

Introductie

Apps maken met App Inventor

ICT & Innovatie

Deze lessenserie is gemaakt als opdracht voor de module *ICT & Innovatie* in leerjaar 3 van de opleiding tot Docent ICT aan de Lerarenopleiding Technisch Beroepsonderwijs van Christelijke Hogeschool Windesheim in Zwolle.

Het doel van de lessen is een korte kennismaking met de wereld van het programmeren en een eerste verkenning van het programmeren van software voor een besturingssysteem.

Om deze eerste stappen laagdrempelig te houden, en daardoor geschikt te maken voor kinderen vanaf ongeveer 10 jaar, is gekozen voor App Inventor. Dit is een [website](#) waar met behulp van kant-en-klare blokken een app geprogrammeerd kan worden voor Android.

Inhoud

Artikel 'Apps programmeren met App Inventor' voor Vives Magazine

Docentenhandleiding

Les 1: De basis van programmeren

Les 2: Designer mode

Les 3: Block mode

Les 4: Camera integratie

Amsterdam, 3 januari 2016

Berry Nieskens

berry@nieskens.nl

Lerarenopleiding Technisch Beroepsonderwijs

Christelijk Hogeschool Windesheim

Dit document is als pdf te downloaden op <http://informatics.nu/appinventor>.

Apps programmeren met App Inventor

De voortdurende digitalisering van onze maatschappij blijft ons geweldige kansen bieden. Wil je hier optimaal gebruik van maken, dan vraagt dat om buitengewone kennis en kunde van computertechnologie. Hoe bijzonder zou het zijn als kinderen zich al op jonge leeftijd hiervan bewust worden en een gevoel van empowerment opdoen door zelf apps te programmeren voor hun telefoon of tablet?



Met behulp van MIT's App Inventor wordt het maken van mobiele apps letterlijk kinderspel. De software is gratis beschikbaar en er zijn talloze tutorials te vinden die je op weg kunnen helpen bij het programmeren van je eerste app.



De software is opgedeeld in twee onderdelen met elk hun specifieke functie: Designer Mode en Block Mode. In de ontwerpmodus plaats je onder meer buttons, tekstlabels en afbeeldingen uit het pallet op het canvas.

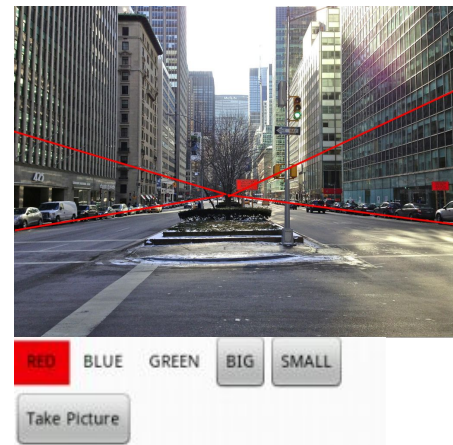
Hiermee krijg je een goed beeld van hoe jouw app er uiteindelijk uit zal komen te zien. In de Block Mode ga je vervolgens met het programmeerwerk aan de slag. Dit gebeurt niet volledig tekstueel zoals bij volwassen programmeertalen maar door het slepen van vooraf gedefinieerde blokken. Super simpel, en ontzettend leuk en leerzaam!

Vakoverstijgend

Het maken van apps is een hele leerzame ervaring. En je kan er vervolgens ook nog andere leerzame dingen mee doen. Ga je voor Beeldende Vorming een straatbeeld natekenen? Met App Inventor maak je in een handomdraai een app waarmee je perspectieflijnen op een zelfgemaakte foto kan tekenen.

Code.org

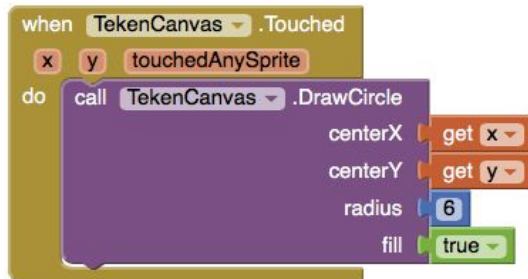
Ben je een 'absolute beginner' op het gebied van programmeren dan is www.code.org een goede tussenstap op weg naar App Inventor. Daar leer je structuren zoals *sequences*, *loops* en *conditionals*. Lukt het je om alle puzzels op te lossen, dan schuilt er een echte programmeur in jou.



Programmeren met blokken

Zowel de Hour of Code cursus als App Inventor zelf leunen grotendeels op het programmeren in zogenaamde 'block editors'. In plaats van tekstueel alle code te moeten typen, kan een leerling de computer voorgeprogrammeerde opdrachten laten uitvoeren door

het slepen van blokken naar het werkveld. Hierdoor wordt het kennismaken met programmeren laagdrempeliger. De leerling loopt minder snel tegen foutmeldingen aan en kan al zijn creativiteit in het programma stoppen, ongehinderd door gebrek aan kennis van syntax.

