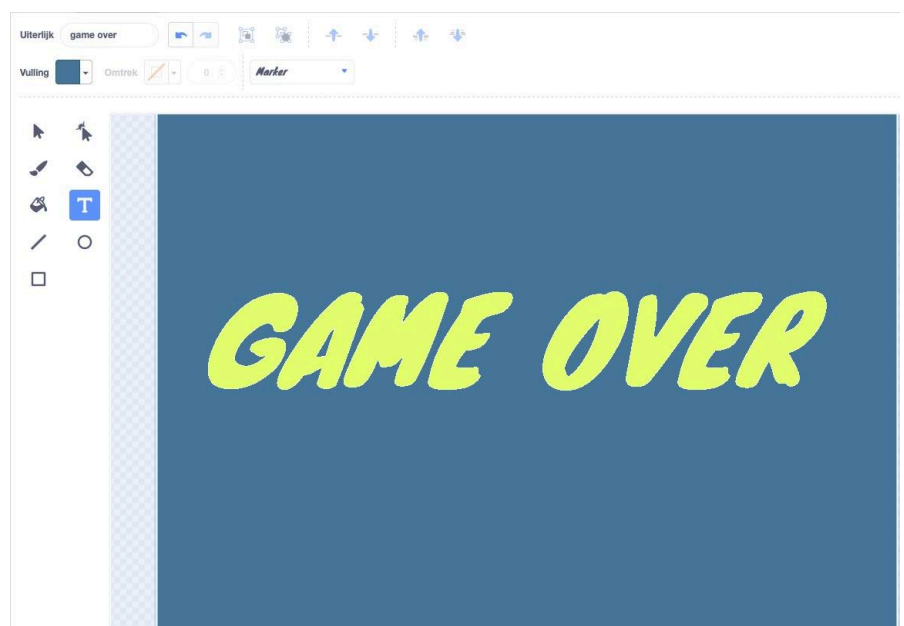
De eerste keuze is de gemakkelijkste. Want als keuze A gemaakt wordt, dan blijft 'Dee' thuis en wordt de prinses niet gered. **De speler verliest: game over...**

3.1.1 EEN 'GAME OVER' SCHERM MAKEN

1. Een 'game over' achtergrond kan je maken door helemaal rechtsonder over het icoontje van de achtergronden te gaan met je muis, nog niet te klikken, en dan op het **penseeltje** te klikken bij de nieuwe opties die daar tevoorschijn komen.
2. In deze editor kan je tekst schrijven, een grote rechthoek tekenen en dit vullen met een kleur. Klik op de grote 'T' en zet de tekst 'game over' in het midden. Je kan deze vergroten door met de muis te slepen aan de hoekpunten. Je kan de tekst ook verplaatsen door de tekst zelf vast te pakken en deze naar een nieuwe locatie te slepen.
3. Klik op het vierkant om een **groot vierkant** te tekenen. Bovenaan kan je bij vulling de kleur veranderen en dan met het emmertje deze vulling aan het vierkant geven.
4. Vergeet ook zeker niet om de naam van dit uiterlijk aan te passen door bovenaan bij uiterlijk 'game over' te typen.



Het kan er dan ongeveer zo uit zien:



HET 'GAME OVER' SCHERM PROGRAMMEREN

Klik links bovenaan op 'code'. Je keert nu terug naar het codescherm en ziet opnieuw een wit programmeerveld. Dit komt omdat elke code die je schrijft **aan een object hangt**.

De stukken code die al geschreven zijn, hangen aan 'Dee'. Om deze code weer zichtbaar te maken, klik je rechtsonder op het icoontje van 'Dee'.

Nu staan alle codeblokken er weer, maar 'Dee' staat wel niet meer in het bos. Hij staat voor een 'game over' scherm. Geen probleem, dat wordt later opgelost. Eerst moet je **het 'game over'scherm programmeren** zodat je het kan tonen en doen verdwijnen.

Dit 'game over'scherm ga je meerdere keren gebruiken, namelijk elke keer als de speler de foute keuzes maakt en het spel verliest. Het is daarom niet handig dat de code hiervoor gekoppeld is aan 'Dee'.

