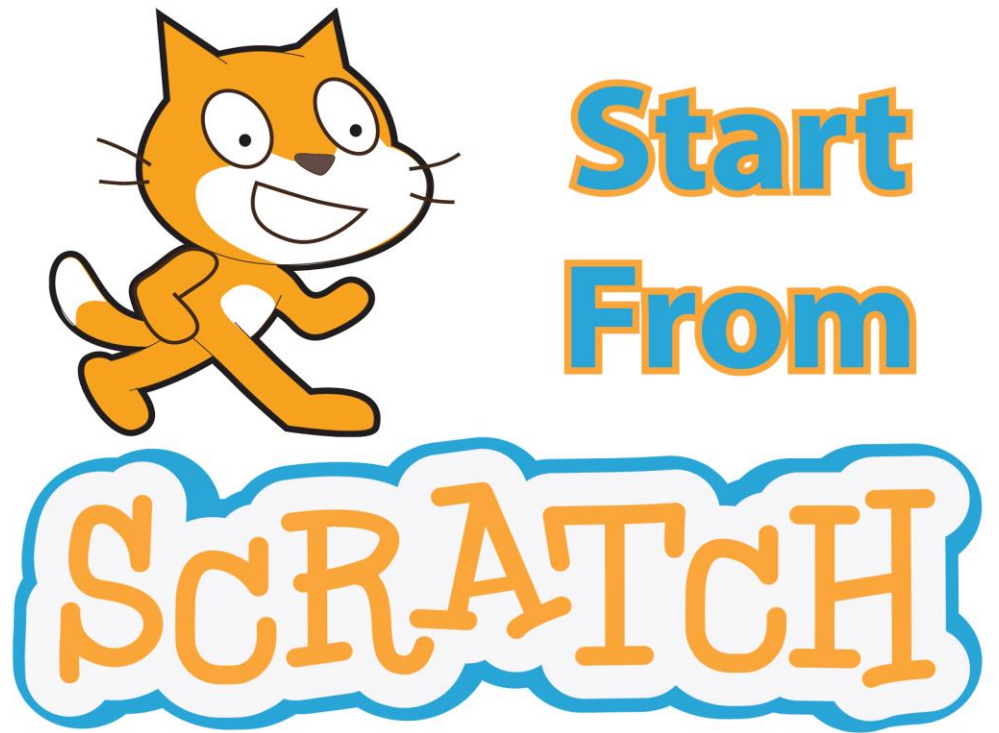




Introductie

1 INTRODUCTIE

Beste meester, beste juf, beste leerkracht, beste instructeur

Je leest dit bestand omdat je geïnteresseerd bent in het aanleren van programmeren aan je leerlingen en dit dan eventueel via Scratch. Ik ben een student in mijn 3^{de} jaar van de bachelor opleiding secundair onderwijs aan de Thomas More Hogeschool te Mechelen. De cursus die je nu leest is het gevolg van mijn educatief ontwerponderzoek dat ik doorheen het jaar heb doorlopen.

De bedoeling van deze cursus is om een inleiding tot programmeren aan te bieden met het programma Scratch. Dit omdat het gaat over een gratis te gebruiken toepassing en omdat het een zeer uitgebreid en krachtig programma is. Leerlingen kunnen snel en eenvoudig leren programmeren terwijl de moeilijkheidsgraad en de complexiteit van de oefeningen en projecten heel hoog kan oplopen. Verder is Scratch ook een toekomstgerichte keuze gezien het samenwerkingsverband tussen Scratch en Google voor het ontwerpen van een opvolger van Scratch 2.0 namelijk Scratch Blocks.

Dit document is bedoeld om je wegwijs te maken in de mogelijkheden van het lessenpakket 'Start from Scratch!'. Lees het goed door voordat je de andere leerkrachtenhandleidingen doorneemt. In dit document leg ik het systeem waarmee ik mijn lessenpakket mee heb vormgegeven uit als ook de manier waarop je volgens mij het beste met het pakket werkt. Want, zoals je wel zal merken, 'Start from Scratch!' wijkt vrij ver af van een normaal lessenpakket of een normale cursus.