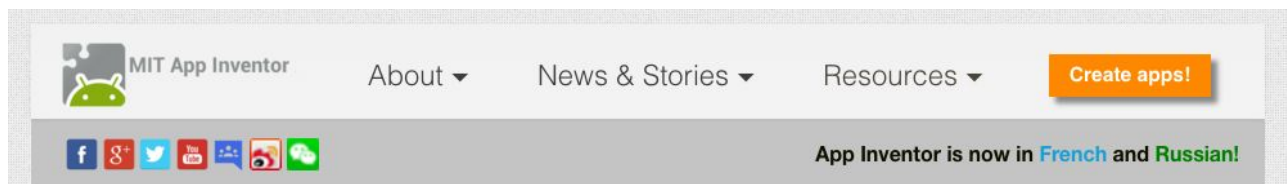


Een van de vaakst voorkomende constructies in een applicatie is ALS-DAN. Als er op een knop wordt gedrukt, dan moeten de lijnen op het scherm gewist worden. Of als er op het scherm wordt gedrukt, dan moet er een stip getekend worden op die positie. App Inventor maakt het programmeren van zo'n voorwaardelijke uitvoer heel eenvoudig. Elke knop die je aan de layout toevoegt, heeft in de Block Mode een lade met

daarin kant-en-klare blokken.

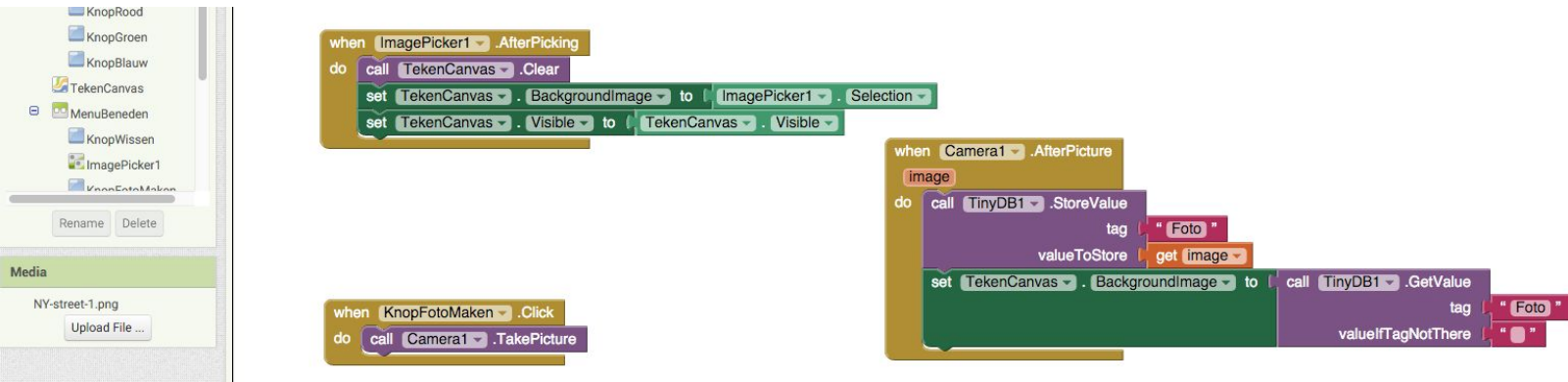
Aan de slag met App Inventor

Op de [website van MIT](#) vind je de software en alle resources die je nodig hebt om te starten. Neem ook vooral een kijkje bij de [Beginner Tutorials!](#)



Docentenhandleiding

Apps maken met App Inventor



Tijdsduur

Deze cyclus bestaat uit vier lessen van minimaal 45 minuten.

Opdracht

Het eindresultaat is een mobiele applicatie voor Android waarin een zelfgemaakte foto als achtergrond van een tekencanvas geplaatst kan worden om er perspectieflijnen op te kunnen tekenen.

Achtergrond

In deze lessenserie wordt aangetoond dat het kunnen programmeren niet alleen een doel op zich is. Het heeft een meerwaarde wanneer het gebruikt kan worden om problemen in de werkelijkheid op te lossen of om het leven gewoon een stuk aangenamer te maken. Programmeren kan zeer nuttig zijn voor andere vakgebieden dan informatica, bijvoorbeeld Beeldende Vorming. Om dit aan te tonen maakt de leerling een applicatie waarmee hij perspectieflijnen kan tekenen op een zelfgemaakte foto alvorens hij het straatbeeld zelf gaat tekenen.

De leerling heeft nodig

Een computer met een browser en internetverbinding. Een Android telefoon of tablet is niet noodzakelijk maar wel wenselijk.

Indeling van de activiteiten en doelen

Les 1 Hour of Code: De basis van programmeren

Doel: De leerling leert drie structuren die bij het programmeren van computers vaak worden gebruikt: opeenvolging, herhaling en voorwaardelijke uitvoer.

Activiteit: De leerling volgt de 'Hour of Code' cursus van Code.org. Door puzzels op te lossen maakt hij kennis met onder meer herhalingen en conditionele uitvoer. Vooral de ALS-DAN structuur is belangrijk om te begrijpen met het oog op het programmeren van een applicatie.

