

Scratch (19)

René Suiker

HCC!kennisdag

Allereerst, voor wie Scratch nog heel nieuw is, elders in dit nummer beschrijf ik het ook, maar op de HCC!kennisdag op 17 juni a.s. ga ik een workshop of lezing geven over Scratch. En daar kunt u natuurlijk ook vragen stellen. Ik hoop op een grote opkomst. Ik krijg 90 minuten de tijd om hier het een en ander te laten leven.

Ik ga me daarbij vooral richten op beginners. Bij gebleken succes zou ik dat op de CompUfair van 16 september kunnen herhalen.

Gezondheid

De vorige keer had ik wel heel weinig energie en op dit moment is mijn energieniveau nog zeker niet terug zoals het was, maar wel al iets meer dan de vorige keer. Bovendien, toen was er ook nog eens weinig tijd. Deze keer ben ik nog steeds vaak moe, maar heb ik iets uitgebreider de tijd genomen om jullie over Scratch te vertellen. En dus wil ik de vragen die ik had gesteld zelf gaan proberen te beantwoorden. Of dat allemaal in één artikel lukt is nog maar de vraag, maar daar komen we vanzelf achter. Ik denk dat ik maximaal 4 pagina's heb voor dit artikel en dat ik om zo'n tien wijzigingen gevraagd heb. Dat wordt dus een uitdaging. En zo erg is het niet als ik het over twee artikelen moet verdelen.

Objecten

Ik weet dat veel van onze lezers niet meer heel jong zijn, meestal zelfs iets ouder dan ik. Als wij hebben leren programmeren, dan was dat meestal sequentieel, je had één programma en dat deed alles. En als je dus meerdere dingen tegelijk wilde laten plaatsvinden, dan moest je dat zelf programmeren. Ergens in de jaren '80 van de vorige eeuw raakte object-georiënteerd programmeren in gebruik en dat werd gaandeweg steeds populairder. En, ik zei het reeds vaker, Scratch is ook een object-georiënteerde programmeertaal. En dat maakt veel zaken overzichtelijker, zeker als je met zo'n programma als Bricks bezig bent.

Ik weet nog dat ik voor de eerste keer las over object-georiënteerd programmeren en ik begreep er geen snars van. Het was ook allemaal Engels en mijn Engels was toen nog niet zo als nu, maar het hele concept was nog zo nieuw en ik was talen als Pascal en Basic gewend. Maar met Scratch ging het allemaal leven en nu begrijp ik de achterliggende gedachten ook beter, en jullie hopelijk ook. Het gaat in elk geval helpen bij de opgaven als je de uitdagingen bekijkt vanuit objecten. Het voert in dit kader te ver om alle aspecten van object-georiënteerd programmeren nu uit de doeken te doen, maar besef wel dat er meer is dan ik hier nu vertel.

Een van de eerste dingen die ik leerde en nog niet begreep, was dat objecten zowel data als code bevatten. Deze manier van tegen programmeren aankijken was volledig nieuw voor me. Maar in Scratch zie je het ook. Je bekijkt een object, bijvoorbeeld in dit spel de bal. En je ziet bij de bal alle code die betrekking heeft op de bal. En als de bal interactie heeft met andere objecten, dan sturen ze elkaar berichten. Maar

de bal bevat geen code voor het batje en andersom. En variabelen die kunnen voor het hele programma gelden, maar ook alleen maar binnen één object actief zijn.

En als je op die manier naar de opgaven kijkt, dan weet je vaak ook al waar je de oplossing moet zoeken.

HCC!Kennisdag

En nu even niet over de komende, maar de laatste, die van 25 maart jl. Daar woonde ik een lezing bij over programmeren met pictogrammen. Dat ging ook over Scratch, maar ook over andere manieren en zelfs niet alleen over het programmeren van computers, maar bijvoorbeeld ook van groepen mensen. Het ging vooral over ScratchJr, een versie van Scratch die bedoeld is voor nog kleinere kinderen, die nog niet kunnen lezen.

