

Scratch (3), het vervolg

René Suiker

Wat vooraf ging...

En zo is het alweer 2019 en is het tijd voor aflevering 3. Een korte recapitulatie.

In deel 1 gaf ik een inleiding in scratch. In deel 2 gaf ik daar een aanvulling op en begonnen we met programmeren. In deze aflevering gaan we hiermee verder, maar ga ik ook nog even in op wat andere overwegingen.

We gingen in op sprites, op events, op lussen, op pen op en neer, in feite op de basiselementen om een programma te maken dat reageert op prikkels van de omgeving en dan vervolgens enige logica hanteert.

En nu gaan we ook eens tijd maken voor 'anderen'. We hadden het al aangekondigd, Scratch was leuk om met kinderen te doen. Liefst niet te grote kinderen, want die zullen het misschien weer te kinderachtig vinden, maar leeftijd basisschool is zo'n beetje waar je aan kunt denken. Hoewel er natuurlijk ook altijd kinderen buiten die categorie zijn die het wel leuk vinden. En binnen de categorie heb je ook kinderen die er niets aan vinden. Het blijft natuurlijk iets wat u het best voor uw (klein)kinderen kunt aanvoelen, er is geen universele waarheid.

Met anderen

Zo ving ik op het werk wat op in een gesprek tussen collega's over Scratch, van wie er een enthousiast over aan het vertellen was, dat kinderen het heel leuk vonden. En dat er bij hem in de buurt ook bijeenkomsten waren waar het uitgelegd werd, waar kinderen konden oefenen met computers die al klaar stonden en dat ze met vragen geholpen werden. En zo maakte ik kennis met het fenomeen 'Coder(Dojo)'. Om eens een kijkje te nemen ga ja naar:

<https://coderdojo.nl>

Misschien is dit iets om als CompUsers mee samen te werken, want dan staat de nieuwe generatie doelgroep gelijk klaar. Maar dat moeten we dus nog even uitzoeken. Ze zijn in elk geval al verbonden met wat andere instanties, dus waarom niet CompUsers, of HCC. We zouden zelf een Dojo kunnen opstarten, als één van de mogelijkheden.

Ik ga natuurlijk niet dit hele artikel aan CoderDojo besteden, en CoderDojo is niet alleen Scratch. Maar het is toch wel interessant genoeg om er even bij stil te staan. Want we zien in de vereniging de veroudering toeslaan, maar jongelui worden heel enthousiast van zelf programmeren.

Van hun site citeer ik:

CoderDojo is een open source, door vrijwilligers opgezette, beweging, die staat voor het houden van gratis, non-profit development clubs en reguliere sessies voor jonge mensen. Omdat CoderDojo open source is kan elke Dojo anders zijn en volledig los staan van andere Dojo's.

Bij een Dojo leren jonge mensen, tussen de 5 en 17 jaar, programmeren, websites bouwen, apps ontwikkelen, programma's maken, games maken, en meer. Dojo's worden opgezet en georganiseerd door vrijwilligers. Naast het leren programmeren, ontmoeten deelnemers ook andere, like-minded¹ mensen, delen deelnemers waaraan ze aan het werken zijn en maken ze nieuwe dingen. CoderDojo maakt development en leren programmeren een leuke, sociale en 'kick ass'-ervaring². CoderDojo legt ook veel nadruk op open source en gratis software en heeft wereldwijd een breed

netwerk aan vrijwilligers. CoderDojo heeft slechts één regel: 'Boven Alles: Wees Cool!', pesten, liegen en mensen hun tijd verdoen is niet cool. Om aan een Dojo mee te doen kun je er een bij jou in de buurt zoeken of er zelf een opstarten.