Les 2 App Inventor: Designer Mode

Doel: De leerling maakt kennis met de webapplicatie App Inventor. Hij leert hoe in de Designer Mode een layout ontworpen kan worden voor een mobiele applicatie. Hij leert dat die wordt samengesteld uit componenten en dat elk component verschillende eigenschappen heeft die aangepast kunnen worden.

Activiteit: De leerling plaatst en structureert componenten zoals buttons (knoppen), een canvas en een layout arrangement in de applicatie. Hij bepaalt het uiterlijk van deze componenten door de eigenschappen ervan aan te passen. Hieronder vallen weergave van tekstlabels, kleur en grootte van het component, en diens plaats in het beeld.

Les 3 App Inventor: Blocks Mode

Doel: De leerling leert werken in de Blocks Mode van App Inventor. In deze modus wordt de code geprogrammeerd om te kunnen tekenen op het canvas.

Activiteit: De leerling plaatst blokken die voor de componenten beschikbaar zijn in het werkveld en past waar nodig parameters aan. Er worden zogenaamde 'event handlers' aan de buttons gekoppeld die bepalen welke acties er worden uitgevoerd bij een druk op zo'n knop.

De leerling programmeert de verwerking van interactie met het scherm. Een druk op het scherm of het slepen wordt gekoppeld aan, respectievelijk, het zetten van een stip en het tekenen van een lijn op het canvas.

De leerling gaat de code testen met behulp van een Android apparaat of de ingebouwde emulator.

Les 4 App Inventor: Camera integratie

Doel: De leerling leert hoe hij het maken van een foto en het kiezen van een foto uit een bestaand album kan integreren in de applicatie, en maakt hierbij gebruik van een database die voorziet in de opslag van de gemaakte foto.

Activiteit: In de Designer Mode worden twee extra buttons toegevoegd voor het kiezen en maken van een foto die als achtergrond van het canvas zal worden gebruikt. Vervolgens worden de event handlers geprogrammeerd die ervoor zorgen dat de juiste code wordt uitgevoerd bij een klik op een van die knoppen. Bij het maken van een nieuwe foto wordt gebruik gemaakt van een database. De camera-app slaat de gemaakte foto eerst op in die database. Vervolgens wordt die foto weer opgehaald uit de database om als achtergrond van het canvas ingesteld te worden.

Tenslotte gaat de leerling de voltooide app exporteren zodat deze op elk Androidtoestel kan worden geïnstalleerd.

Vervolg

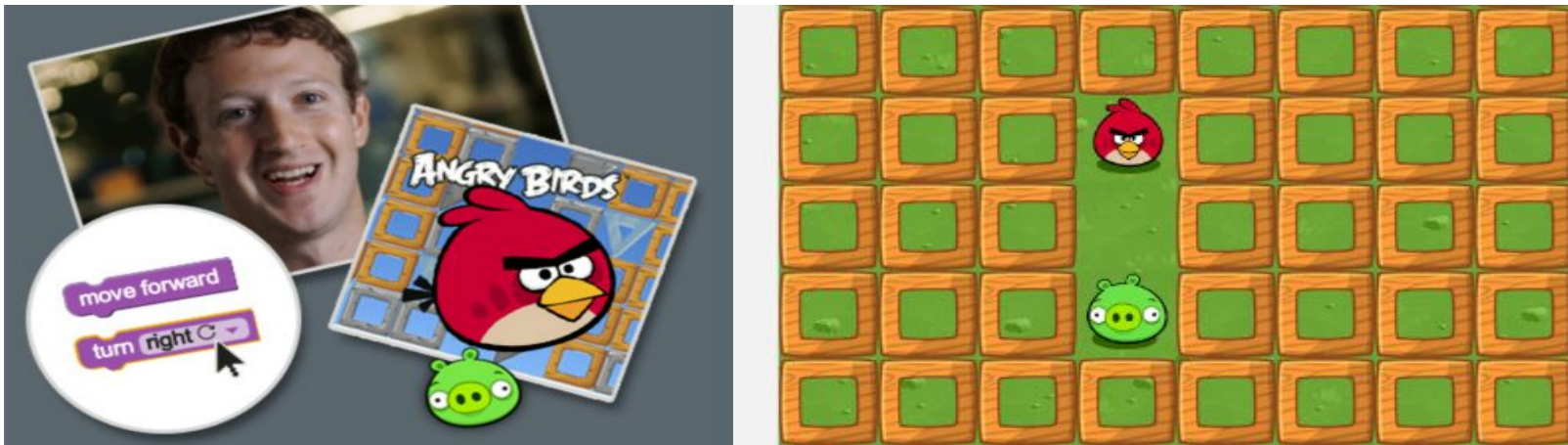
Dit is slechts één van de vele mogelijkheden die App Inventor, en programmeren in het algemeen, te bieden heeft. Er zijn talloze tutorials beschikbaar voor leerlingen die zich verder willen verdiepen in het programmeren met App Inventor. Een goede start hiervoor is [deze website](#), maar een goede zoekopdracht levert ook een heleboel inspiratie op!