Cobie van de Ven hield een bijzonder levendig betoog namens de SeniorenAcademie en ze liet ook zien dat het goed mogelijk en leuk is om met elkaar samen te werken en dat het ook goed is om met elkaar te communiceren, want door verschillende invalshoeken te gebruiken krijg je de beste oplossingen.

En zo is het dus ook als je met je (klein)kinderen met Scratch aan de gang gaat. De kinderen kunnen van ons leren, maar wij zeer zeker ook van de kinderen. En voor veel kinderen blijkt het toch heel leuk te zijn om de computer iets te laten doen, bijvoorbeeld een spel of het besturen van een robot. En waarschijnlijk snappen de kinderen het werken met objecten makkelijker en eerder, want ze hebben nooit op een andere manier leren programmeren.

Het huiswerk, de uitwerking

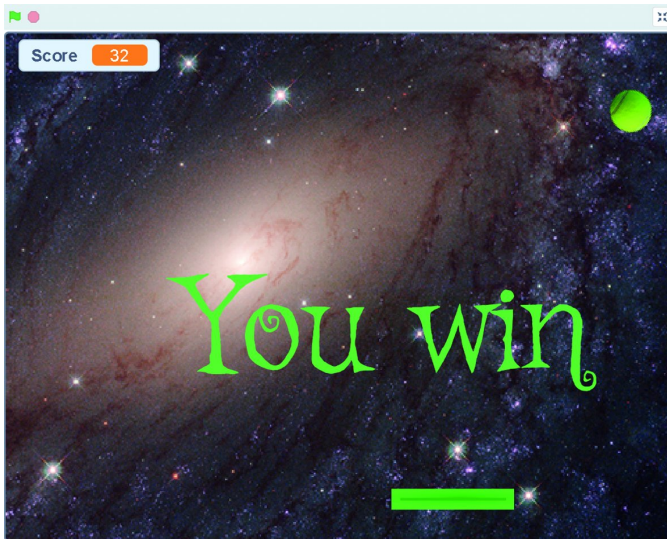
Zoals ik al zei, we benaderen het vanuit de objecten. In dit spel moeten we er een aantal onderscheiden. In beginsel is alles een object, maar laten we ons even beperken tot die zaken die we zien:

1. Het batje
2. De bal
3. De blokjes
4. De tekst 'You win'
5. De tekst 'Game Over'

We hebben in het spel vijf sprites gedefinieerd; die tellen alle vijf ook als objecten. Je kunt meerdere zaken als object definiëren, maar dat werkt nu verwarrend, dus we beperken is nu tot deze vijf. Overigens worden er in deze versie: <https://scratch.mit.edu/projects/740481798/> 32 blokjes getoond, maar dat zijn in feite allemaal klonen van één blokje, dat overigens zelf dan weer niet te zien is.

Waarom dan nu zoveel nadruk op die objecten? Wel, de code in Scratch is altijd gekoppeld aan objecten. Dat kunnen deze sprites zijn, maar als je het voorbeeld iets verder bekijkt, dan kom je er achter dat er ook extra code is. We gaan zo kijken.

Het huiswerk van de vorige keer was volledig gebaseerd op dit project. En ik vertelde ook dat het spel uit te spelen was. Je kon het winnend afsluiten:



Figuur 1 - Eindscherm (winst)

Uiteraard is het ook verliezend af te sluiten, en dat plaatje heb ik in het vorige artikel ook getoond, dat ga ik nu niet herhalen. Je ziet wel, de score is 32 als je het uitgespeeld hebt. Dat wil zeggen, je hebt alle 32 blokjes vernietigd.

