

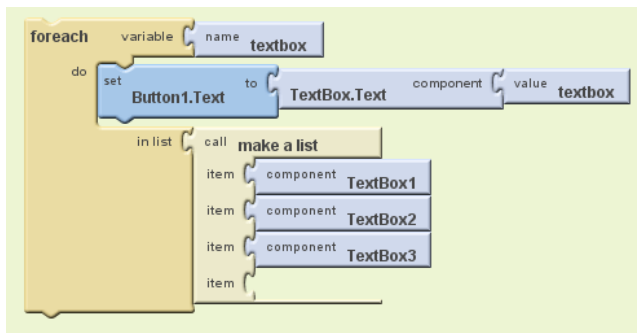
# Android Apps met App Inventor

## Docentenhandleiding

Coen Crombach

Robin Eggenkamp

François Vonk



3 april 2016



Dit werk is gelicenseerd onder een Creative Commons Naamsvermelding-NietCommercieel-GelijkDelen 3.0 Unported. Bezoek <http://creativecommons.org/licenses/by-nc-sa/3.0/> om een kopie te zien van de licentie of stuur een brief naar Creative Commons, 444 Castro Street, Suite 900, Mountain View, California, 94041, USA.

# Inhoudsopgave



# Hoofdstuk 1

## Inleiding

### Noot

Het lesmateriaal dat u hebt gekregen is nog niet af! Zaken zoals lay-out en schrijfstijl zijn bijvoorbeeld nog niet altijd consistent. Ook is het materiaal nog niet helemaal compleet. Het materiaal is volgens ons echter van voldoende kwaliteit om er een oordeel over te vormen en aan te geven of het zinvol is om het af te maken of dat we mogelijk zaken anders aan moeten pakken. Hierover willen we dan ook graag feedback. We zijn in het bijzonder geïnteresseerd in uw mening over motiverende werking en effectiviteit.

Beste docent. Dit is de docenten handleiding bij het lesmateriaal Android Apps met *App Inventor*. Het lesmateriaal is zo opgezet dat leerlingen die dit willen zelfstandig kunnen werken. In het lesmateriaal zijn opgaven verwerkt die makkelijk herkend kunnen worden door de manier waarop we

ze vorm hebben gegeven, zie figuur ???. Hierdoor kunnen meerdere lesstrategieën tegelijkertijd worden toegepast:

- Leerlingen kunnen zelfstandig de opdrachten in het materiaal zoeken en proberen die te maken door eerst zelf zo ver mogelijk te komen en op te zoeken wat ze niet weten.
- Leerlingen kunnen zelfstandig gaan lezen en opdrachten maken wanneer ze die tegenkomen.
- Leerlingen kunnen zelfstandig eerst alles doorlezen en daarna opdrachten gaan maken.
- Je kunt als docent de hele klas of die leerlingen uit de klas die daar behoefte aan hebben een aantal stappen uit het materiaal voordoen en de leerlingen het dan na laten doen.

Dit biedt de mogelijkheid om leerlingen met verschillende leerstijlen op een voor hen fijne manier aan het werk te zetten en ze daardoor beter te bedienen.

In hoofdstukken 1 (Installatie) en 2 (Ontwikkelomgeving) staan nog geen opgaven. Het is daarom raadzaam alle leerlingen die twee hoofdstukken eerst door te laten lezen.

### Opgave 3.1

Bekijk de code van de procedure ‘verplaatsMol’ en beredeneer waarom de mol zich verplaatst zoals je ziet wanneer je erop klikt.

**Figuur 1.1** Opmaak van een opgave in het materiaal

We raden aan om als docent een Google account te hebben zodat je zelf de *App Inventor* ontwikkelomgeving kunt ge-

bruiken. Hieronder de relevante links (die natuurlijk ook in het lesmateriaal staan):

- [accounts.google.com/NewAccount?hl=nl](https://accounts.google.com/NewAccount?hl=nl)
- [beta.appinventor.mit.edu](https://beta.appinventor.mit.edu)

Ook raden we aan om toegang tot een dummy leerling schoolaccount te hebben zodat je kunt testen of alles werkt voor de leerlingen. Dit voorkomt vervelende verrassingen tijdens de eerste les.





## Hoofdstuk 2

# Opzet van het materiaal

Na de uitleg van de installatie en ontwikkelomgeving beginnen we met een luchtig spel genaamd Mollen Meppen. We laten de leerlingen hier allereerst kennismaken met de emulator. We raden aan de leerlingen niet te vroeg naar een echt Android apparaat te laten gaan omdat we het gebruik van de emulator ook belangrijk vinden. Bovendien wil je in het begin dat leerlingen met de ontwikkelomgeving bezig zijn en niet met hun mobieltje. Als ze alle opgaven van Mollen Meppen hebben gemaakt (behalve de bonus opgave) is het een mooi moment om het op hun telefoon te testen.

In het lesmateriaal worden elementen uit de ontwikkelomgeving met een aparte opmaak weergegeven. Voor items uit de menu's gebruiken we tekst op een licht groene achtergrond en voor programmeerblokken (blocks) gebruiken we tekst op een licht blauwe achtergrond. Dit staat uiteraard ook uitgelegd in het lesmateriaal zelf.

