

Scratch (17)

René Suiker

Zachtjes aan krijg ik het gevoel, dat deze serie ten einde loopt. Ik heb de tijd niet om veel nieuwe zaken nu op te pakken, zoals aansluiting van hardware, zeker niet nu ik ook binnen CompUsers een nieuwe rol heb aanvaard (zie het artikel over de *CompUfair*). Ook heb ik op het moment geen concrete plannen voor een nieuw Scratch-project. Op het moment van schrijven ben ik ook bezig met de voorbereiding voor een lezing over Scratch op de HCC!kennisdag op 8 oktober. Als ik dit schrijf is het 2 oktober, dus ik ben nog op tijd. Maar als jullie dit lezen is dit evenement al achter de rug. Maar voor de mensen die de lezing hebben bijgewoond als kennismaking met Scratch ga ik toch nog even terugblikken op die lezing.

Overigens vindt u elders in dit nummer een terugblik op deze kennisdag, inclusief een terugblik op mijn lezingen. Meer-voud? Jawel, want ik mocht ook nog iets vertellen over een ander thema, dus verwijs ik graag naar het artikel zelf. Maar in de lezing gaf ik een aantal hints voor Scratch-projecten om zelf te bekijken, en als service voor de bezoeker neem ik deze links in dit artikel op.

Leren

Het is belangrijk te blijven leren. Daarbij kun je soms wat hulp gebruiken. De vorige keer beschreef ik hoe ik Scratch heb ingezet bij de bijles wiskunde die ik heb gegeven. De volgende bijles die ik ga geven betreft bijles Engels. Misschien dat ik daar ook Scratch bij kan gebruiken, maar ik heb nog geen idee in welke vorm ik dat dan moet gieten. Misschien dat één van mijn lezers/lezeressen een idee heeft. We staan altijd open voor goede suggesties, trouwens ook voor suggesties over nieuwe onderwerpen om aan te snijden.

Lezing

Ik hield dus een lezing en deze was echt gericht op beginners, als kennismaking met Scratch. De vaste lezers zijn in tussen uiteraard een stuk verdergekomen, want we zijn in deze reeks al bijna op het niveau van een commercieel spel. We kunnen in Scratch één van de eerste computerspelen al heel makkelijk namaken, maar we zijn al bezig een stukje verder te gaan.

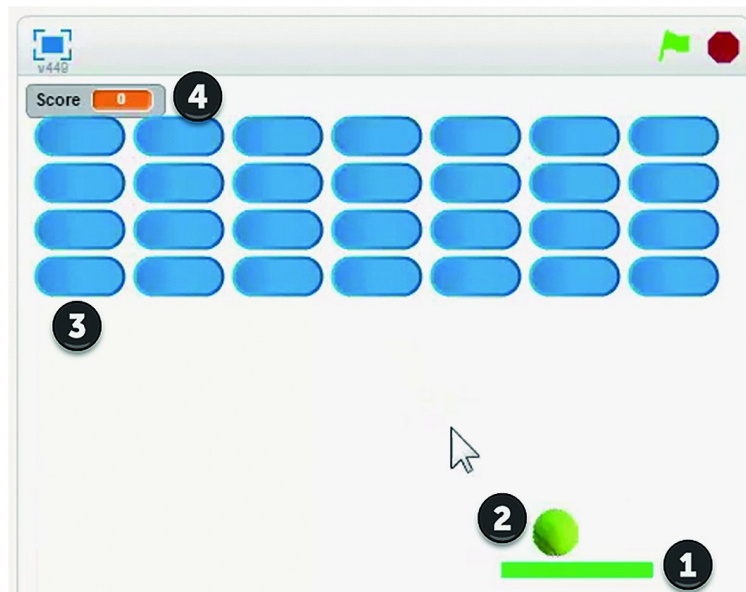
In de lezing ging ik in op Scratch, op programmeren in het algemeen en over de voordelen van Scratch. Voor de mensen die nu pas inschakelen en dit allemaal nog eens terug willen zien, op de website van CompUsers, onder Platform WebOntwerp, kunnen jullie een opname vinden van een Webinar dat ik vorig jaar een keer heb gegeven, toen we als gevolg van Corona-maatregelen geen fysieke bijeenkomsten konden organiseren. Gelukkig is die tijd voorbij, maar we realiseren ons ook dat ons online aanbod best wat uitbreiding kan hebben. Vandaar dat we de opname van het Webinar beschikbaar hebben gesteld.