Ik zou je ten laatste nog willen bedanken om de tijd te nemen om mijn materiaal door te nemen en te gebruiken. Je begint hier waarschijnlijk aan met een nobel doeleind: kinderen en jongeren leren programmeren. Belangrijk hierbij is dat je zelf ook de onvermijdbare ontdekkingsdrang en creatieve diepgang van de leerlingen tentoonspreidt bij het opbouwen van je lessen en het opstellen van je lesmaterialen. Deze cursus kan hier een eerste aanzet van zijn.

Met dat in mijn achterhoofd zou ik je willen verzoeken om tips, opmerkingen en feedback over de cursus te verzenden naar baeyensdiede@gmail.com. Zo kan ik deze cursus blijven verbeteren en krijg ik een zicht over de bevindingen en de bedenkingen van andere onderwijzers.

Als u hier even tijd voor zou kunnen vrij maken wordt dit ten zeerste geapprecieerd en dit zal de algemene kwaliteit van het pakket 'Start from Scratch!' alleen maar verhogen.

Veel programmeer en Scratch plezier

Diede Baeyens
Student 3^{de} Bachelor in het Secundair Onderwijs
Fysica – Informatica
Thomas More Hogeschool
Mechelen

2 HET SYSTEEM

Het lessenpakket 'Start from Scratch!' is in essentie een modulaair pakket. Dit wil zeggen dat het pakket uit een aantal ruimere en uitgewerkte delen bestaat die onderling inwisselbaar, aanpasbaar en samen te voegen zijn.

Er zit dus in mindere mate een lineair idee achter de cursus en voor verschillende fasen in de ontwikkeling van de programmeervaardigheden bij de leerlingen worden er verschillende methodieken en materialen aangeboden. Dit zodat je de lessen niet enkel op maat van de leerlingen kan maken maar ook kan aanpassen aan je eigen lesstijl en je eigen voorkeuren.

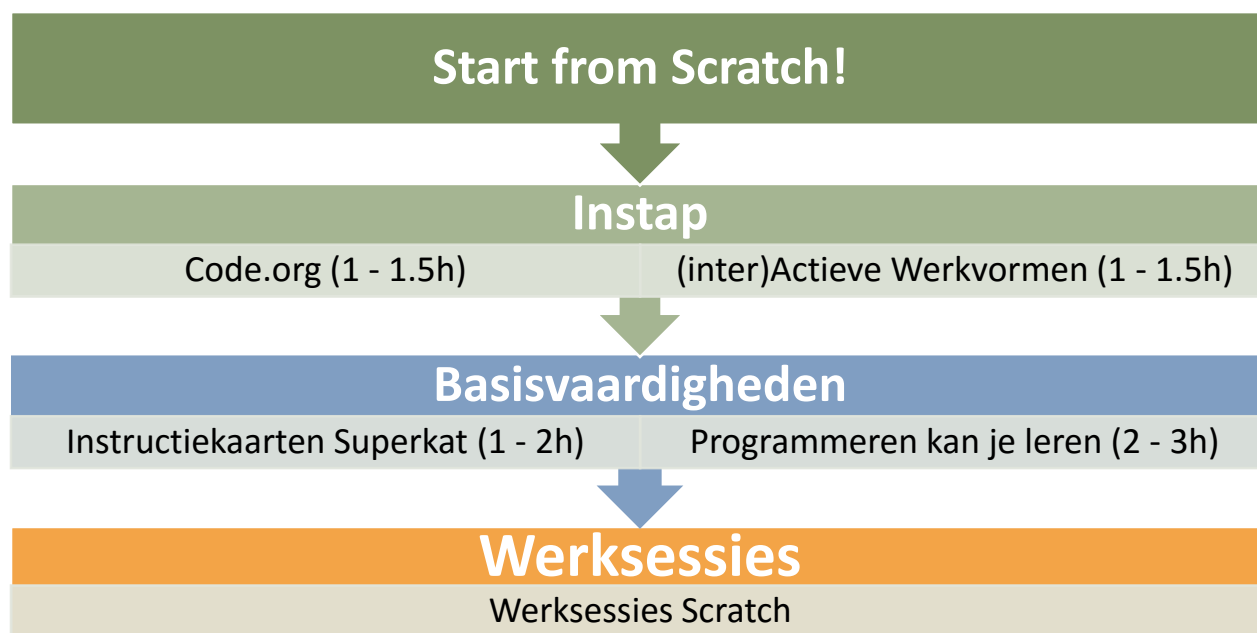
Het pakket is onderverdeeld in 3 grote fasen. De 'instap', de 'basisvaardigheden' en de 'werksessies'. Het idee achter het pakket is dat de leerlingen beginnen zonder enige voorkennis over programmeren. Eerst is het belangrijk dat de leerlingen op weg worden geholpen met het 'algoritmisch denken'. Hiervoor zijn de materialen die onder het deel 'instap' vallen voorzien.