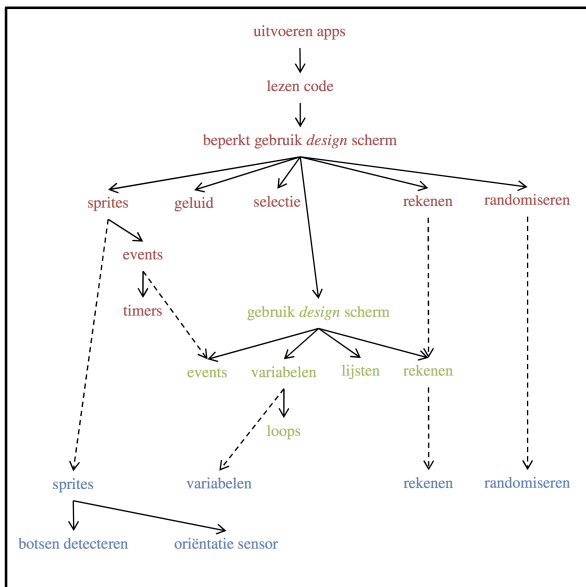
Bij het lesmateriaal worden ook projecten bijgeleverd die als startpunt voor de leerlingen dienen. Omdat scholen verschillende leeromgevingen hebben kunnen we in ons lesmateriaal geen informatie verschaffen over waar deze projecten te vinden zijn. Het handigst is het als de leerlingen dit al weten voor ze met het materiaal aan de slag gaan. Op punten waar zij iets nodig hebben geven we in het lesmateriaal aan dat ze bij de docent kunnen informeren waar het één en ander staat.

Het lesmateriaal begint vrij eenvoudig en wordt gaandeweg moeilijker. We zetten eerst de basis in de steigers (scaffolding) en bouwen daar vervolgens op verder. Ook worden belangrijke concepten herhaald om ze goed te verankeren. Ieder hoofdstuk heeft verder één of meerdere bonus opgaven zodat leerlingen die snel door het lesmateriaal gaan extra uitdagingen krijgen.

In figuur ?? staat een schema dat aangeeft welke *App Inventor* - en programmeerconcepten aan bod komen in de diverse onderdelen van het lesmateriaal. De verschillende fontkleuren in het figuur geven de volgorde in het lesmateriaal aan. Concepten in rode tekst behoren bij Mollen Meppen, groene tekst bij Schrandere Scholier en blauwe tekst bij Meteo. De doorgetrokken pijlen geven afhankelijkheidsrelaties aan. Zo hebben we ervoor gekozen een leerling eerst code te laten lezen voordat hij/zij met het design scherm aan de slag gaat. Voordat een leerling echter zelf gaat programmeren moet deze eerst een beperkte kennis van het design scherm hebben. De onderbroken pijlen geven herhaling weer en staan alleen tussen concepten met gelijke namen. Concepten die op dezelfde hoogte staan in het figuur hebben geen relatie of afhankelijkheid ten opzichte van elkaar.

Een aantal concepten in figuur ?? verdienen enige toelichting:

- Met design scherm bedoelen we gebruik van menu's en componenten.
- Met selectie bedoelen we het conditioneel uitvoeren van stukken code met bijvoorbeeld een if-statement.
- Met events bedoelen we gebeurtenissen gegenereerd door standaard componenten, zoals het indrukken van een knop.
- Met sprites worden bewegende objecten bedoeld.
- We hebben botsen detecteren gebruikt als vertaling van Collision Detection.



**Figuur 2.1** Scaffolding en herhaling

In het lesmateriaal is niet specifiek rekening gehouden met vrouwelijke leerlingen of allochtone leerlingen. De onder-

werpen zijn naar onze mening redelijk neutraal van aard op het gebied van deze vormen van diversiteit.

Op het moment schatten we de benodigde tijd per onderwerp als volgt in:

- Introductie en Mollen Meppen 4 uren
- Schrandere Scholier 6 uren
- Meteor 6 uren
- Onderwerp waarin we procedures met parameters gebruiken en liefst variabelen, selectie en loops herhalen 6 uren (*nog te ontwikkelen*)
- Vrije individuele opdracht 10 uren (*nog te ontwikkelen*)
- Vrije groepsopdracht 15 uren (*nog te ontwikkelen*)

Dit is in totaal 47 uur wat uitkomt op bijna 16 weken bij 3 uur les per week. Vrij vertaald zou dit moeten passen in 2 perioden van ongeveer 8 weken.

## Hoofdstuk 3

# Achtergrond bij keuzes gemaakt in het materiaal

Ons doel met het lesmateriaal was het maken van een les-serie die leerlingen motiveert om te leren programmeren en die dit vervolgens op een effectieve manier doet.

Het gekozen platform is ons grootste motiverende aspect. Echter het niet hoeven omgaan met vaak lastig te interpreteren compiler of interpreter meldingen omdat we voor een grafische ontwikkelomgeving hebben gekozen zal voorkomen dat leerlingen gedemotiveerd raken.