App Inventor heeft een speciale [pagina voor docenten](#). Hier is informatie te vinden over het gebruiken van App Inventor in het onderwijs.

Deze tutorial is gebaseerd op <http://appinventor.mit.edu/explore/ai2/paintpot-part1.html>.

Les 1: De basis van programmeren

Sequence, loop & conditional met Code.org



Intro

[What most schools don't teach](#)

Doelen

In deze les leer je drie structuren die bij het programmeren van computers worden gebruikt: opeenvolging, herhaling en voorwaardelijke uitvoer.

Tijdsduur

Deze les duurt ongeveer 45 minuten.

Je hebt nodig

Een computer met een browser en internetverbinding.

Opdracht in het kort

Je volgt de cursus *Hour of Code* van Code.org. Hierin maak je kennis met drie programmeerstructuren: opeenvolging, herhaling en voorwaardelijke uitvoer.

Aan de slag

Ga naar <https://studio.code.org/hoc/1>.

Bekijk de filmpjes en voer de opdrachten uit.

Je krijgt hieronder extra informatie over de puzzels die je moet oplossen.

Level 1 t/m 5 Opeenvolging (Sequence)

In deze levels breng je de Angry Bird naar het varken door blokken onder elkaar te plaatsen. Deze door jou geprogrammeerde commando's worden achter elkaar uitgevoerd als je op *Run* klikt, van boven naar beneden. Deze manier van programmeren noem je **opeenvolging**.



Level 6 t/m 9 Herhaal ... keer (For loop)

Je hebt vast gemerkt dat je sommige commando's meerdere keren achter elkaar moest plaatsen. Dat soort werk is voor mensen al snel vervelend. Gelukkig zijn computers juist heel erg goed in het herhaald uitvoeren van taken. In deze levels gebruik je **herhalingen** om de computer duidelijk te maken dat een opdracht meerdere keren moet worden uitgevoerd.



In level 9 moet je de grijze blok gebruiken die klaar staat. Kijk eens goed naar de weg die de Angry Bird moet afleggen. Herken je een patroon in de stappen die hij moet zetten? Je kan een verzameling commando's natuurlijk ook laten herhalen.

Level 10 t/m 13 Herhaal totdat ... (While loop)

Wanneer je vooraf niet precies weet hoe vaak jouw commando's herhaald moeten worden, kan je ook een *totdat-lus* gebruiken. Je zegt dan: "**Herhaal** deze stappen **totdat** je bij het varken bent".



Level 14 t/m 20 Voorwaardelijke uitvoer (Conditional)

Je kan een computer ook keuzes laten maken. Hij is hier zelfs heel erg goed in, maar je moet hem wel een beetje op weg helpen. Je kan hem een keuze laten maken op basis van een **voorwaarde**, bijvoorbeeld '**Als** er een pad vooruit is, **dan** ga je rechtdoor'.



Als aan de voorwaarde wordt voldaan, voert de computer jouw commando uit. Maar anders? Wat moet hij doen als dat niet het geval is? Dat kun je hem doorgeven met behulp van een **als-anders** blok. Deze is vergelijkbaar met de als-dan-blok. Onderaan voeg je als extraatje het commando toe dat hoort bij 'anders'.

Gefeliciteerd!



Je hebt je eerste programma's met veel succes geschreven. Je ben hard op weg om een goede programmeur te worden!

In de volgende les ga je kennismaken met App Inventor. Dat is een programma waarmee je apps kan programmeren voor je Android telefoon of tablet. Je zal zien dat 'als-dan'-commando's heel erg belangrijk zijn bij het maken van apps. Ga maar eens na:

Als je met je vinger het icoontje aanraakt, dan start het programma.

Als je op het icoontje met de camera klikt, dan wordt er een foto gemaakt.

Als...

Extraatje!

Wil je meer van dit soort puzzles oplossen, dan is er ook een [uitgebreidere cursus](#) waarin je onder meer leert hoe je het maken van tekeningen kan programmeren met blokken. Je leert er zelfs over andere belangrijke programmeerconcepten zoals variabelen en functies!