1. Klik helemaal rechts bij speelveld op de 'game over' achtergrond. Alle geschreven codeblokken zullen weer verdwijnen. Rechtsboven in het programmeerveld staat nu in het klein en een beetje transparant je 'game over' achtergrond. Dit geeft aan dat je nu een code aan het schrijven bent die gekoppeld zal worden aan die achtergrond. Zorg voor volgende blokken in je programmeervenster:



2. Activeer terug de programmeerblokken van 'Dee' door rechtsonder op het icoontje van 'Dee' te klikken. Voeg volgende blokken toe in de 'als...dan...anders' constructie.

Je zal een nieuw blok zien. Meer over dit 'herhaal' blok leer je zometeen. Eerst nog wat andere dingen in orde brengen!



TIP: Je zal gemerkt hebben dat je op verschillende plaatsen stukken code kan schrijven die met elkaar communiceren. Als je veel programmeert, zal je ontdekken dat er bepaalde patronen zijn die softwareontwikkelaars hierin volgen om hun code een

beetje overzichtelijk te houden. Dit noemen ze **design patterns**. In Scratch wordt alle code aan objecten (de sprites, achtergronden, ...) gekoppeld. Dit wordt 'objectgeoriënteerd programmeren' genoemd.


RESET ACHTERGROND EN ZET 'DEE' TERUG OP ZIJN PLAATS

Momenteel staat de 'game over' achtergrond nog steeds actief, wat niet de bedoeling is. Na een eerste test heb je misschien ook gemerkt dat 'Dee' uit beeld gaat lopen als je een paar keer na elkaar test.

Het spel moet zo geprogrammeerd worden dat elke keer wanneer er op start gedrukt wordt, de achtergrond een bos wordt en 'Dee' terug linksonder staat.

1. De achtergrond veranderen naar het bos doe je door bij de achtergrond twee extra blokken toe te voegen:
 - Een gebeurtenis: 'wanneer op het groene vlaggetje wordt geklikt'
 - Verander achtergrond naar 'Forest'



2. Om 'Dee' terug linksonder te zetten, selecteer je hem eerst rechtsonder door op het icoontje van 'Dee' te klikken. Om de **exacte positie** van 'Dee' te vinden kan je 'Dee' in het scherm rechtsboven terugslepen naar zijn beginpositie. Als je dan in de linkerbalk het blok 'ga naar x: y:' zoekt, zal je zien dat daar de cijfertjes aangepast zijn. Sleep dit blok op zijn plaats in de rij:
 



3. Test nu het stuk code dat begint met 'wanneer ik signaal vraag1 ontvang' door erop te klikken. **Geef antwoord A om te testen.**

'Dee' antwoordt, de achtergrond wordt 'game over' en 'Dee' wandelt weg.

3.2 BOTERHAMMEN GEGETEN?

Optie A is nu klaar. Optie B en C lijken dan weer heel goed op elkaar: 'Dee' gaat de prinses redden. Alleen zit het verschil in de maag van 'Dee': heeft hij gegeten of niet?

Het bijhouden van een bepaalde waarde kan bij programmeren in een '**variabele**'. Een variabele kan je vergelijken met een doos met een etiket erop. Op dat etiket staat de naam van de variabele. Telkens wanneer je kijkt wat er in de doos zit, krijg je de waarde van de variabele terug. Je hebt dit stiekem al geprogrammeerd toen je 'antwoord' gebruikt hebt in de tekstballonnen van 'Dee'.

Je kan **nieuwe waarden** in dezelfde variabele opslaan. Op dat moment verdwijnt de vorige waarde automatisch. Ook elke keer dat je het programma opnieuw opstart, worden alle variabelen gewist. (Als je in een random spel 15 als score haalde, wordt die score terug op nul gezet bij het begin van je spel).



