

Scratch (10)

René Suiker

In het vorige nummer heeft u de oproep van het bestuur gezien. Binnenkort is er een ALV en daar wordt over de toekomst van CompUsers gesproken en besloten en daarmee impliciet ook over de toekomst van de SoftwareBus. Wat is de rol van Scratch in dit geheel?

CompUsers en de crisis

Ik heb zo'n twintig jaar in het bestuur gezeten en een ruime tien jaar in de redactie en we zien dat we met steeds minder vrijwilligers de zaak draaiend moeten houden. Daarbovenop nog wat verzwarende omstandigheden, en dan wordt het voor veel mensen toch lastig om vol goede moed door te gaan. Daarom de oproep voor versterking en verversing van het kader. Een oproep die ik hier gaarne ondersteun en die geenszins als motie van wantrouwen voor het zittend kader kan worden aangemerkt.

Als we de vereniging willen behouden, moeten we met z'n allen de schouders eronder zetten. En met de mogelijkheid de ALV virtueel bij te wonen (we zullen wel moeten, vrees ik) kunnen alle leden ook deelnemen en stemmen. En virtueel, Corona of niet, daar ligt toch wel de toekomst van de vereniging. Met behulp van Internet kunnen we het voor onze doelgroep mogelijk maken de producten van CompUsers in huis te halen. Maar, dan moeten ze wel gemaakt worden. Dus, vrijwilligers, meedoen!

De rol van Scratch in dit geheel

Nou, Scratch speelt hierbij slechts een bescheiden rol, maar het biedt je wel de mogelijkheid, producten te maken die je via Internet beschikbaar kunt stellen. Je kunt er interactieve toepassingen mee maken die je ook bij andere aspecten van de hobby kunt gebruiken.

Naast mijn reeks over Scratch schreef ik de vorige keer over opleidingen online. In dat kader wil ik wel kwijt dat ik overweeg een cursus op Soofos te zetten over Scratch. In feite een reeks workshops en lezingen aan elkaar geknoopt die ik graag voor CompUsers had willen verzorgen, maar in onzekere tijden denk ik dus ook na over een andere vorm van distributie. Niet zozeer om daar financieel beter van te worden, maar de cursus zal daar dan verkocht worden.

En nu, aan de slag

Na deze 'algemene beschouwingen' zal ik u niet langer vervelen met perikelen maar u weer bij de hand nemen op onze ontdekkingsreis door Scratch. De vorige keer meldde de eindredacteur mij het verwarrend te vinden dat ik de behandeling van de oude opdrachten en het opgeven van de nieuwe opdrachten door elkaar heen deed.

Daar zat wel een zekere logica in, maar vandaag probeer ik het dus op een andere manier. Ik ga eerst de oude opgaven behandelen, dan kijken of ik nog ruimte heb voor nieuwe theorie en dan tot slot geef ik u de nieuwe opgaven.

En omdat we deze keer in de luxe situatie verkeren, dat ik me moet beperken tot drie pagina's, maak ik uitwerkingen iets minder uitgebreid dan u van mij gewend bent, maar ik

zal wel proberen, de belangrijkste aspecten uit de doeken te doen.

De oude opgaven

Opgave 9.1: Kijk of het tellen goed verloopt; zo ja, prima, zo nee, wat gaat er verkeerd?

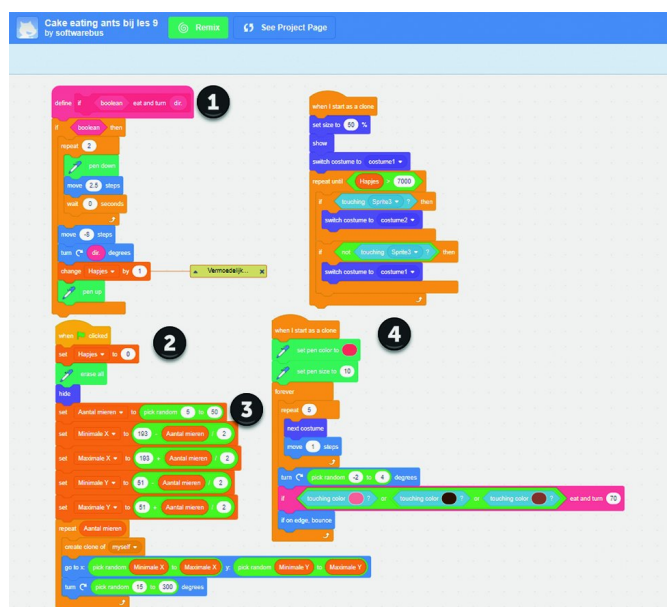
(<https://scratch.mit.edu/projects/38155543/>)