Verbeteringen

Zoals ik al aangaf, er treedt iets gekks op. En overigens, ik zei daarbij, als je het spel nog een keer opstart, met het vlaggetje, nadat je gewonnen hebt, bouwt het maar gedeeltelijk op, maar in feite treedt het probleem min of meer random op, dus ook het goed opstarten bij een nieuwe poging is niet altijd (maar wel meestal) het geval. Voor een experimenteel stadium geen ramp, maar als je het spel op de markt zou willen zetten mag zoiets natuurlijk niet gebeuren. Hoewel natuurlijk ook in de praktijk van de commerciële spelontwikkelaars wel eens iets fout gaat. Het testen van software is geen sinecure en is misschien wel een heel artikel waard.

Enfin, dit fenomeen wilden we te lijf gaan met:

Opgave 18.1

Waarom treedt voornoemd effect op?
Hoe kan je dit voorkomen?

Uitwerking:

Veel activiteiten worden opgestart op het moment dat op het groene vlaggetje wordt geklikt. Je weet eigenlijk nooit in welke volgorde deze zaken opgepakt worden. Als je wel een bepaalde volgorde wilt bereiken, is het beter om op basis van de groene vlag je initialisaties te starten en daarna pas met berichten de volgende acties op te starten.

Als de volgorde niet uitmaakt, mits het na de initialisatie gebeurt, kan je volstaan met één signaal, als de volgorde wel belangrijk is, dan kun je ook overwegen meerdere signalen te gebruiken.

Laat het initialisatieproces de juiste startsignalen uitsturen en laat je diverse objecten luisteren naar het voor hen geldende signaal.



Je dient daarvoor bij de diverse sprites het blokje 'wanneer op de groene vlag wordt geklikt' te vervangen door het hiernaast stande blok:

Figuur 2 - Berichten ontvangen

Daarbij kies je dan 'Nieuw bericht' als het bericht nog niet in de lijst staat. En in de initialisatie procedure moet je dan de tegenhanger van dit commando opnemen, namelijk 'Zend signaal' met daarbij het betreffende signaal. Op die manier wordt de initialisatie in de door jou bepaalde volgorde uitgevoerd.

Minder saai

We vinden het spelletje een beetje saai en willen het dus wat aanpassen, want in de huidige setting, als je het een beetje door hebt, win je elke keer. Daarop kwam ik met:

Opgave 18.2

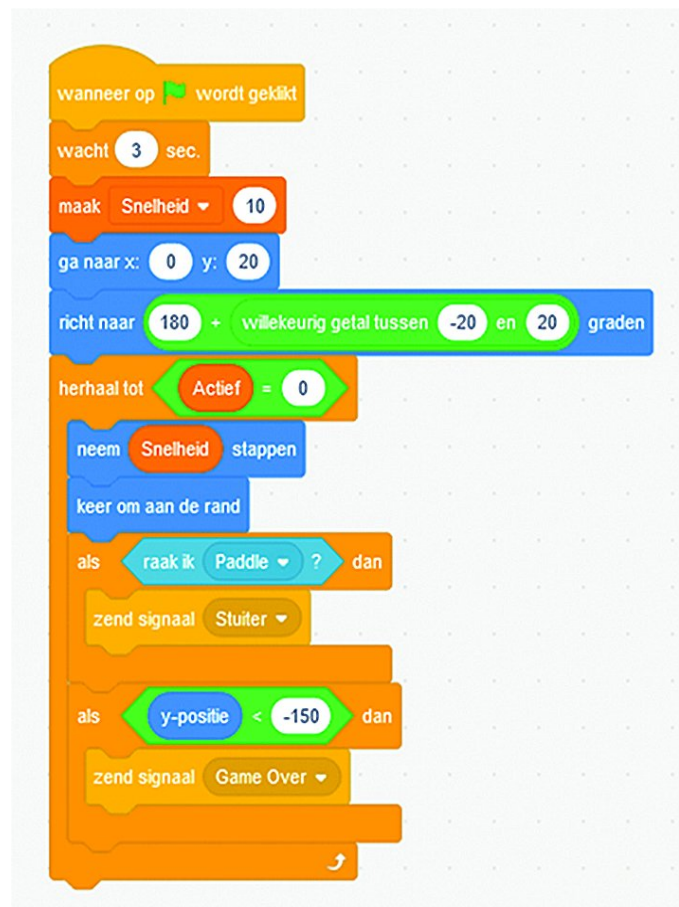
Hoe kun je het beter speelbaar maken? Ik heb zelf een aantal suggesties, je kunt mijn project klonen en dan de volgende maatregelen uitvoeren:

- 1) Naarmate het spel vordert, kan je de bal kleiner maken
- 2) Naarmate het spel vordert, kan je het batje verkleinen
- 3) Naarmate het spel vordert, kan het balletje sneller gaan

Uitwerking:

Ik begon zo uitgebreid over objecten en dat was niet zonder reden. Om de aanpassingen te kunnen doen, moet je dus kijken waar de aanpassing onderdeel van uitmaakt. Wil je iets met de bal doen, dan moet je dus in het codeblok van de bal kijken. Wil je iets met het batje doen, dan moet je in het codeblok van het batje kijken. En wil je iets doen op basis van het verloop van het spel, dan moet je naar de variabele 'score' kijken, of je kunt een aparte variabele definiëren, die op andere zaken kan reageren. Maar de variabele 'score' is zo opgezet, dat die voor alle objecten benaderbaar is.

Dus, bijvoorbeeld de bal verkleinen, dan kijken we naar het codeblok voor de bal:



Figuur 3 - Codeblok voor de bal

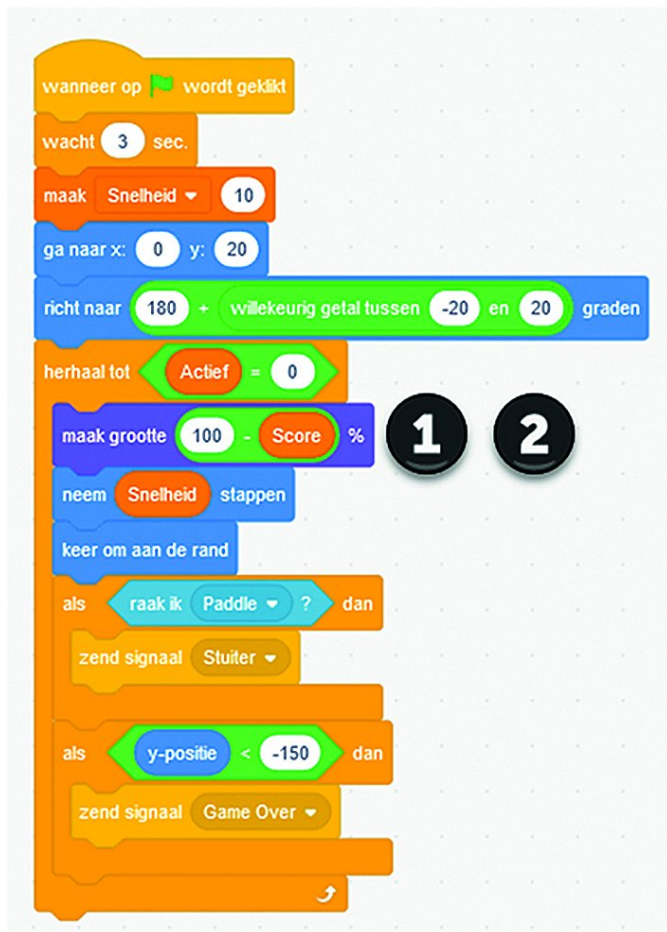
Als je naar het project kijkt, dan zie je nog een blokje, maar dat is een nadere uitwerking van het stuiteren en is op dit moment niet zo relevant.

Als je naar de code kijkt, zie je dat we tijdens de initialisatie naar een positie gaan en min of meer naar beneden richten. En verder kijkt het balletje of het te laag valt, waarna het spel over is. Je ziet nergens een aanpassing van de grootte van het balletje. Blijkbaar gebruiken we het balletje met de grootte zoals die gedefinieerd is. Dat kun je vergelijken met 'Maak de grootte 100%.'