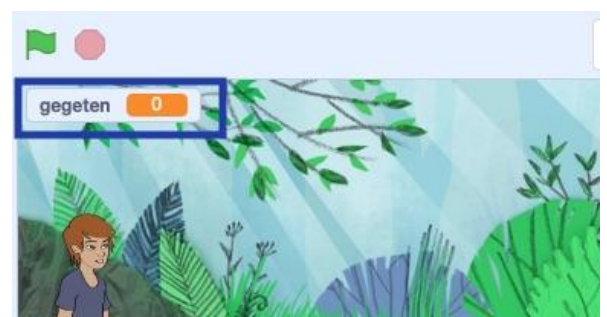
1. Maak een variabele 'gegeten'. Klik hiervoor links bij 'Variabelen' helemaal bovenaan in de kolom op de knop 'Maak een variabele'. Zorg dat alle sprites deze variabele kunnen gebruiken.

2. Gebruik deze variabele dan twee keer:

- één keer bij antwoord B. Daar zet je die op '0'
- één keer bij antwoord C. Daar zet je die op '1'



Nu houdt je programma bij of 'Dee' gegeten heeft of niet. Later kan je dan een **beslissing** nemen op basis van deze variabele.



TIP: Je ziet deze variabele nu ook linksboven in je outputscherm. Voorlopig is het handig dat die nog zichtbaar is.

Dit soort extra zichtbare informatie noemt men debug-informatie. Debuggen is Engels en betekent: fouten in je code kunnen opsporen en verbeteren. Je kan deze informatie verbergen door het blauwe vinkje naast de naam van de variabele uit te zetten.



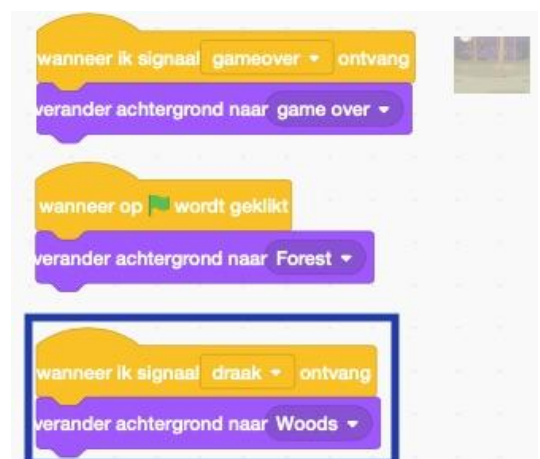
3.3 DE DRAAK

Tijd voor de volgende fase in het avontuur van 'Dee': de ontmoeting met de draak...

1. Kies helemaal rechtsonder via het icoontje een nieuwe achtergrond: 'Woods' (bij 'buiten'). Klik daarna op 'speelveld'. Je ziet het stukje code van 'game over' nog staan in je programmeerveld.



2. Voeg nu een nieuwe gebeurtenis toe met een nieuw signaal: wanneer ik signaal 'draak' ontvang en verander de achtergrond.



3. Voeg een nieuwe sprite toe, net zoals je 'Dee' helemaal in het begin hebt toegevoegd. Dat kan via het icoontje van de poes rechtsonder. Kies 'dragon'.



Er zijn zo nog een paar dingen die moeten gebeuren. De draak kijkt naar de verkeerde kant en de draak moet nog even verdwijnen en pas verschijnen als 'Dee' in het bos is. En daarna zal 'Dee' moeten kiezen wat hij doet: vechten of vluchten.

3.3.1 DE DRAAK DRAAIEN

De draak naar de andere richting laten kijken is heel eenvoudig, daar is een blauw blok voor: 'richt naar...graden'. Zorg dat de sprite van de draak actief is. Klik daarvoor op het icoontje van de draak rechtsonder zodat er een blauw kader rond staat. Sleep het blok in dit leeg programmeerveld en vul '-90' in om de draak te draaien.



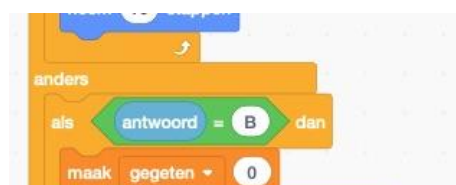
Staat de draak op zijn kop? Klik dan op de twee pijltjes die naar elkaar wijzen bij 'richting' onder je speelveld.