Daarna schakelt het pakket over op Scratch en worden de basisvaardigheden die de leerlingen nodig hebben om vlot te kunnen werken met Scratch en om vlot te kunnen programmeren aangeleerd. Dit gebeurt met de materialen die onder het deel 'basisvaardigheden' vallen.

Ten laatste is het de bedoeling van het pakket dat de leerlingen zo zelfstandig mogelijk volledige Scratch projecten en oplossingen gaan kunnen uitbouwen. Om dit te bereiken wordt er veel oefenmateriaal voorzien onder het deel 'werksessies'. De leerlingen maken aan de hand van dat materiaal op verschillende verwerkingsniveaus nieuwe Scratch projecten aan en passen zo de verworven vaardigheden op het vlak van algoritmisch denken en Scratch op nieuwe probleemstellingen toe.

Om je een overzicht te geven van hoe het pakket in elkaar zit en welke onderdelen er beschikbaar zijn bij de verschillende fasen heb ik een structuurmodel gemaakt. Dit model vind je hieronder terug.

2.1 STRUCTUURMODEL START FROM SCRATCH



Zoal je kan zien gebruik ik kleurencodes voor de documentatie van de fasen. Instap documenten gaan over het algemeen een groene tint hebben, documenten die horen bij basisvaardigheden gaan over het algemeen een blauwe tint hebben en documenten die ik zelf heb gemaakt voor de werksessies gaan over het algemeen een rood-oranje tint hebben.

2.1.1 DE INSTAPFASE

De eerste fase is de instapfase. Het doel van deze fase is om de leerlingen vertrouwd te maken met het systeem en het belang van de sequentie (volgorde van de commando's) bij programmeren. Verder leren de leerlingen ook om grote problemen op te splitsen in kleinere deelproblemen die oplosbaar zijn.

Voor de instapfase voorzie ik 2 grote onderdelen. De eerste optie is om de instap te doorlopen aan de hand van Code.org. Dit is een website waarop leerlingen zelfstandig stap voor stap leren werken met een gelijkaardig blokkensysteem als Scratch. Aan de hand van Code.org en de 'hour of code' lessen krijgen de leerlingen op een uur tijd een inleiding tot visueel programmeren.

Hierdoor zijn ze goed voorbereid op de komende lessen programmeren met Scratch. De 'hour of code' lessen bouwen heel gestaag op qua moeilijkheidsgraad en zijn een perfecte instap tot programmeren voor leerlingen met verschillende beginsituaties. Dit onderdeel bestaat dus uit een les begeleid zelfstandig leren waarbij de leerlingen aan de hand van de website zelfstandig de deelproblemen proberen op te lossen.

Het alternatief voor deze methode bij de instapfase is de les met (inter)actieve werkvormen. Het doel van deze les is om de leerlingen op een alternatieve en actieve manier kennis te laten maken met algoritmisch denken en het belang van de sequentie.

Dit gebeurt door de leerlingen eerst in groepen van 4 het gezelschapsspel 'schattenjacht' te laten spelen. Dit gezelschapsspel zorgt ervoor dat de leerlingen onbewust kennis maken met het belang van de sequentie en verder ook hun ruimtelijk inzicht en oriëntatie vermogen trainen.

Na het gezelschapsspel is er een actieve werkvorm voorzien waarbij de leerlingen in groep moeten samenwerken om een probleemstelling tot een goed einde te brengen. Deze werkvorm bouwt rechtstreeks verder op het gezelschapsspel en heeft als extra doel het aantonen van het systeem bij programmeren dat ervoor zorgt dat 1 probleemstelling verschillende juiste oplosmogelijkheden biedt. Ze maken tijdens deze actieve werkvormen dus kennis met eenzelfde probleem dat op verschillende juiste manieren kan worden opgelost.