Je kunt natuurlijk op de groene vlag klikken en kijken wat er gebeurt. Het is heel simpel, je ziet twee tellers, één met het aantal mieren en één met het aantal hapjes.

Die hapjes lopen zo rap op, dat je niet kunt volgen of het klopt of niet. Wat moet je dus doen?

Je hebt twee mogelijkheden. Je kunt kijken of je de interne logica kunt volgen en zien of je hier fouten in ontdekt. Bij de ontwikkeling van software in een professionele omgeving is het gebruikelijk dat een collega je broncode beoordeelt. Je kunt ook empirisch aan de slag, maar dan moet je een en ander beter zichtbaar maken.

In onderstaand figuur zie je de code zoals die nu in het project staat. Maar met software die in ontwikkeling is kun je er niet van uitgaan dat dit ook de code is die er staat als je dit



Figuur 1 - code van mieren eten taart

leest.

Als je ziet wat er allemaal gebeurt op het scherm, is het eigenlijk verrassend weinig code die je nodig hebt. Bij blok (1) wordt de functie gedefinieerd die 'eat and turn' heet. Ofwel 'eet en draai'. In deze functie wordt het aantal hapjes telkens met 1 verhoogd als deze wordt aangeroepen. Dat lijkt wel te kloppen, althans, als je een goede startwaarde hebt.

Bij blok (2) wordt gezegd wat er moet gebeuren als er op de groene vlag wordt geklikt. Hiermee start je in Scratch de uit-

voering.

Bij blokje (3) worden een aantal constanten bepaald. In Scratchtermen zijn het nog steeds variabelen, maar de waarden hier worden éénmalig bepaald bij het opstarten en veranderen verder niet meer. Hiermee wordt de leesbaarheid van het programma verbeterd en daarmee wordt het beter mogelijk om te beoordelen of het ook goed werkt.

Bij blokje (4) zie je wat de gekloonde mieren doen. In feite lopen ze wat rond, als ze eten vinden dan nemen ze een hapje en dan draaien ze een beetje om. Als ze tegen de rand aanlopen, dan keren ze om.

Als je wilt vaststellen wat er gebeurt tijdens het draaien van het programma, kun je meerdere dingen doen. Je kunt in elk geval het aantal mieren verminderen, tot bijvoorbeeld één of twee, dat maakt het overzichtelijker.

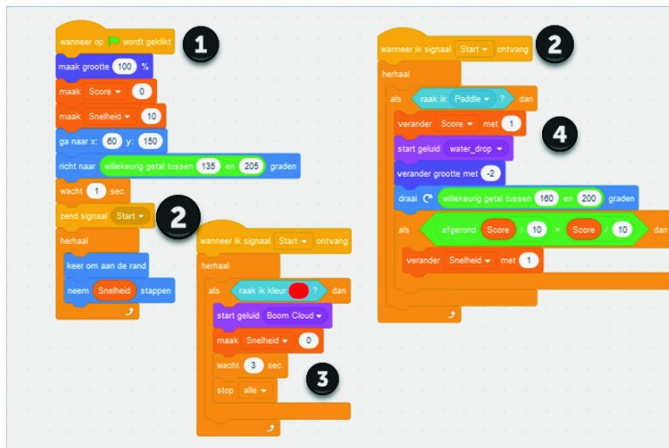
Je kunt de snelheid veranderen, zodat je beter kunt waarnemen wat er gebeurt. En ten slotte kun je ook alle variabelen die je definieert zichtbaar maken op je scherm. Het totaal wordt minder leuk uiteraard, maar je kunt wel beter in de gaten houden wat zich allemaal afspeelt.