3.3.2 DE DRAAK VERBERGEN EN LATEN VERSCHIJNEN

Je zal merken dat bepaalde patronen altijd terugkomen. Om de draak te laten verschijnen gebruik je weer de gebeurtenis 'wanneer ik signaal draak ontvang' en voeg je het paarse blok 'verschijn' toe

Maar de draak moet wel **bij de start** van het spel (als op de groene vlag geklikt wordt) verborgen worden. Dat doe je door een extra gebeurtenis met het groene vlaggetje toe te voegen en daaraan het blokje 'verdwijnen' te hangen.

Je kan de draak nog een beetje op zijn plaats zetten in het venster rechtsboven door hem een beetje naar rechtsonder te slepen.



3.3.3 DE DRAAK AANROEPEN

De laatste stap is om het signaal 'draak' aan te roepen nadat de speler een antwoord heeft gegeven aan 'Dee'. Selecteer hiervoor terug de sprite 'Dee' en voeg bij antwoord B én antwoord C volgend blok toe: 'zend signaal draak'.

3.3.4 'DEE' EN DE DRAAK

vraag Wat moet ik doen? Typ A als ik naar huis moet vluchten, typ B als ik moet vechten tegen de draak. en wacht

Nu heeft 'Dee' de keuze: **vechten of vluchten**. Dit betekent dat er nu ook bij 'Dee' extra codeblokken moeten komen die uitgevoerd worden als het signaal 'draak' ontvangen wordt. Zet daarom in het programmeerveld bij 'Dee' de volgende blokken bij. Deze blokken zouden allemaal bekend voor je moeten zijn:

TIP: Doordat je daarnet het 'game over' scherm apart hebt gemaakt, kan je dit nu heel eenvoudig opnieuw gebruiken. Je hoeft er namelijk niets aan te veranderen, maar gewoon het signaal 'game over' te verzenden! Softwareontwikkelaars hergebruiken heel graag andere stukken code. Als dit lukt, is dat meestal een teken dat de code goed geschreven is.



WINT 'DEE' HET GEVECHT?

In dit spel is het heel simpel: 'Dee' wint het gevecht als hij **genoeg boterhammen** gegeten heeft. In een vorig deel heb je dit bijgehouden in een variabele: gegeten. Nu kan je dus gaan testen door deze variabele op te roepen en te gebruiken in een 'als...dan...anders' blok.

Maak deze constructie van blokken: ↴



Voeg er een signaalblok aan toe met nieuw bericht 'gevecht'.

En plaats deze in het lege vakje



Als 'Dee' dus in een vorige stap gegeten heeft, wordt signaal 'gevecht' verstuurd. Als hij niet gegeten heeft, wordt signaal 'game over' verstuurd.

TIP: De waarde van de variabele 'gegeten' is 0 of 1. Computers denken graag in eentjes en nullen. Een 1 staat meestal voor 'ja' of 'juist' terwijl een 0 'nee' of 'fout' betekent. Deze waarde heb je dus in computertaal bijgehouden in de variabele 'gegeten'. De '1' betekende dat 'Dee' gegeten had, de '0' dat 'Dee' niet gegeten had.

HET GEVECHT

Vooraleer 'Dee' de prinses kan redden, moet de draak nog **verslagen** worden. Dit kan je eenvoudig programmeren door het blok 'verdwijnt' te koppelen aan de gebeurtenis 'gevecht' of je kan het iets cooler maken door de draak bijvoorbeeld kleiner te maken en rond te laten draaien tot hij weg is.

'DEE' IS KWAAD

'Dee' is nu kwaad. Voeg in het programmeerveld van 'Dee' deze blokken toe om het uiterlijk van 'Dee' te veranderen als het gevecht begint:



De draak gaat even wachten en daarna kleiner worden, ronddraaien en uiteindelijk verdwijnen. Hiervoor gebruik je nog een aantal nieuwe blokken.