Evaluatie

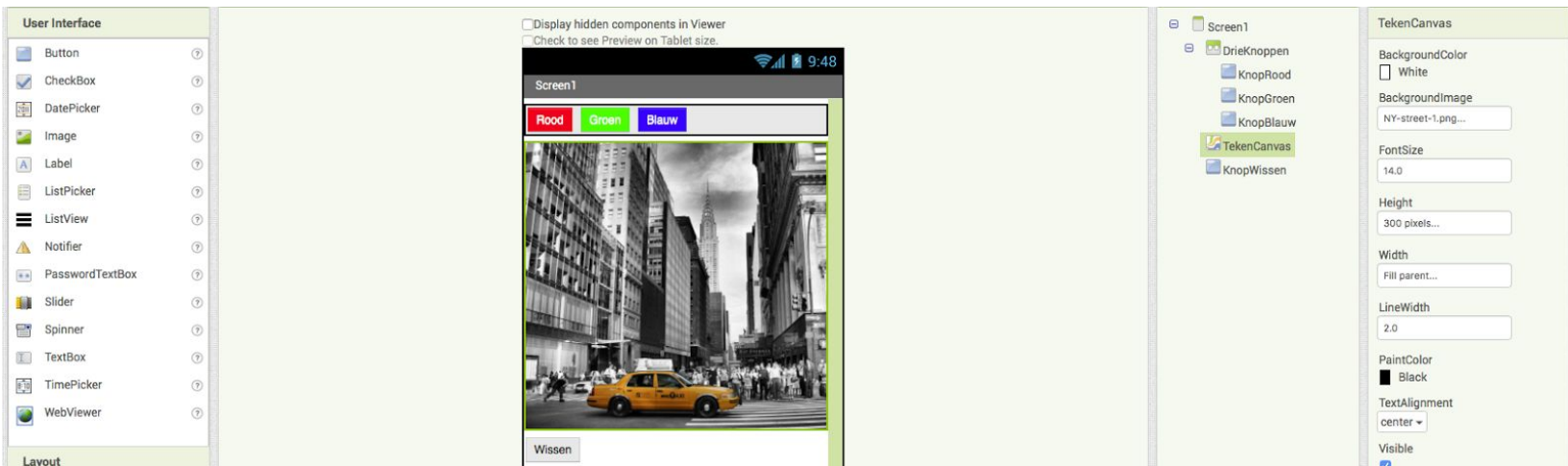
Wat ging goed?

Waar had je hulp bij nodig?

Hoe zou deze les verbeterd kunnen worden?

Les 2: Designer Mode

Layout van apps ontwerpen met App Inventor



Doelen

In deze les leer je hoe je in de Designer Mode van App Inventor de user interface van een app ontwerpt.

Tijdsduur

Deze les duurt ongeveer 45 minuten.

Je hebt nodig

Een computer met een browser en internetverbinding. Een Android telefoon of tablet is niet noodzakelijk maar wel wenselijk.

Opdracht in het kort

Je ontwerpt de user interface van je app. Dit is alles wat de uiteindelijke gebruiker ervan krijgt te zien: De knoppen, de afbeeldingen, de tekst, enzovoort.

Perspectief

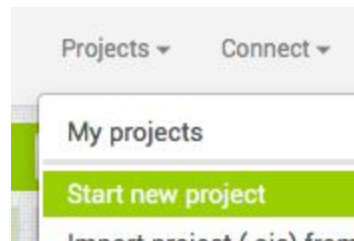
Wanneer je een app gaat maken, zijn de mogelijkheden natuurlijk eindeloos. Om App Inventor te leren kennen en gebruiken, is het handig om eerst een voorbeeld na te bouwen.

Je gaat een app maken die het mogelijk maakt om in drie verschillende kleuren te tekenen op een foto die je zelf hebt gemaakt met de camera van het toestel. We noemen de app

'Perspectief'.

In deze les maak je de layout oftewel de voorkant van die app in Designer Mode. In de volgende les programmeer je de achterkant in de Blocks Mode. Dat is het gedeelte waar de code wordt geplaatst met behulp van blokken.

Ga naar <http://appinventor.mit.edu>. Kies rechtsbovenaan in het menu voor de optie Create apps! Je wordt naar een pagina geleid waar gevraagd wordt in te loggen met een Google Account.

Mocht er niet vanzelf een nieuw project aangemaakt worden, dan kun je dat zelf alsnog doen met de menu-optie Projects – Start new project. Geef je project de naam 'Perspectief' en klik op OK.

scherm.

Je bevindt je vervolgens in de Designer Mode. Hierin ga je de opmaak van de app bepalen. Je kiest elementen uit het Palette aan de linkerkant en sleept die naar de Viewer in het midden van het

Buttons

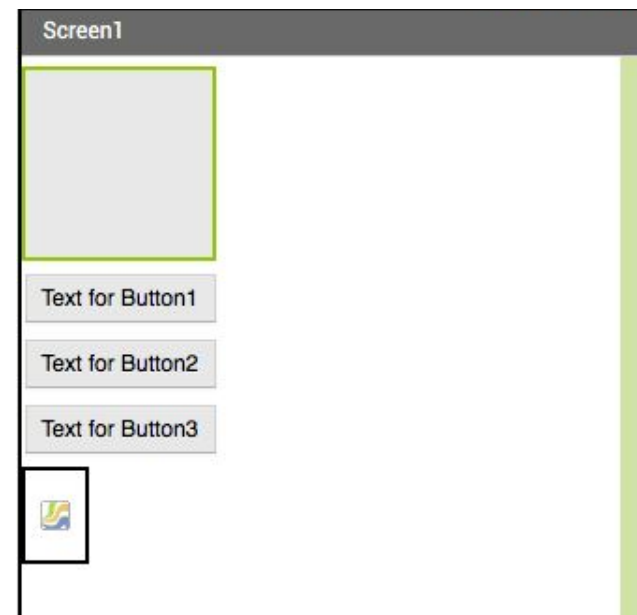
Om straks de kleur te kunnen kiezen waarmee je tekent, zijn er drie Buttons nodig. Sleep daarom drie keer een Button vanuit het Palette naar de Viewer.