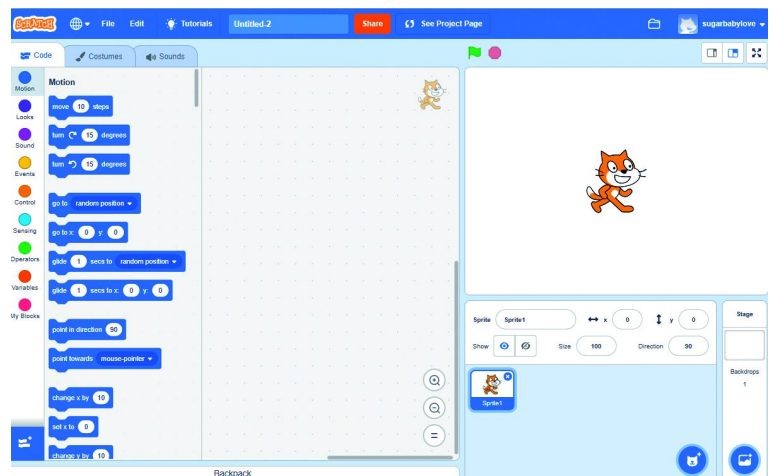
Oftewel, wat wij ook doen, met vrijwilligers, gratis leuke dingen doen met de computer. Ze werken met Scratch, dat doen wij ook, ze werken met HTML, JAVA, zaken waar je ook bij ons voor terecht kunt als het Platform WebOntwerp weer opgericht is, ze werken met Arduino, dat je bij ons bij Domatica aantreft. Enfin, hoezo nou vergrijzing? Kijk eens wat er daar gebeurt, kijk ook eens op <https://coderdojo.com/nl-NL> en neem je (klein)kind bij de hand en ga leuke dingen doen. En, niet onbelangrijk, de kennis en vaardigheden die ze hierbij opdoen zijn mogelijk heel belangrijk voor hun toekomst.

Goed, genoeg reclame, we gaan zelf weer aan de slag, want daar is het toch om begonnen.

Scratch zit niet stil

We starten Scratch weer op via <https://scratch.mit.edu/>

We worden meteen geconfronteerd met het nieuws dat er een nieuwe versie is uitgekomen: versie 3.0. En dat heeft wat te betekenen, want zo is bijvoorbeeld de hele werkomgeving omgegooid.



Afbeelding 1 - Vernieuwde werkomgeving

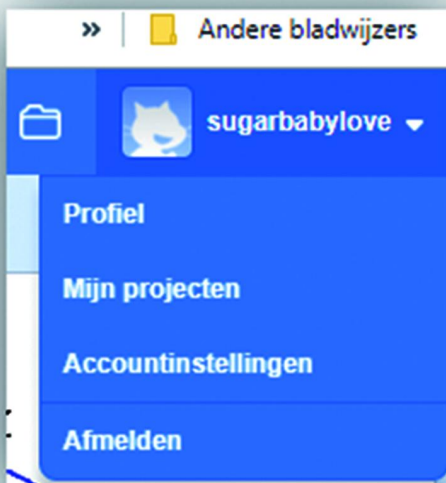
Het idee is nog wel hetzelfde, maar de positionering van de onderdelen is gewijzigd. De bibliotheek van alle commando's bevindt zich nu links. En het schijnt dat er de nodige blokken bij zijn gekomen in deze nieuwe versie. En helemaal links zie je dan de categorieën, daarnaast de actuele opdrachten. Dan zie je in het midden het werkgebied. Daarboven zijn de tabbladen, in dit geval voor 'code', 'uiterlijken' en 'geluiden'. Normaal gesproken staat hij op 'code' en kun je in je werkgebied je programma bewerken. Maar je kunt dus ook de geluiden bewerken en de sprites of achtergronden bewerken. Overigens, als je een ander tabblad kiest, dan verandert ook het gedeelte links. De hierboven beschreven opdrachten zijn aan het werkblad 'code' gekoppeld. Kom je in de grafische editor of ik de geluidseditor, dan zie je links de geluiden of grafische elementen die je kunt bewerken. Rechts daarvan is het uitvoeringsgebied, ofwel speelveld, waar de acties uitgevoerd worden. Zoals jullie ongetwijfeld nog weten, kun je dat gebied ook maximaliseren, dan wordt het beeldvullend.

Onder het speelveld zie je een gebied waar je alle sprites kunt opslaan die je wilt gebruiken. Daarnaast nog een stukje met de naam 'speelveld', waar je je achtergronden bewaart.

Wat me de vorige keer niet opviel, dus misschien was het er ook niet, maar dat weet ik niet zeker, is dit: in het werkgebied, nu weer terug op 'code', zie je rechts bovenin een gebleekte afbeelding van de geselecteerde sprite. Dat geeft aan dat de ingevulde code betrekking heeft op die specifieke sprite. Daar gaan we straks verder op in.