Bricks

We waren dus bezig met een Scratch-versie van het oude computerspel Bricks. De vorige keer hebben we een aantal zaken geregeld om het elementair speelbaar te maken. Nog lang niet alles zoals we het straks willen hebben, zeker niet genoeg om er een commercieel spel van te maken, maar het

begin was er. Ik had al aangegeven dat ik het zeker niet allemaal zelf heb verzonnen, maar mezelf gebaseerd heb op een cursus die ik volg, een cursus van Al Sweigart. Zoals ik vorige keer al aangaf, voor hen die het Engels goed beheersen, kijk gerust eens op zijn website: <https://inventwithscratch.com/> aangezien hij daar in hoofdstuk 5 dit spel volledig beschrijft. Gratis online te lezen.

We hebben een speelveld ingericht, met blokjes die bij de start van het spel op hun plek gezet worden. We hebben een scoreteller, een batje en een balletje. Allemaal in eerste instantie recht-toe-recht-aan. Al met al zag het er als volgt uit:



Figuur 1 - Bricks game tot dusver

Bij (1) zie je dus je batje, dat je met behulp van de muis kunt laten bewegen. Bij (2) zie je het balletje, dat op je batje kan stuiteren, zodat het weer omhoog beweegt. Bij (3) zie je de 'bricks' die je met je balletje kapot moet schieten en bij (4) dus de scoreteller.

Vervolgens hebben we de besturing van het batje geschreven, waarbij we een stukje vertraging hebben gebouwd, om wat uitdaging aan het spel toe te voegen.

Daarna beschreven we de beweging van het balletje. Ik zeg nadrukkelijk beweging en niet besturing, want het balletje beweegt autonoom, maar wel beïnvloed door de omgeving en uiteraard het batje. Hierbij wordt in principe hoek van inval is hoek van uitval gehanteerd, maar met een beetje willekeurige variatie, ook weer om wat onvoorspelbaarheid toe te voegen, want we willen het spel niet te makkelijk maken. Verder hebben we een aantal variabelen in het programma geïntroduceerd, zodat we gedurende zowel de testfase als gedurende het verloop van het spel de moeilijkheidsgraad kunnen bijsturen. Als je alles met constanten in je code regelt dan is het onderhouden en het testen van je programma een stuk complexer.

Verder hadden we nog een trucje van Al gebruikt voor het stuiteren op het batje, als een soort anti-dender-schakelaar. Op die manier stuitert het balletje gewoon omhoog als het

batje geraakt wordt en blijft het er niet aan 'kleven'. De blokjes hadden we op een speciale manier geplaatst door de sprite naar de positie te brengen, zichzelf te klonen, naar de volgende positie te gaan, zichzelf weer te klonen, etc. Aan het eind van dit opbouwtraject verdwijnt de originele sprite. Het voordeel van klonen is dat je maar voor één sprite de code hoeft te schrijven en dat die dan op alle klonen van toepassing is.

We behandelden ook het verschil tussen de 'gebeurtenis' van 'op de groene vlag drukken' en de 'besturing' 'wanneer ik als kloon wordt gestart'. Die laatste code wordt uitgevoerd als de kloon wordt gestart. Als je wilt dat alle klonen tegelijkertijd worden geactiveerd, dan kun je dat weer met een gebeurtenis 'signaal' doen. Maar dat hadden we nu niet nodig. Later misschien nog wel, als we de blokjes willen laten bewegen.

Verder legde ik nog uit waarom we de code voor het raken bij de stenen legden en niet bij het balletje.

Als de bal de steen raakt, dan tellen we 1 bij de score op. Als alle stenen weg zijn hebben we gewonnen. Daar hebben we ook een sprite voor gemaakt, die dan getoond wordt. We hebben trouwens ook al een sprite gemaakt voor als wordt verloren, maar in de versie van vorige keer was verliezen nog niet geïmplementeerd.

Het project is intussen gedeeld op de site van Scratch: <https://scratch.mit.edu/projects/697012542/>

Dus als je het huiswerk wilt uitproberen, of andere dingen met deze code wilt doen, leef je uit.

Huiswerk

Na deze terugblik is het traditie om even naar het huiswerk te kijken.

Om het spel als verliezer te beëindigen moeten we iets anders doen. We hebben wel een vergelijkbaar stukje code nodig, maar met een andere sprite, dus een andere tekst. En dus met een ander signaal, namelijk 'Verloren'. Alleen, dat hebben we nog niet verzonden. En daar gaat het huiswerk van deze keer dus over. Nog even ter herinnering:

Opgave 16.1:

- Onder welke sprite moet je een code-aanpassing doen?
- Hoe ziet deze code-aanpassing eruit?
- Als je dat gedaan hebt, werkt het spel dan correct?