Canvas

Tekenen doe je op het Canvas. In het Palette vind je in de categorie Drawing and Animation een Canvas. Sleep dit component naar de Viewer.

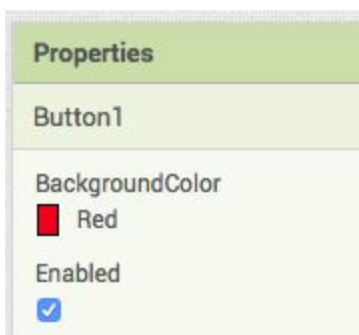
Arrangement

Om de buttons horizontaal uit te lijnen gebruik je uit de categorie Layout het component HorizontalArrangement. Sleep dit naar de Viewer en plaats hem boven de buttons.



Opmaken van componenten

Rechts van de Viewer zie je een overzicht van alle Components die je hebt toegevoegd. Wanneer je in dit overzicht of in de Viewer op een component klikt, verschijnen de eigenschappen ervan in het Properties overzicht aan de rechterkant van het scherm.



Verander de 'property' BackgroundColor van de eerste Button naar 'Red' en verander de eigenschap Text (bijna onderaan) naar "Rood".

Vervolgens maak je de achtergrondkleur van de tweede en derde

Button groen en blauw en zorg je ervoor dat de tekst op de Button daarmee overeenkomt. Om de tekst beter leesbaar te maken, kun je de eigenschap TextColor aanpassen.

In het overzicht van de components klik je op Button1 en vervolgens onderaan op 'Rename' om de naam van de button te veranderen naar 'KnopRood'. Doe dit ook voor de andere twee knoppen, 'KnopGroen' en 'KnopBlauw'.

Horizontaal menu

Je gaat de buttons nu naast elkaar plaatsen met behulp van het component HorizontalArrangement. Klik eerst op dat component en verander de naam naar 'DrieKnoppen' zoals je dat ook voor de buttons hebt gedaan. Verander ook de eigenschap 'Width' naar 'Fill parent' zodat de drie knoppen over de gehele breedte van het scherm uitgelijnd worden. Sleep tenslotte de drie Buttons in het component.

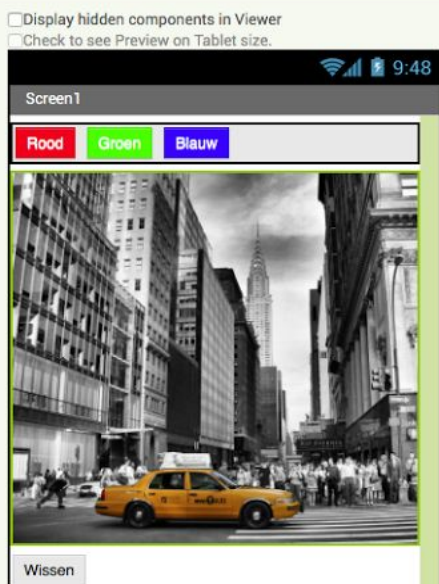


Canvas

Selecteer Canvas1 in de Viewer of Components en verander de naam naar 'TekenCanvas'. Zet ook van dit component de eigenschap 'width' op 'Fill parent'. Zorg er vervolgens ook voor dat de hoogte ervan '300 pixels' wordt. Zoek op Internet een afbeelding van een straatbeeld en download die eerst naar je computer. Vervolgens kun je deze toevoegen als 'BackgroundImage' van 'TekenCanvas'.

Wissen

Om na het tekenen op het canvas alles in één keer te kunnen wissen, voeg je onderaan in de Viewer een vierde Button toe en die noem je 'KnopWissen'. De tekst op de button wordt natuurlijk 'Wissen'.



Components

- Screen1
 - DrieKnoppen
 - KnopRood
 - KnopGroen
 - KnopBlauw
 - TekenCanvas
 - KnopWissen

Properties

TekenCanvas

BackgroundColor

White

BackgroundImage

NY-street-1.png...

FontSize

14.0

Height

300 pixels...

Width

Fill parent...

LineWidth

2.0

PaintColor

Black

TextAlignment

center ▾

Visible

Als het goed is, komt bovenstaande afbeelding overeen met wat je tot nu toe hebt gemaakt

Afronding

Je hebt in deze les geleerd dat de opmaak van een app bestaat uit componenten. Een button en een canvas zijn voorbeelden van zulke componenten. Je hebt ook gezien dat zo'n component bepaalde eigenschappen heeft die je kan aanpassen, in het Engels noem je dat 'properties'.