Als je het balletje steeds kleiner wilt maken, dan kun je dat doen door het met zekere tussenpauzes kleiner te maken. Je kunt echter ook het balletje elke keer een grootte geven die afhankelijk is van de score. Op die manier wordt elke wijziging van de score gelijk gevolgd door een aanpassing van de bal.

De eenvoudigste manier is dus door in het codeblok een stap op te nemen uit de categorie 'Uiterlijken' die de grootte afhankelijk maakt van de score. En omdat hij steeds kleiner moet worden naarmate het spel vordert, kun je dus de grootte bijvoorbeeld gelijk maken aan '100 - score'.

Dit voer je in binnen in de spel-lus, dus zo:



Figuur 4 - Aanpassing van de bal

Bij (1) zie je het blokje uit 'uiterlijken' om de grootte van het object aan te passen. De (2) staat er nog naast, omdat je het groene blokje uit 'functies' er nog in moet stoppen om de twee waarden van elkaar af te trekken.

Je kunt dit gelijk uitproberen en je ziet het balletje kleiner worden. Om het effect te versterken kun je ook nog een vermenigvuldiging op de score toepassen, dan wordt het balletje sneller kleiner. Pas wel op dat je het balletje niet te klein maakt, want als je het niet ziet, kun je het ook niet bespelen. Dus een factor 1,5 of 2 is realistisch, daarboven wordt het echt een geluksspel.

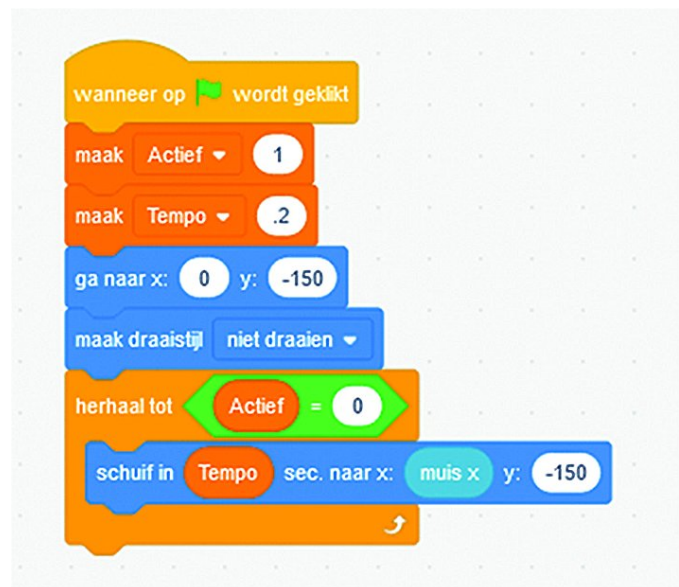
Als je het echt netjes wilt doen, dan maak je tijdens de initialisatie de score 0 en maak je ook het balletje 100%. Als

dat dan voor de eerste keer verschijnt bij de start van het spel heeft het altijd dezelfde grootte. Anders begint het klein en wordt het meteen weer groot bij de eerste keer in de loop gaan. Dat is een ongewenst effect.

De volgende suggestie was om het balletje sneller te laten gaan naarmate het spel vordert. We zien in het codeblok dat er in de lus gestapt wordt met 'snelheidsstappen'. Dit is vrij langzaam. Je zou hier de score bij op kunnen tellen. Dat heeft gelijk een vrij drastisch effect, dus misschien wil je de halve score erbij optellen. Kijk maar wat je leuk vindt, de plaats om het aan te passen en het gebruik van de score is nu voldoende beantwoord, denk ik.

Het batje

Het codeblok voor het batje kan bijna niet simpeler:



Figuur 5 - Code voor het batje

Je ziet dat hij in een bepaald tempo de x-waarde van de muiscursor volgt, terwijl de y-waarde constant op -150 blijft. En misschien herinner je je nog dat als de bal op -150 kwam, hij het signaal 'game over' gaf. Dan is de bal dus op de hoogte van het batje.