Opgave 16.2:

- Is het spel uit te spelen?
- Als het te moeilijk is, hoe kun je het makkelijker maken?
- Als het te makkelijk is, hoe kun je het moeilijker maken?

Opgave 16.3:

- Het spel werkt, maar wat mis je nog?
- Het spel stopt toch niet nadat de stenen weg zijn, wat ging er mis en hoe kun je dat herstellen?

Uitwerking Opgave 16.1:

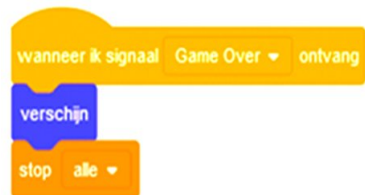
- Het ligt voor de hand om hier de bal voor te gebruiken. Zodra de y-positie van de bal onder de y-positie van het batje komt is die er dus langs gekomen en dan is het einde van het spel. In een latere fase kunnen we natuurlijk ook een aantal herkansingen opgeven, zodat je niet gelijk verliest als er een bal gemist wordt, maar bijvoorbeeld na drie ballen. Ook dit kan in een variabele, dan kun je net als in een flipperkast tijdens het spel nog bonusballen vergaren. Dit soort ideeën kunnen ontstaan terwijl je het spel aan het maken bent.
- In de 'eeuwige lus' van de bal voegen we een blokje in als volgt:



Figuur 2 - Game Over

- Nee, dan moet je ook nog iets doen in de code van de sprite 'Game Over'. Tot dusver stond hier alleen dat hij moest verdwijnen zodra op de groene vlag werd geklikt. Nu moet hij ook reageren zodra het signaal wordt ontvangen.

Als het echt 'Game Over' is kun je volstaan met de volgende code:



Figuur 3 - Sprite Game Over

Als je straks eventueel met extra ballen wilt gaan werken moet je de code onder 16.1.b aanpassen, de code onder c is echt voor Game Over.

Opgave 16.2. Hier geldt, dat ik wel wat antwoorden kan geven die op zich wel correct zijn, maar het zijn niet de enige correcte antwoorden. Zoals ik al vaker aangaf, er zijn vele wegen die naar Rome leiden en ik behandel er maar een paar. Toch een poging:

- Voor mij niet, in de huidige vorm. Ik vermoed dat het voor veel van onze leden ook niet zal meevallen, maar voor de jeugd is er waarschijnlijk weinig uitdaging aan. Voor mij is met name de trage reactie van het batje de oorzaak van falen.
- Je kunt het makkelijker maken door het batje sneller te laten reageren. Op dit moment zit er 1 seconde vertraging in, en in een spel met enige snelheid is dat echt te lang. Je kunt het uiteraard ook makkelijker maken door de bal wat langzamer te laten bewegen. Je kunt alle afbeeldingen iets kleiner maken, waardoor het veld relatief groter is en de bal dan langer onderweg lijkt.
- Moeilijker kan het worden door de punten hierboven om te draaien, maar je kunt bijvoorbeeld ook het batje kleiner maken, bijvoorbeeld: elke keer als een steen geraakt wordt gaat er een stukje af.

Opgave 16.3:

- Er zijn een hoop manieren waarop je het spel kunt uitbreiden. Daar gaan we straks ook aan werken. Denk hierbij aan geluid, grafische effecten, een meer tot de verbeelding sprekende achtergrond en wellicht ook een aangepast batje, waarmee je de bal nauwkeuriger kunt besturen. En misschien kunnen er ook wat verrassingen optreden, bijvoorbeeld een bom die naar beneden valt; als je batje geraakt wordt dan breekt het.
- In het artikel liet ik de bricks het signaal 'einde' verzenden. Dat is mogelijk niet zo heel handig, omdat dan alle klonen dat deden en misschien deden die dat dan niet. In een aangepast versie liet ik dit aan de sprite 'You win' over en nu werkt het:



Figuur 4 - You win

En nu verder

Er zitten hier en daar wel wat schoonheidsfoutjes in het spel. Soms start de bal te hoog, waardoor vanaf het begin al stenen verwijderd worden. Gelukkig worden die wel geteld, dus het spel eindigt nog steeds correct. Maar in het kader van opgave 16.3.a gaan we wat dingen aanpassen en bekijken we of dat leuker is.