Opgave 9.2: Werk de opdracht uit

Het zou bladvulling lijken, als ik deze opgave nog eens herhaal en uitwerk. Alle kennis om dit uit te voeren is in feite in de vorige opgave aan de orde gekomen.

Opgave 9.3: Wat is de complicerende factor bij opgave 8.7? (project 10128515)

Vanaf nu zou het voldoende moeten zijn als ik alleen het projectnummer opnoem. Vanuit de papieren versie kun je toch de hyperlinks niet gebruiken, dus dit zou net zo



makkelijk moeten zijn. Het was de bedoeling dat je niet alleen regelmatig de bal een stukje kleiner maakte, maar ook het batje. Dat was de essentie van opgave 8.7.

Figuur 2 - Code voor pingpongbal

Ik heb geen goede inzendingen ontvangen en heel complex is het allemaal niet, maar de bedoelde factor is dat je tot dusver alles in de code van het balletje kon regelen, daar ook over de variabelen beschikte, maar dat je voor het batje in de code van het batje moet zijn.

Opgave 9.4: Voeg bij de score 50 een extra bal toe

Hiervoor moet je uiteraard weer een kloon aanmaken. Misschien is het makkelijker om de hele code te herzien, zodat je gelijk al start met een kloon en het originele balletje gewoon onzichtbaar ergens wegzet. Dan heb je alle code al klaar en in feite gaat dan in plaats van signaleren, zoals bij blokjes (2) hierboven aangegeven, met klonen het circus van start.

Dat betekent wel dat je de beweging van de bal uit blokje (1)

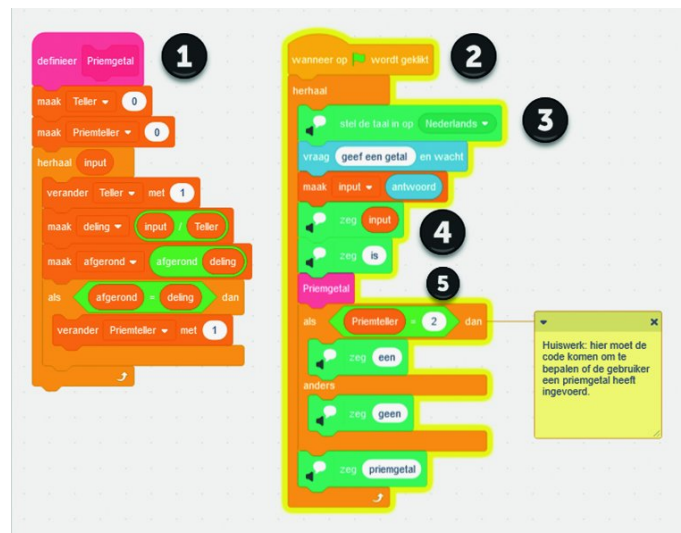
weg moet halen en in blokje (3) of (4) moet onderbrengen. Want je wilt wel dat elk balletje vergelijkbaar gedrag vertoont, maar niet op dezelfde plek beweegt.

Opgave 9.5: Bepaal of een getal een priemgetal is

Een stukje code was al meegegeven, project 388895306

Een priemgetal is een natuurlijk getal, dat alleen maar deelbaar is door twee natuurlijke getallen, te weten 1 én zichzelf. Er is dus welgeteld één even priemgetal, want 2 is deelbaar door 1 en door 2, en één is geen priemgetal, want één is alleen maar deelbaar door 1 getal, dat toevallig wel 1 en zichzelf is, maar dat is niet relevant.

Ik heb een stukje code geschreven om jullie op weg te helpen want ik heb geen bruikbare antwoorden gekregen. Daarbij is weer iets nieuws gebruikt, waar ik dan even op in ga. In het begin van de serie heb ik ooit wel eens gebruik gemaakt van de opdrachten 'zeg' en 'denk', maar 'spreek' is nog niet aan de orde gekomen. Daarvoor moet je ook een uitbreiding toevoegen, 'tekst naar spraak'. Via het blokje uit-



breidingen kun je deze kiezen en dan komen er spraakopdrachten beschikbaar. En met de code 'spreek' kun je de computer laten praten.