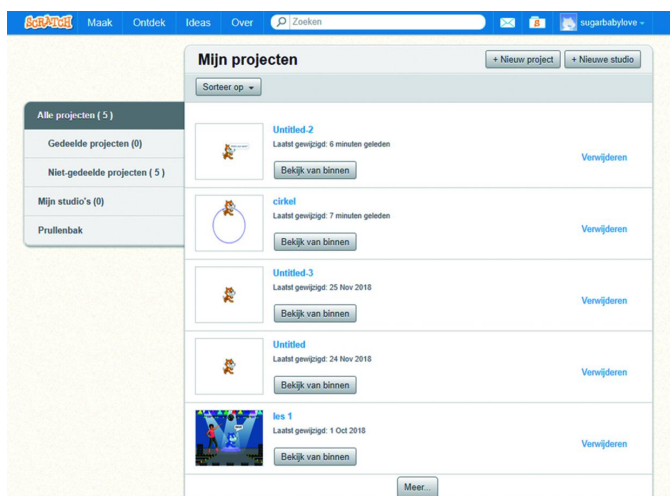
Mijn project

Vorige keer ben ik geëindigd met mijn projectje, een programma dat een cirkel ging tekenen. Dat was nog in Scratch 2.0. Een mooie proef; hoe zou die nu laden, in Scratch 3.0? Welnu, dat viel reuze mee, want ik kon mijn programma gewoon openen. Rechts bovenin klik ik op het driehoekje naast mijn gebruikersnaam en vervolgens klik ik dan op 'Mijn projecten'.



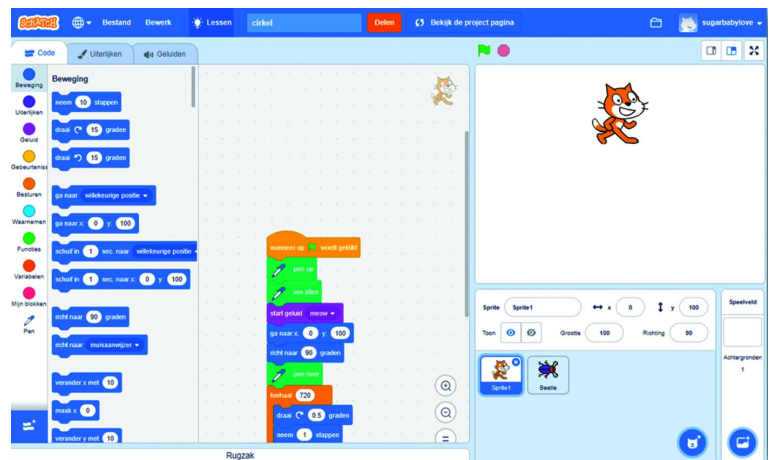
Afbeelding 2 - Mijn projecten

Je krijgt dan een overzicht van je projecten en daaruit kun je er dan één selecteren. Let wel, in principe zijn deze projecten in de cloud van MIT opgeslagen, je kunt er dus ook bij als je, bijvoorbeeld op de CompUfair, hier verder aan wilt werken.



Afbeelding 3 - Mijn projecten voor CompUsers

Ik zie hier de projecten die ik in het kader van dit artikel heb gemaakt, of waar ik aan begonnen ben en ik open project cirkel, gemaakt in Scratch 2.0, maar dat probleemloos opent in 3.0. Overigens hoeft dat niet voor al uw projecten te gelden, er zijn wat problemen bekend, maar daarvoor verwijs ik naar de website. Overigens krijg je bovenstaand overzicht ook als je op het mapjes-icoon klikt. Uiteraard ziet het er nu wel iets anders uit, want de wijziging in de omgeving heeft zijn invloed:



Afbeelding 4 - Project 'cirkel'

U ziet geen meldingen en als ik op het groene vlaggetje klik hoor ik een miauw en gaat de kat een rondje lopen, ondertussen een cirkel tekenend. Tot zover geen probleem. De oplettende lezer die nog goede ogen heeft is misschien iets opgevallen? Een verschil tussen figuur 1 en figuur 4? Ik heb tussendoor de taal weer even op Nederlands gezet. Ik weet niet of het met versie 3 of gewoon met verstreken tijd of iets anders te maken heeft, maar hij was weer naar Engels teruggesprongen. Weet u het nog? Het wereldbolletje naast het SCRATCH-logo links bovenin. Daar kunt u de taal selecteren.