Een heel eenvoudige aanpassing, die gelijk al een effect heeft, is het aanpassen van de achtergrond. Rechts onderin zit een knop:



Figuur 5
Kies een achtergrond

Als je op het groene vakje klikt komt er een aantal opties tevoorschijn. Bij (1) krijg je de gelegenheid een eigen afbeelding naar Scratch te uploaden. Bij (2) kiest Scratch zelf een willekeurige achtergrond bij je project (verrassing) en bij (3) kun je zelf een achtergrond tekenen. En bij (4) kun je een achtergrond kiezen uit de uitgebreide bibliotheek. Voor deze aanpassing aan ons project kies ik een achtergrond.

Als je daarvoor kiest, dan krijg je een enorme hoeveelheid achtergronden te zien, waaruit je kunt selecteren. Gelukkig zijn ook de achtergronden onderverdeeld in categorieën en kun je ook nog een keer zoeken. Zie deze zoekbalk:



Figuur 6 - Achtergronden

Voor ons project kiezen we in de categorie 'Ruimte' de achtergrond 'Galaxy' en gelijk ziet ons spel er een stuk minder kaal uit.

Een andere eenvoudige wijziging, die direct effect heeft, hoewel niet iedereen het zal waarderen, is door in de code van de stenen bij het veranderen van de score gelijk het geluid 'Plop' af te spelen. Het is een kleinigheid, maar het geeft een hoorbare feedback als je een steen raakt. Het blokje van de stenen ziet er dan als volgt uit, het geluidsblokje is erbij gekomen:

Links

Als service voor de bezoekers van de lezing over Scratch tijdens de HCC!kennisdag op 8 oktober jl. hierbij een overzicht van de getoonde links:

1. <https://scratch.mit.edu/projects/174095454/editor> een animatie vanuit de TU Delft
2. <https://scratch.mit.edu/projects/174095454/editor> mijn profiel op Scratch, met mijn gedeelde projecten
3. <https://scratch.mit.edu/projects/174095454/editor> project 'Pong', basic pingpong spel
4. <https://scratch.mit.edu/projects/697012542/> het hier beschreven Bricks spel na artikel 16
5. <https://www.compusers.nl/webontwerp/scratch> de Scratch pagina op de CompUsers website
6. <https://inventwithscratch.com> de pagina van Al Sweigart met een compleet online boek over Scratch. In hoofdstuk 5 wordt dit project beschreven.



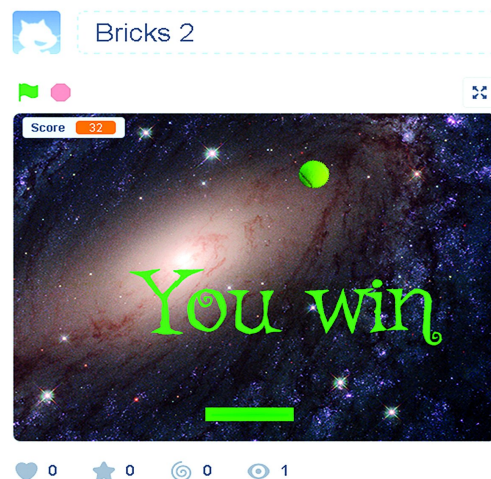
Figuur 7
Geluid toevoegen

Bij (1) zie je de toegevoegde instructie.

Als je in het batje het tempo verandert naar 0.2 dan is het spel eenvoudig speelbaar en daarmee dus goed te testen. In deze vorm ziet het er nu wat toonbaarder uit, we hebben een geluidseffect toegevoegd en de volgende keer gaan we nog wat leuke dingen toevoegen. Maar kom ook gerust zelf met ideeën over hoe dit verder ontwikkeld kan worden.

Game-design is, zoals ik al eerder aangaf, een samenspel van verschillende disciplines. Je kunt als programmeur niet ook nog een keer een expert zijn in alle andere aspecten die erbij komen kijken. Maar als je aan het bouwen bent, dan schieten je soms de ideeën door het hoofd. In Scratch kan je, dat hebben we gezien, met relatief kleine ingrepen grote effecten sorteren. Ik kijk er dus naar uit om hierop door te gaan, totdat we een spel opleveren dat vrij is van fouten, dat lekker speelt en met een aantal levels, oplopend in moeilijkheid.

Als je ook mijn artikelen over Unity volgt, dan zie je dat dat een meer professionele game engine is, maar Scratch zou je bijvoorbeeld kunnen gebruiken op prototypes mee uit te testen, voordat je die finaal uitontwikkelt in Unity.



Figuur 8 - Het spel uitgespeeld

Hier laten we het weer even bij, de SoftwareBus is vol. Maar laat gerust eens weten hoe het met u vergaat binnen Scratch.