Figuur 3 - Code voor priemgetallen (388895306)

Omdat ik nog wat over wil houden voor de nieuwe opgave, heb ik een werkend, maar niet geoptimaliseerd project gemaakt.

In blok (1) definieer ik een eigen blok 'priemgetal'. Zoals ik al schreef, niet geoptimaliseerd. Dat laat ik aan de lezers over. In elk geval, ik heb een paar variabelen gedefinieerd, waarmee ik ga bepalen of iets een priemgetal is.

Dat doe ik heel rechttoe rechtaan. Ik neem de input van de gebruiker en maak een lus die van 1 tot dat getal gaat. Ik maak daarvoor een teller. En ik houd bij hoeveel keer de input deelbaar is door deze 'teller', in de 'priemteller'.

Omdat ik in eerste instantie wat merkwaardig gedrag waarnam, heb ik ook nog een variabele 'deling' en 'afgerond' ingevoerd, waarin ik de deling en de afgeronde waarde opsla.

Deelbaar wil zeggen dat de deling gelijk is aan de afgeronde deling. Tijdens het 'debuggen' had ik alle variabelen op het scherm en dan kan je zien wat er gebeurt. Heel leerzaam.

In blok (2) zie je, zoals gebruikelijk, wat er gebeurt wanneer je op de vlag klikt. De groene blokken met het luidsprekericoontje (leuke voor Wordfeud/Scrabble) zijn dus afkomstig van de 'tekst naar spraak'-uitbreiding.

In blok (3) zet ik de taal op Nederlands. In blok (4) begin ik te spreken. Bij blok (5) roep ik de functie aan die ik zojuist ge-

definieerd heb en als de 'priemteller' gelijk is aan twee, dan hebben we een priemgetal.

Als het kleiner is, dan kan het eigenlijk alleen maar 1 zijn, geen priemgetal, en als het groter is hebben we ook nog een andere deler gevonden en is het dus ook geen priemgetal.

De code is niet geoptimaliseerd, dat gaf ik al aan. Voor zover ik kan beoordelen klopt de logica wel, maar het programma is te foppen. Daarnaast kan het efficiënter gemaakt worden. En daar wil ik graag wat over horen.

Opgave 10.1

Hoe is het programma te foppen? En wat kun je er aan doen? Maak een remix van dit programma en los het op.

Opgave 10.2

Zowel de functie 'Priemgetal' als het hoofdprogramma kan efficiënter. Welke mogelijkheden zie je? Hoe los je het op? Maak een remix van dit programma en los het op.

Bij stem heb je een beperkte keuze. Bij de keuze voor 'alt' hoor je een damesstem. Bij de keuze voor 'tenor' hoor je een herenstem. Bij 'piep' hoor je een kinderstem en bij 'reus' hoor je een zware mannenstem.

Een uitbreiding dus

Links onderin beeld zie je een blauw knopje om uitbreidingen toe te voegen. We hebben nu dus 'tekst naar spraak' toegevoegd.

Technisch gesproken is dit een uitbreiding, die best heel wat



Figuur 4 - Tekst naar spraak

betekent, want het is helemaal niet eenvoudig om een computer te laten praten. Maar het mooie van Scratch is, dat dit allemaal achter de schermen plaatsvindt.

Als programmeur krijg je binnen het codeblok drie nieuwe opdrachten tot je beschikking en daar kun je het allemaal mee doen.

Bij blok (1) zie je dat de uitbreiding onderin staat. Als je die kiest, dan krijg je dus de opdrachten ter beschikking.

Bij blok (2) kun je instellen welke stem en welke taal worden gekozen.

En blok (3) wordt gebruikt om de computer feitelijk te laten



praten. De taalkeuze is niet zo uitgebreid als de taalkeuze waarin Scratch beschikbaar is, maar misschien is dat een kwestie van tijd.

Stemkeuze