In de volgende les ga je in de Block Mode werken. Je gaat leren hoe je de buttons actief maakt zodat ze bepaalde handelingen kunnen laten uitvoeren.

Evaluatie

Wat ging goed?

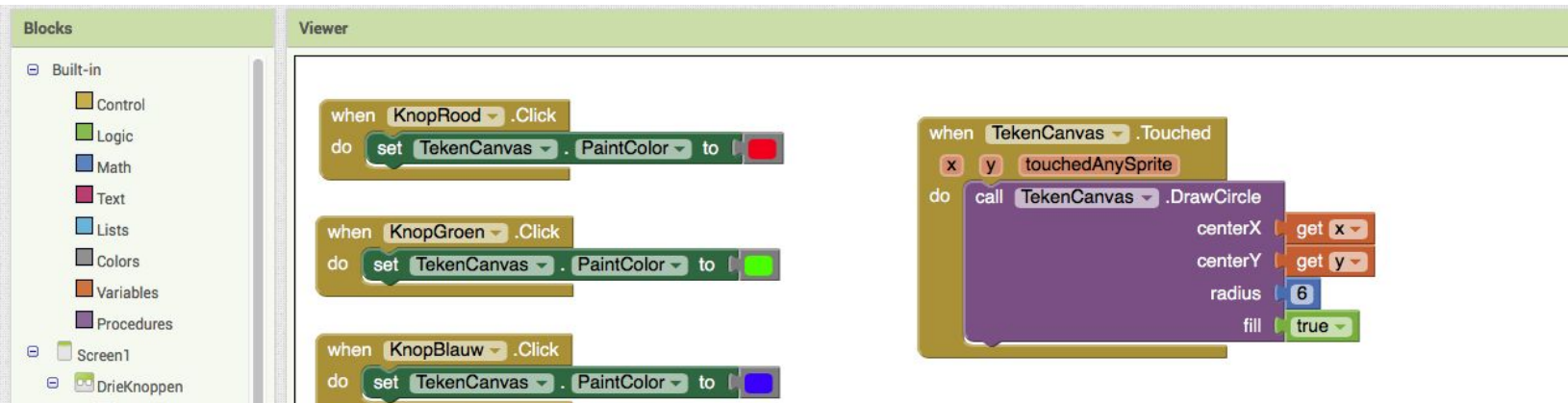
Waar had je hulp bij nodig?

Hoe zou deze les verbeterd kunnen worden?

Deze tutorial is gebaseerd op <http://appinventor.mit.edu/explore/ai2/paintpot-part1.html>.

Les 3: Blocks Mode

Componenten programmeren met App Inventor



Doelen

In deze les leer je hoe je in de Blocks Mode van App Inventor componenten programmeert tot het uitvoeren van acties.

Tijdsduur

Deze les duurt ongeveer 45 minuten.

Je hebt nodig

Een computer met een browser en internetverbinding. Een Android telefoon of tablet is niet noodzakelijk maar wel wenselijk.

Opdracht in het kort

Na het ontwerpen van de user interface van de app, ga je nu de componenten programmeren tot het uitvoeren van acties. Dit doe je met behulp van kant-en-klare blokken die relatief gezien weinig aanpassing nodig hebben om de gewenste acties te kunnen uitvoeren.

Perspectief

Ga naar <http://appinventor.mit.edu>. Kies rechtsbovenaan in het menu voor de optie 'Create apps!'. Je wordt naar een pagina geleid waar gevraagd wordt in te loggen met een Google Account. Open via de menuknoppen Projects – My projects vervolgens het project 'Perspectief' van de vorige les.

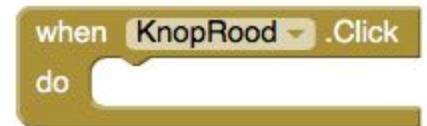
Rechtsbovenaan het scherm kies je naast Designer voor Blocks. Zo kom je in de Blocks Editor van je app.

1. Knoppen

Als

Met een klik op 'KnopRood' aan de linkerkant van het scherm open je een soort lade met daarin verschillende blokken. Kies vervolgens voor 'when KnopRood.Click do'.

In de eerste les heb je geleerd wat conditionals zijn. Met deze blok vertel je de app: "Als er op de rode knop wordt geklikt, dan moet deze actie uitgevoerd worden".



Dan

Wat je met een druk op de rode knop wil bereiken is dat voor het TekenCanvas de kleur waarmee wordt getekend verandert naar rood.

Omdat je een eigenschap van het TekenCanvas gaat veranderen, zul je ook in de lade van dat component op zoek moeten gaan naar 'set TekenCanvas.PaintColor to'. Sleep dit blok in de conditional van KnopRood.



Kleur

In het linker gedeelte vind je onder de categorie Built-in de verzameling Colors. Kies voor de kleur rood en maak die blok vast aan de blok 'set TekenCanvas.Paintcolor to'.

Hiermee heb je de actie voor KnopRood geprogrammeerd. Natuurlijk moet je vergelijkbare handelingen uitvoeren voor KnopGroen en KnopBlauw. Voer die ook uit voordat je verder gaat met de volgende stappen.