Je ziet ook dat we hier ook niets met de grootte hebben gedaan. Omdat de score niet aan de bal is gekoppeld maar algemeen beschikbaar is, kun je de ook hier de score gebruiken om het batje kleiner te maken gedurende het spel. En je kunt ook iets met tempo doen zodat, naarmate het spel vordert, het steeds langer duurt voordat het batje de cursor volgt. Je kunt hier mee spelen, maar maak het effect niet te sterk, want dan volgt het batje helemaal niet meer. Als het meer dan een seconde duurt, dan is de bal al beneden.

Hiermee heb ik denk ik voldoende basis gegeven om opgave 18.2 af te kunnen ronden. Kijken we nog even naar de game, dan zien we dat het balletje bijna altijd ofwel één blokje verwijderd, ofwel geen, omdat het plafond geraakt wordt. Dat is best wel eentonig en hoewel we de moeilijkheid wat hebben aangepast, kan er nog veel meer leuks:

Opgave 18.3

In de Arcade-versie stuitert het balletje soms boven de blokjes en haalt dan wat blokken weg zonder dat je als speler iets hoeft te doen. Ik denk dat je het volgende moet doen, maar sta open voor betere suggesties:

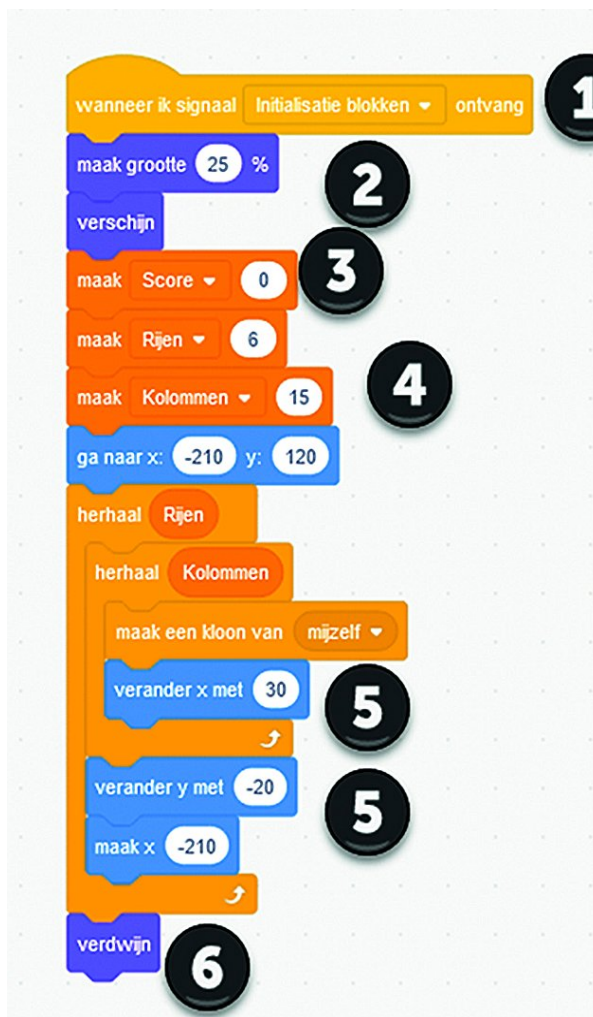
- 1) Blokken en balletje iets kleiner
- 2) Blokken iets lager plaatsen zodat er ruimte is Voer deze verbeteringen door in een kloon.

Uitwerking:

Met alles wat we al behandeld hebben wordt het tijd om de initialisatie eens finaal te bekijken. Wat willen we allemaal doen in de initialisatie:

1. De score op 0
2. Het speelveld leegmaken, m.u.v. achtergrond
3. De grootte van de diverse objecten aanpassen
4. De snelheden op de gewenste waarde zetten
5. Vervolgens het veld van blokken opbouwen
6. Vervolgens het batje tonen
7. Vervolgens het balletje tonen
op een geschikte hoogte, dus onder de blokken maar wel boven het batje
8. De muziek starten
9. Het balletje laten vallen en de acties starten.