Wat misschien ook opvalt is dat in het speelveld de kat staat, maar eronder, in het sprite veld, staan twee figuren opgesteld. Eén ervan staat zichtbaar in het speelveld, de kat dus, de ander staat nog niet zichtbaar in het speelveld, maar is wel beschikbaar.

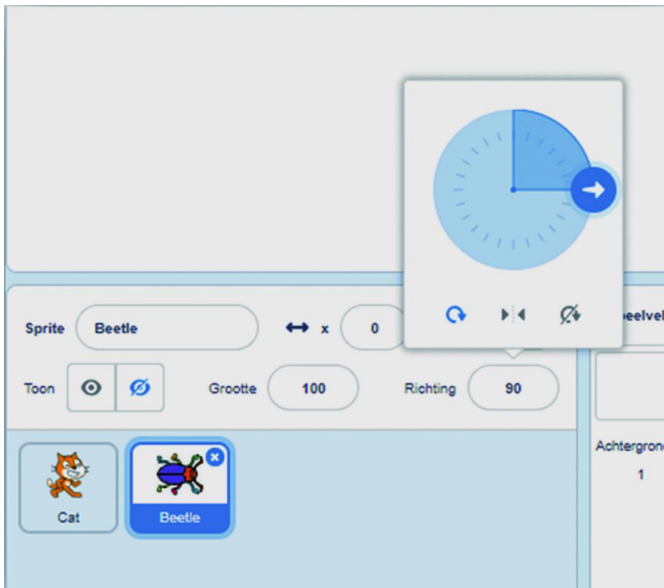
We kijken eens naar het sprite-gedeelte, want dat was mijn bedoeling.



We kijken er eens goed naar. We zien nu twee sprites, waarvan er één geselecteerd is: die in het blauw. Daarboven zie je de naam van de sprite. In dit geval staat er bij de kat 'Sprite 1' of zo iets en bij het insect zie je 'Beetle' staan. Boven de figuurtjes zelf zie je de details van de geselecteerde figuur.

Afbeelding 5 - Sprite-veld bij het speelveld

Hier kun je details ook aanpassen. Zo kun je de naam van de kat veranderen, maar je kunt ook zijn positie veranderen, door andere x- en y-coördinaten op te geven. Verder kun je de looprichting ook nog aanpassen. Als je op het veldje 'richting' gaat staan, dan wordt er een cirkel getoond met daarin de looprichting aangegeven.



Afbeelding 6 - Looprichting visueel

Standaard staat de looprichting op 90 graden. Zoals vorige keer besproken: dit is de richting waarin de sprite beweegt als er stappen genomen worden.

Onder de naam van de sprite kun je aangeven of hij zichtbaar moet zijn in het speelveld, of juist onzichtbaar. Daarnaast kun je de grootte nog aanpassen. Als je dit doet, dan zie je gelijk het effect in het speelveld. Als je de kat bijvoorbeeld in plaats van 100 de waarde 10 geeft, dan wordt het echt een speldenpuntje.

Onderin het veld is nog een knop om extra sprites toe te voegen.

In mijn originele cirkelprogramma had ik de kat een rondje laten lopen, terwijl hij een cirkel tekende. Als je nu de kat onzichtbaar maakt, dan kan het programma nog steeds lopen, maar dan zie je alleen maar de cirkel getekend worden, en niet de sprite.

Vorige keer had ik een aantal zaken geregeld aan het begin van de programma-uitvoering, waaronder de startpositie. Hoe meer je leert tijdens het ontdekken van Scratch, des te uitgebreider wordt ook je set van commando's aan de start van je programma. Want als ik per se zeker wil weten dat de kat ook zichtbaar is, dan moet ik in mijn programma zorgen dat hij verschijnt, ook als de gebruiker op dat moment de sprite niet zichtbaar heeft staan. Gelukkig hebben we daar ook een commando voor, in het blok 'uiterlijkheden', het paarse blok.

Ik plaats dus het blokje 'verschijn' in mijn code en wel ergens bovenin, zodat hij in elk geval niet binnen de lus telkens aangeroepen wordt. Want elke instructie kost tijd. Het is dus zaak goed na te denken over wat je binnen een lus zet en wat niet.