2.1.2 BASISVAARDIGHEDEN

De tweede fase gaat over de basisvaardigheden. Het doel van deze fase is om de leerlingen te begeleiden in het leren werken met Scratch. De benodigde vaardigheden en achtergrondkennis worden ingeoefend en de leerlingen leren een project uit te bouwen in de werkomgeving van Scratch. De leerlingen maken hier onder andere kennis met de andere grote onderdelen van de programmastructuur: Selectie (keuze) en Iteratie (herhalingen).

Deze fase is ook weer onderverdeeld in 2 grote onderdelen. De eerste optie is om de fase te doorlopen aan de hand van de instructiekaarten Superkat. Deze instructiekaarten zijn op maat gemaakt van de leerlingen, met screenshots en korte instructies en tips. Aan de hand van deze instructiekaarten bouwen de leerlingen het project Superkat op en leren ze de basisvaardigheden om te kunnen werken met Scratch aan.

De bedoeling van de instructiekaarten is dat de leerlingen volledig zelfstandig de basisvaardigheden inoefenen. Er wordt heel competentiegericht gewerkt waarbij het belang bij het eindresultaat ligt, namelijk een werkend programma. De leerstof aangaande Scratch en de oplosstrategieën van deelproblemen zijn hieraan ondergeschikt.

De didactische methode die je bij deze optie best toepast is het begeleid zelfstandig leren. De leerlingen kunnen volledig zelfstandig aan het werk achter de computer en je taak als leerkracht bestaat er in om hen bij te sturen, te motiveren, aan te sporen en te helpen waar nodig. Verder moet je uiteraard ook toezicht houden en de klasorde bewaren.

De tweede optie bestaat uit de invulcursus 'Programmeren kan je leren'. Deze cursus gaat veel dieper in op leerinhouden en oplosstrategieën waardoor het zich ook iets minder leent tot begeleid zelfstandig leren. De leerlingen moeten bij deze cursus zelf de meeste redeneringen doorlopen waarmee de verschillende algoritmen zijn opgebouwd.

Ze krijgen een aantal denk- en theorievragen voorgeschoteld aan het begin van elk deelprobleem. Daarna krijgen ze een overzicht van de benodigde codeblokken en moeten ze deze in de juiste volgorde proberen te zetten. Dit gebeurt echter niet rechtstreeks aan de computer maar de leerlingen doen dit door de blokken te nummeren in de cursus. Hierdoor worden ze verplicht om zelf eerst na te denken over de logische opbouw van het programma voordat ze het gaan omzetten in een werkend programma op de computer.

De leerstof en de competenties worden op die manier bewuster en met meer diepgang bij de leerlingen ingeoefend. Ook hebben ze dankzij de invulcursus een naslagwerk ter beschikking. Dit stelt hen dan op latere datum in staat om alles nog eens te bekijken.

De invulcursus leent zich echter zowel tot begeleid zelfstandig leren als tot meer klassiekere vormen van lesgeven. Om aan de hand van de invulcursus de leerlingen zelfstandig aan het werk te zetten kan je gebruik maken van de instructiekaarten om de leerlingen bij te sturen wanneer ze vast zitten. Leerlingen die veel moeite hebben met het vinden van de juiste oplossing bij een deelprobleem kunnen aan de hand van de instructiekaart van dat deel de codeblokken opbouwen in Scratch. Daarna kunnen ze reflecteren door de invulcursus in te vullen om zo de leerstof als nog eigen te maken.

Meer over de didactische keuzes bij het geven van de cursus programmeren kan je leren vind je terug in het document 'programmeren kan je leren leerkrachtenhandleiding'.

2.1.3 WERKSESSIES

De laatste fase van 'Start form Scratch!' bestaat uit een werksessieconcept. De bedoeling is dat de leerlingen op een zelfstandige wijze de verschillende competenties inoefenen en toepassen op nieuwe projecten binnen de Scratch werkomgeving.

Hiervoor wordt er een verscheidenheid aan ondersteunende materialen aangeboden. De bedoeling is dat de leerlingen zelf kunnen kiezen welke opdracht, oefening of uitdaging ze aangaan. Beperk je zeker niet tot de materialen van deze cursus, je kan zelf heel gemakkelijk extra materiaal maken of zelfs vinden via het internet.

De materialen van de cursus worden ingedeeld in 4 verschillende moeilijkheidsniveaus. Deze zullen de leerlingen helpen bij het kiezen van de meest geschikte opdracht. De indeling van de oefeningen in de niveaus is gebaseerd op de kennis en competenties die de leerlingen tijdens het deel basisvaardigheden eigen maken.