Kijken we dan naar de blokken, dan willen we die gedurende het spel niet van grootte laten veranderen, maar we willen ze wel kleiner maken dan ze nu zijn. Op dit moment hebben we grootte 40% ingesteld en dan kunnen er acht op een rij, dus dat mag wel iets meer. We willen ze ook iets lager laten starten en met één of twee rijen meer niet lager laten uitkomen dan nu. Dus ik denk dat 25% een goede omvang is. Dat is een kleine moeite en dus proberen we uit:



Figuur 6 - Code voor de blokken

Zie hier de aangepaste code, maar dit is natuurlijk nog niet alles. We moeten er wel voor zorgen dat we dat bericht/signaal ook sturen tijdens de initialisatie. Verder moeten we dan wel even kijken hoe de 'You win' wordt geactiveerd, want als die al bij score van 32 activeert, dan is het spel ook snel afgelopen. Als je zes rijen van 15 blokken hebt, dan kan de score oplopen tot 90. En bij 90 wordt de bal al erg klein.

Als je dan ook nog de snelheid flink opvoert bij score 90, dan is het misschien niet meer te halen. Al met al, leef je uit en

maak een versie die speelbaar is, die dus uitgespeeld kan worden, maar die ook uitdagend is, dus niet elke keer gewonnen kan worden.

De vorige keer had ik nóg een opgave in huis, maar die gaan we deze keer niet meer behandelen, want de ruimte ontbreekt. Ik had het al aangekondigd, maar ik denk dat het niet erg is. Daar ga ik dan gewoon de volgende keer mee verder, ik hernummer gewoon de opgave.

Opgave 19.1 (voorheen 18.4)

Om de variatie iets te vergroten heb ik nog de volgende suggesties, voor de iets gevorderde programmeurs:

- 1) Maak een paar blokjes rood. Als deze geraakt worden kleuren ze blauw. Als ze nog eens geraakt worden, verdwijnen ze.
- 2) Geef de speler dan drie ballen voordat hij af is.
- 3) In mijn beleving ging in de Arcade-versie het batje altijd heen en weer, als je de rechter pijltoets indrukt gaat hij langzaam rechtsaf tot de rand en blijft daar, idem voor linksaf. Maak deze besturing.
- 4) Als je de bal raakt in het midden van het batje, dan blijft het spel zoals het is. Als je hem helemaal rechts raakt, stuiter hij iets meer naar rechts, als je hem helemaal links raakt, stuiter hij iets meer naar links. Helemaal links en rechts is, zeg maar, 25% van de lengte.

Ik zou zeggen, probeer dit eens uit en kijk hoe ver je komt. Ik zal de aangemelde programma's bekijken en in het volgende artikel ga ik erop in. Ik hoop dat er veel goede antwoorden binnenkomen.

En omdat ik ook nog iets nieuws wil inbrengen krijgen jullie ook nog een theorievraag. Als je goed naar de blokken hebt gekeken, zien jullie nergens de muziek starten. Dus, los van de benoemde objecten is er blijkbaar nog een object.

Opgave 19.2

Wat is dat mysterieuze object, dat tot nog toe niet benoemd is?

Opgave 19.3

Definieer de initialisatie in dit mysterieuze object en bepaal hoeveel signalen er nodig zijn om het geheel in goede banen te leiden.

Opgave 19.4

Hoe kun je bereiken dat de timing echt correct is bij het uitvoeren van de initialisatie, ongeacht het aantal blokjes dat je maakt of de snelheid die je gebruikt?

Nu, dat ziet er weer uit als een mooi onderwerp voor de HCC!kennisdag op 17 juni. Eens kijken of we met beginners hier kunnen komen. Tot ziens daar, of in het volgende artikel.