Samenwerken

Ik heb mijn programma nu zo aangepast dat de twee sprites ieder een halve cirkel gaan tekenen. Daartoe kopieerde ik het codeblok van de kat naar het codeblok van het insect. Daarna paste ik de beginpositie aan en de aanvangslooprichting. In beide stukjes code stelde ik het aantal herhalingen in op de helft, omdat ze nu nog maar ieder een halve cirkel hoeven te tekenen. De eerste keer mislukte natuurlijk, omdat ik niet goed had opgelet hoe groot de cirkel uiteindelijk werd, dus sloten de delen niet goed aan. De straal

was wel precies hetzelfde, maar ik moest de positie in de code aanpassen.

Iets meer over de blokken

De code voor de kat ziet er nu zo uit:



Afbeelding 7- Code voor de kat

Je kunt voor jezelf ongetwijfeld nagaan wat al die blokken precies doen, die kennis moeten jullie intussen hebben. En als je de startpositie van onze tweede sprite wilt weten, dan moet je kijken waar de kat ophoudt. Als die twee posities overeenkomen, dan gaat het precies goed.

Nog even over de blokken zelf. Naast het kleurverschil valt vast nog iets op. Er zijn verschillende vormen te zien. De meest voorkomende vorm is het blokje met bovenin een kuiltje en onderin een uitstulping. Deze passen in elkaar. De blokken in deze vorm worden wel de 'stapelblokken' genoemd.

Daarnaast zie je hierboven ook een bijzondere soort stapelblok, dat een blok boven en onder kan hebben, net als een normaal stapelblok, maar dat ook een blokje (of een set blokken) er tussenin kan hebben. Dit is de lus.

Verder zie je hier in de blokken de witte velden, waarin je waarden kunt invullen. Als dat een afgerond vakje is, een cirkel of een ovaal, dan wordt een numerieke waarde verwacht. Is het een rechthoek, dan kun je ook een tekst invoeren. Je kunt dan ook een nummer invoeren, maar dat wordt in principe als tekst geïnterpreteerd.

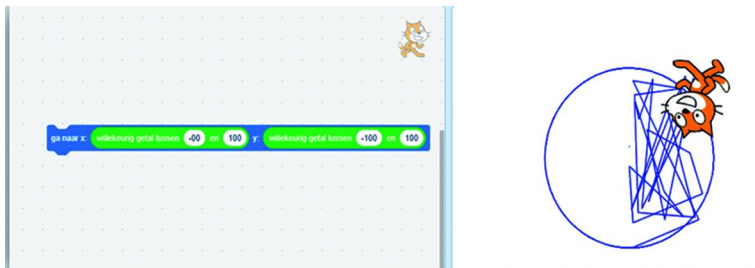
Dan zie je bovenaan nog het blokje met een uitstulping naar boven en naar beneden. Hier kunnen dus wel blokken onder worden gezet, maar niet erboven. Deze blokken starten de uitvoering van de onderliggende code, als de gebeurtenis

welke het blok benoemd wordt uitgevoerd. Overigens is de kracht van programma's natuurlijk dat je niet altijd alle waarden van tevoren invult. Variabelen gaan we de volgende keer bespreken, maar je hebt nog een paar andere vormen, en één ervan is een afgerond blok, zoals dit:



Afbeelding 8 - Afgerond blok

Zo'n afgerond blok kun je plaatsen binnen zo'n wit cirkeltje. Het kan er bijvoorbeeld bij een startpositie zo uitzien:



Afbeelding 9 - Willekeurige startpositie

Ik weet niet of het plaatje heel leesbaar wordt, maar ik heb dus even een blokje gepakt, zoals in figuur 7 wel duidelijk leesbaar is, met een 'Ga naar x: ... y: ...' invullen (blauw blokje - beweging). Daarbinnen heb ik voor x en voor y het groene blokje van figuur 8 ingevoerd en dit heb ik als zelfstandige instructie uitgevoerd. Omdat de pen nog naar beneden stond, kun je in het speelveld mooi het spoor volgen dat de kat heeft afgelegd tussen de diverse punten. Ik had de willekeurige waarden laten variëren tussen -100 en 100.

U had misschien al opgemerkt dat er ook een blauw blokje bestaat met de opdracht 'ga naar willekeurige positie'. Dat heeft niet helemaal hetzelfde effect, probeer het maar eens uit. Het effect is, dat willekeurig ook echt willekeurig is, en met de willekeurige getallen kun je de willekeurige begrenzen. Er zijn nog meer varianten die we in de loop van de tijd gaan bekijken.