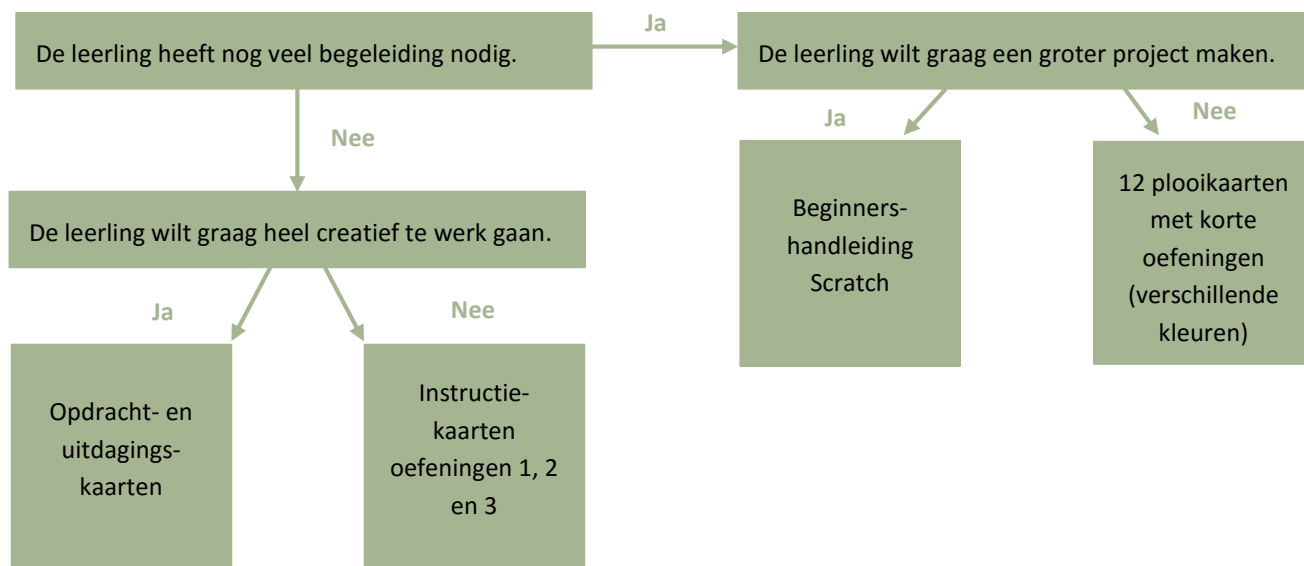
Hieronder zie je een overzicht van de verschillende moeilijkheidsniveaus en wat dit concreet betekent voor de oefeningen.

MOEILIKHEIDSNIVEAU	OMSCHRIJVING
	Eenvoudig: de vaardigheden van de voorgaande lessen worden heel eenvoudig en kort toegepast op nieuwe probleemstellingen
	Gemiddeld: de vaardigheden van de voorgaande lessen worden uitgebreid toegepast op nieuwe probleemstellingen.
	Moeilijk: er zijn nieuwe vaardigheden nodig om deze probleemstellingen op te lossen of de oplossing van de probleemstellingen is vrij uitgebreid en vergt dus een groter inspanning
	Uitdagend: de leerlingen worden bij deze oefeningen echt uitgedaagd en moeten hard werken om deze probleemstellingen tot een juiste oplossing te brengen.

De bedoeling van de werksessies is dat de leerlingen zelfstandig aan het werk gaan met Scratch. Nuttig hierbij kan zijn dat je hen eerst laat zien hoe ze via de Scratch website een programma kunnen 'remixen' om zo achter de schermen te kijken hoe het programma in elkaar zit en dit zelf aan te passen.

Het staat de leerlingen dan ook vrij om zelf een project te verzinnen en dit uit te werken in Scratch. Belangrijk is dus dat je als leerkracht de leerlingen bijstaat in de keuzes die ze maken. Help ze eerst mee beslissen zodat ze een haalbaar project aanvatten dat aansluit bij hun verwerkingsniveau. Daarna begeleid je de leerlingen in het oplossen van de probleemstellingen en het afwerken van de projecten door hen bij te sturen waar nodig.