Voor startende programmeurs is een grafische ontwikkelomgeving een effectieve manier voor het leren van programmeerconcepten om de volgende redenen:

- Syntax van tekstuele programmeertalen leidt af van

de programmeerconcepten.

- Lastig te interpreteren meldingen van compilers en interpreters kosten vaak veel tijd die niet besteed kan worden aan het leren programmeren.
- Grafische bouwblokken die in elkaar klikken geven direct feedback over de semantiek van de bouwblokken zodat leerlingen niet steeds hoeven te zoeken naar de regels zoals bij tekstuele programmeertalen.

Onder programmeerconcepten verstaan we dingen als variabelen, selectief code uitvoeren, herhaald code uitvoeren (loops), events, testen, debuggen enzovoorts. Als leerlingen eenmaal bekend zijn met de programmeerconcepten kunnen ze veel makkelijker en sneller overstappen naar een tekstuele ontwikkelomgeving omdat ze ook moeten ervaren hoe deze werken. Met name testen en debuggen is heel anders in dit type omgevingen.

Zeker in het begin bij Mollen Meppen hebben we ervoor gekozen om de leerling steeds iets kleins te laten wijzigen zodat er voldoende succesmomenten zijn. Dit zal bijdragen aan de motivatie van leerlingen om door te gaan.

Als eerste ervaring met programmeren hebben we er voor gekozen om de leerlingen al iets te geven dat werkt maar nog verre van perfect is. Dit is een goed beginpunt om voor het eerst een app uit te voeren in de ontwikkelomgeving. De opzet is dusdanig eenvoudig dat leerlingen het moeten kunnen begrijpen. Daarna mogen de leerlingen zelf dingen in de code veranderen en er dingen bij maken. Dit sluit aan bij de 'cognitieve doelen strategie' van Jens Kaasbøll (Kaasbøll, 1998). In deze strategie onderkent Kaasbøll 4 stappen:

1. Voer een programma uit

2. Lees de code van een programma
3. Breng veranderingen aan in een programma
4. Schrijf zelf een programma

Onderdeel van deze strategie is tevens het ‘programmeren door completeren’ zoals beschreven door Merriënboer (Merriënboer, 1992). Dit wordt doorgaans ook als een goede didactische aanpak gezien bij het beginnen met leren programmeren.

Verder hebben we gepoogd zoveel mogelijk van de zogenaamde Big Ideas van programmeren (Saeli, 2012) aan bod te laten komen omdat Informatica docenten deze belangrijk vinden binnen het programmeeronderwijs:

- Controle structuren met de focus op loops
  - In Schrandere Scholier bij het doorlopen van een lijst
- Data structuren
  - De ontwikkelomgeving kent getallen, booleans en tekst als data structuren maar dit is een zeer klein begin en booleans zijn zeer impliciet
  - In Schrandere Scholier wordt de datastructuur lijst gebruikt
- Arrays
  - Zitten niet in de ontwikkelomgeving en komen daarom niet als zodanig aan bod
  - In Schrandere Scholier worden wel lijsten gebruikt die feitelijk een meer algemene vorm van een array zijn
- Probleem oplossende vaardigheden
  - Worden aangesproken in de opgaven in het lesmateriaal

- Decompositie
  - In lichte mate in Mollen Meppen door met een procedure te werken in de versie die de leerlingen krijgen maar er wordt hier niet direct op gestuurd
  - In Schrandere Scholier moeten leerlingen zelf een procedure maken
- Parameters
  - In Schrandere Scholier worden deze genoemd maar de leerlingen hoeven er nog niet mee aan de slag
- Algoritmes
  - In lichte mate in Mollen Meppen door het laten tellen van missers en het inbouwen van een score
  - In lichte mate in Schrandere Scholier door het laten berekenen welk cijfer er gehaald moet worden voor een 8 gemiddeld
  - In Metoor bij het werken met de hellingshoek van het mobiele apparaat



## Hoofdstuk 4

### Tot slot

Het materiaal zoals het er nu ligt is wat ons betreft nog niet af. Er staan nog een aantal onderwerpen in de rij. Na het doorlopen van de onderwerpen raden we aan om leerlingen eerst individueel een App te laten bedenken en maken. Hiermee kun je toetsen of alle leerlingen iets hebben opgestoken van de programmeerconcepten. Vervolgens is het misschien leuk om leerlingen in groepjes een wat grotere App te laten bedenken en maken. Hierbij is het wellicht leuk om leerlingen educatieve Apps te laten maken die aansluiten bij één van de andere vakken die ze hebben zoals biologie, geschiedenis, aardrijkskunde enzovoorts.



# Hoofdstuk 5

## Referenties

Kaasbøll, J.J. (1998). Exploring didactic models for programming. Tapir, pp. 195-203.

Merriënboer, J.J.G. van (1992). Programmeren door completeren. TINFON, 1(2), pp. 66-71.

Saeli, M. (2012, February). Teaching Programming for Secondary School: a Pedagogical Content Knowledge Based Approach. Technische Universiteit Eindhoven.