EVEN WACHTEN

Klik op het icoontje van de draak en voeg dit signaalblok toe:



TIP: Een computerprogramma voert alle instructies direct na elkaar uit. Als je wil dat het programma even wacht, bijvoorbeeld om de **spanning wat op te bouwen**, moet je dat ook extra programmeren. Dat kan je in Scratch doen door het blok 'wacht...sec.' te gebruiken. Het computerprogramma pauzeert dan even.

DE DRAAK WORDT KLEINER EN VERDWIJNT

Nu zal je de draak kleiner laten worden en laten ronddraaien in 20 stappen. Hiervoor gebruik je een **herhaalblok**. Je hebt dit al eens gedaan bij het 'game over' scherm om 'Dee' helemaal uit beeld te laten lopen. Nu gebruik je een iets ander blok 'herhaal...'

Herhaalblokken voeren de code die er tussen staat een aantal keer uit. Het witte veld bovenaan dit blok bepaalt hoeveel keer iets uitgevoerd wordt.

1. Zet in dit wit veld '20'. De code in dit blok zal dus 20 keer direct na elkaar uitgevoerd worden.
2. Sleep in dit blok 'verander grootte met -5' om de draak **kleiner te maken**.

Aangezien de standaardgrootte van de draak gelijk is aan 100, zal de draak 20 keer verkleinen met 5 (het minteken wil zeggen: verklein). Zo wordt de draak dus 0 groot.

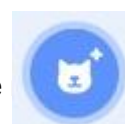
Het draaiblok maakt het allemaal nog wat spectaculairder.



3. Uiteindelijk moet de draak verdwijnen en het volgende stuk aangeroepen worden: zend signaal 'prinses'



DE PRINSES



Klik rechtsonder in het spritescherm op het icoontje van de poes om een nieuwe sprite toe te voegen. Kies bij 'fantasie' de 'Princess'. Deze kijkt ook de verkeerde kant op, maar dat ga je direct oplossen. We gaan hier een andere manier voor gebruiken dan bij de draak.

UITERLIJK AANPASSEN

Selecteer de prinses (= Princess) bij de icoontjes rechtsonder. Kies hierna linksboven 'uiterlijken' en kies daar Princess-a, dit is de prinses met de glimlach.

Klik met je rechtermuisknop op het icoontje van Princess-a en kies 'dupliceren'.

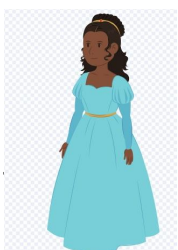
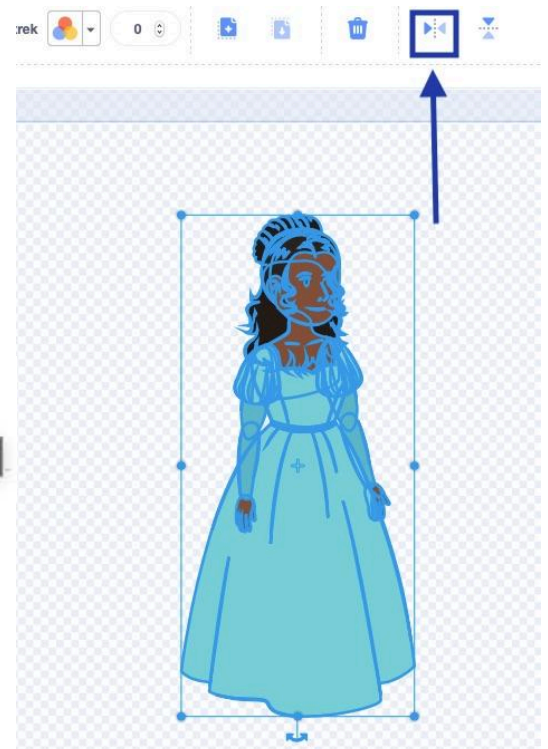
Je krijgt nu een **extra variant** van deze sprite. Hernoem die naar 'prinses-gered'.



Selecteer in het tekenschermbord (het veld met ruitjes als achtergrond) de volledige prinses-gered door te slepen met je muis van linksboven tot rechtsonder.