Maar eerst nog even een klein zijspiongetje. Ik zei net zo makkelijk: 'kopieer die code van de kat naar ons insectje'. Om dat eenvoudig te doen hebben we helemaal onder in beeld een rugzak. Die kun je gebruiken om stukken code op te slaan en op een andere plek weer te gebruiken. Had ik al verteld, dat je oneindig veel blokjes hebt, dus instanties van hetzelfde blokje? Als je iets uit je rugzak gebruikt, blijft het er ook nog eens gewoon in zitten. Erg handig. Ik spring natuurlijk van de hak op de tak, maar dat komt omdat hier geen expert aan het woord is die even uitlegt hoe het werkt, maar iemand die jullie deelgenoot maakt van een ontdekkingsreis door Scratch. Ik weet nog lang niet alles, maar ik word wel steeds enthousiaster om ermee door te gaan. En ik nodig jullie uit om hier ook zelf in te duiken, misschien eerst aan mijn hand, maar toch steeds meer op eigen benen.

De eerste stappen

De eerste stappen zijn misschien wel het moeilijkst, want je begint weer aan iets nieuws, en daar voel je je misschien te oud voor. Maar zie het als terug naar je jeugd, met Lego, Meccano of Fischertechnik spelen, maar dan op je computer. Je wordt er weer jong van, de computer doet weer dingen die je hem opdraagt.

We zijn natuurlijk van de applicaties, de toepassingen. Maar met domotica sturen we in huis van alles aan, via zo'n Ardui-

no-apparaat. Die schijnen ook met Scratch bestuurd te kunnen worden, in plaats van met alleen maar ingewikkelde talen als C++.

Huiswerk

Intussen heb ik zonder dat je het misschien helemaal door had, al een aardige inleiding Scratch bijelkaar geschreven. En daar ga ik ook mee door, maar ik wil niet elke keer alles moeten herhalen, dus ga eens proberen of je het nu ook kunt.

Ik wil op een komende CompUfair een keer een Scratch-workshop verzorgen, waar we zelf kunnen gaan spelen. Ik zal daar wel e.e.a. moeten toelichten, maar het zou fijn zijn als dan sommige lezers al een beetje verder zijn en mij bij kunnen staan.

Daarom een paar eenvoudige opdrachten, om eens te kijken of er nog iets is blijven hangen.

Vorbereiding

Regel een account bij MIT, waardoor je je projecten ook op kunt slaan. Dit kost niets en is in een mum van tijd geregeld.

Opdracht

Maak een programma dat zonder zichtbare sprites ergens in het midden van het scherm drie concentrische cirkels tekent.

Punten van aandacht

- Programma wordt gestart door het groene vlaggetje
- Startpositie voor het programma, wat moet er allemaal aan- en uitgezet worden?
- Hoe teken ik een cirkel?
- Wanneer ben ik klaar?
- Werkt het programma, doet het wat het moet doen?
- Hoeveel regels code heb je nodig gehad? Kan het met minder?
- Hoe lang duurt de uitvoering? Kan het sneller?
- Zijn aanpassingen op (6) en (7) aanvullend, of werken ze elkaar tegen?

Ik hoop dat jullie hier een uitdaging in zien en ik zie graag wat stukken code uit jullie handen komen.

En om nu te voorkomen, dat we allemaal screenshots met elkaar gaan delen, nog als toegift het volgende. Bovenin je Scratch-venster zit een knopje 'Bekijk de projectpagina'. Als je daarop klikt, dan krijg je de mogelijkheid je project te delen. Het maakt niet uit dat je een heel eenvoudig project hebt gemaakt, er staan miljoenen heel eenvoudige projecten online, en niemand die daar aanstoot aan neemt.



Afbeelding 10 - Naar de projecten pagina

Via <https://scratch.mit.edu/projects/264866296/> kunnen jullie mijn cirkel zien. Gewoon de URL kopiëren, dan kun je die aan anderen beschikbaar stellen. Dat is nog iets anders dan dat een ander het ook kan wijzigen, maar je kunt ze wel naar de inhoud laten kijken. Dat heet dan 'Bekijk van binnen'.

Veel plezier weer met Scratchen.

1. like-minded = gelijkgestemde
2. ass-kicked = ruig, wild, baldadig (Am.-Eng. slang)