Om je te helpen bij het aansturen van de keuzes van de leerlingen heb ik hieronder een beslissingsdiagram toegevoegd waarin de verschillende aangeboden materialen staan:



Wanneer je deze diagram volgt ga je de leerlingen ten alle tijden voorzien van een opdracht die zoveel mogelijk op maat is van zowel hun interesse als hun huidige competenties en verwerkingsniveau aangaande Scratch.

2.2 KEUZEHULPMIDDEL

Om je bij te staan in de keuze aangaande de opties bij de instap- en de basisvaardighedenfase heb ik een klein overzicht gemaakt met de pro's en contra's van de verschillende opties. De beslissing voor een bepaalde werkvorm is echter nog altijd het best geschoeid aan de hand van je eigen lesstijl en de manier waarop je wilt dat de leerlingen de leerstof verwerken. Bekijk de verschillende materialen dus goed en baseer daar je keuze op.

FASE	ONDERDEEL	PRO	CONTRA
INSTAP	Code.org	✓ gelijkaardig met de Scratch werkomgeving en dus een goede inleiding voor de komende lessen; ✓ reeds expliciet werken aan de programmeer-competenties; ✓ rustige opbouw qua moeilijkheidsgraad; ✓ doordachte leerpaden met het oog op een inleiding tot programmeren.	○ moeilijk in te passen in lesuren van 50 minuten; ○ afhankelijk van de internetverbinding; ○ leerlingen moeten weinig zelf nadenken het is vooral lezen en dan uitvoeren.
INSTAP	(inter)actieve werkvormen	✓ doorbreekt het dagelijkse stramien van het onderwijs; ✓ zonder de computer leren algoritmisch denken; ✓ werken aan dynamisch, affectieve en sociale competenties.	○ tragere methode dan Code.org aangaande programmeer-competenties aanleren; ○ klasmanagement wordt moeilijker; ○ eventueel nood aan verlaten van het klaslokaal.
BASISVAARDIGHEDEN	Instructiekaarten Superkat	✓ verwerkingssnelheid van de vaardigheden ligt hoog en de lestijd dus lager; ✓ leerlingen zijn zelf verantwoordelijk voor hun leerproces; ✓ leerlingen lossen zelfstandig de probleemstellingen op.	○ grote differentiatie op het vlak van verwerkingssnelheid van de leerlingen; ○ rendement ligt lager bij gedemotiveerde leerlingen; ○ weinig controle over het lesverloop.
BASISVAARDIGHEDEN	Programmeren kan je leren	✓ leerstof en vaardigheden worden heel diepgaand aangeleerd; ✓ leerlingen leren beter om code zelf te structureren; ✓ naslagwerk voor latere data. ✓ Veel controle over het lesverloop.	○ dezelfde competenties worden trager ingeoefend; ○ succeservaring van het maken van Superkat laat langer op zich wachten; ○ snelle leerlingen worden niet (of nauwelijks) uitgedaagd.

3 BESCHIKBARE MATERIALEN PER ONDERDEEL

FASE	MATERIAAL
INSTAP	1. algoritmisch denken met code.org leerkrachtenhandleiding; 2. algoritmisch denken (inter)Actieve werkvormen leerkrachtenhandleiding; 3. personages Code.org; 4. werken met Code.org; 5. spelbord Schattenjacht; 6. speelkaarten Schattenjacht; 7. speluitleg schattenjacht.
BASISVAARDIGHEDEN	1. programmeren kan je leren; 2. programmeren kan je leren leerkrachtenhandleiding; 3. instructiekaarten Superkat; 4. instructiekaarten Superkat leerkrachtenhandleiding; 5. oplossingenblad Kat op je plaats & De kat springt; 6. oplossingenblad De stenen bewegen & Wie stopt de stenen; 7. oplossingenblad GAME OVER; 8. werkomgeving Scratch; 9. super Kat.sb2 .
WERKSESSIES	1. begeleiden van de werksessies leerkrachtenhandleiding; 2. oefening 1 ‘appeltjes vangen’; 3. oefening 2 ‘enge vleermuizen’; 4. oefening 3 ‘help ik zit vast’; 5. 20 opdracht- en uitdagingskaarten; 6. Scratch beginnershandleiding; 7. 12 Scratch plooi kaarten met korte oefeningen 8. Oplossingen van oefening 1, 2 en 3 in .sb2 bestanden.