Je hebt nu ongeveer rechts in het midden boven de geselecteerde tekening een blauw driehoekje. Als je daar met je muis op staat, komt er de tekst 'links-rechts omdraaien'.

Klik hier nu op en de prinses is omgedraaid. Je hebt nu een extra variant van deze sprite gemaakt die je straks kan gebruiken in je code.



Heb je geklikt op 'links-rechts omdraaien'? Dat kijkt je prinses nu de andere kant uit.

'DEE' ONTMOET DE PRINSES

Alleen deze dingen moeten nu nog gebeuren: de prinses moet verschijnen, 'Dee' en de prinses moeten naar elkaar toelopen en een scherm moet verschijnen dat aangeeft dat de speler **gewonnen** heeft.

1. Klik op het icoon van de prinses en zet deze blokken in het programmeerveld:



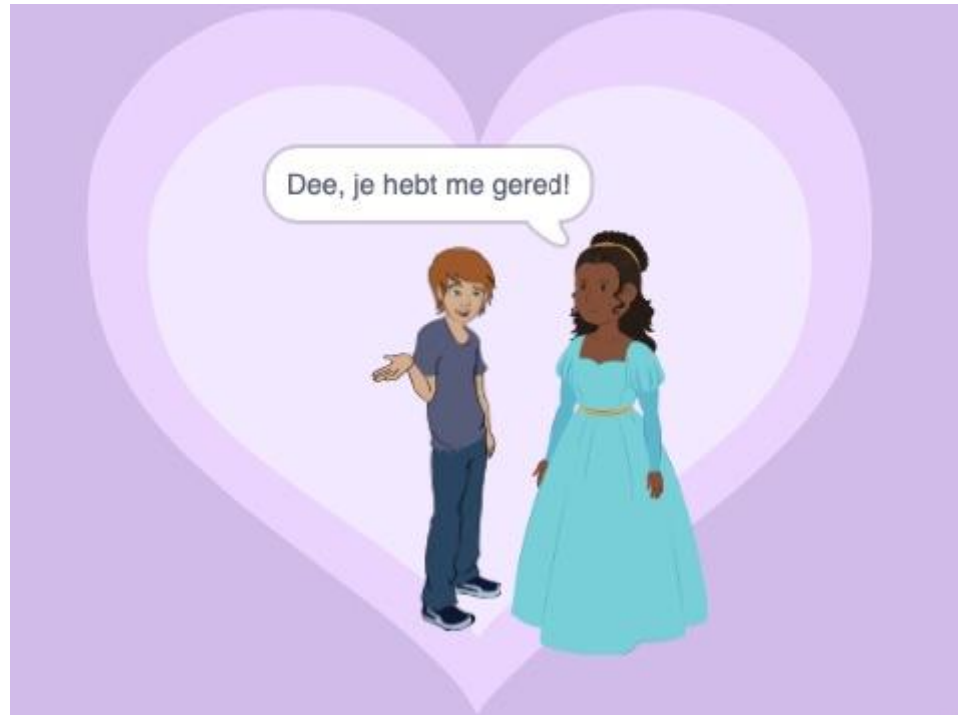
2. Bij 'Dee' zet je deze blokken:



3. Als laatste verander je de achtergrond. Voeg een nieuwe toe: 'hearts' en voeg bij speelveld volgende code toe:

KLAAR!

Als alles goed gaat krijg je nu volgend eindscherm zodat de speler weet dat hij of zij gewonnen is.



Je bent nu klaar om heel je spel te testen en te spelen. Probeer alle mogelijke antwoorden en combinaties van antwoorden eens uit en laat dan zeker eens iemand anders spelen.

Je mag natuurlijk altijd nog extra code toevoegen, zodat deze verhaallijn nog uitgebreider wordt. Misschien kan je 'Dee' echt laten vechten met de draak? Of je laat de prinses haar speciale krachten gebruiken, zodat zij de held van dit verhaal wordt. Je mogelijkheden zijn onuitputtelijk! Geef je creativiteit alle kansen!