Wissen

Met een druk op de KnopWissen moet het TekenCanvas weer leeg worden gemaakt. Het commando dat daarvoor nodig is heet 'call TekenCanvas.Clear'. Dit commando zet je net zoals de kleurknoppen in een conditional, maar dan voor KnopWissen.

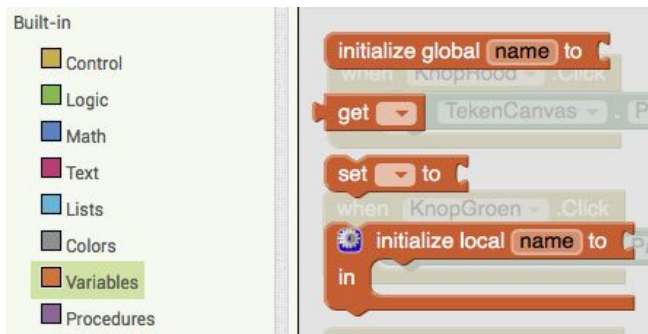
2. Teken

Je gaat twee verschillende manieren van tekenen programmeren: *touch* (raken) en *drag* (slepen). Wanneer je met je vinger het TekenCanvas aanraakt, verschijnt er een stip. Wanneer je met je vinger over het scherm sleept, wordt er een lijn getrokken. Voor elk van deze handelingen wordt er weer een aparte conditional geprogrammeerd met behulp van blokken uit

de lades.

Touch

Open de lade van het TekenCanvas en sleep de blok 'when.TekenCanvas.Touched' naar het werkveld. Deze blok bevat ook een aantal argumenten die in het oranje worden weergegeven: *x*, *y* en *touchedAnySprite*. Deze elementen zijn lokale variabelen.



Je kan de waarde ervan opvragen met 'get' en bepalen of veranderen met 'set'. De blokken hiervoor vind je in de lade van 'Built-In – Variables'.

Als je in het werkveld de muis boven deze variabelen plaatst, verschijnen deze opties ook.

Een klik op een button is een hele eenvoudige 'event handler'. De actie die hij moet waarnemen is namelijk heel simpel: wordt er op de knop gedrukt?

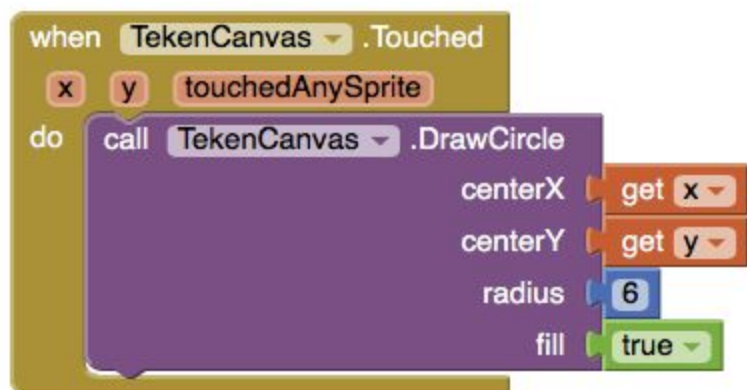
Andere event handlers zoals 'when... Touched' hebben echter wat meer informatie nodig over de gebeurtenis die plaatsvindt. Zo moet je bij het tekenen op het canvas natuurlijk weten waar het canvas wordt aangeraakt om de stip op de juiste plek te kunnen plaatsen.

Je telefoon signaleert op welke positie jouw vinger het scherm raakt. De *x*- en *y*-positie daarvan geeft hij vervolgens door met behulp van de argumenten *x* en *y*. Die waarden opvragen om op die plek op het scherm te kunnen tekenen, doe je met 'get *x*' en 'get *y*'. Het argument 'touchedSprite' doe je in deze app niks mee.

Om een kleine cirkel te tekenen waar jij het canvas aanraakt, gebruik je de blok 'call TekenCanvas.DrawCircle'.

De *x*- en *y*-positie van de cirkel worden bepaald met behulp van 'get *x*' en 'get *y*'.

In de lade van 'Built-in – Math' vind je een blauwe blok waar je een getal in kan vullen als waarde voor de radius van de cirkel.

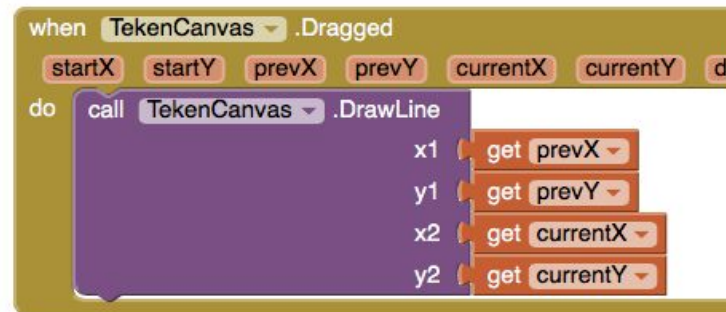


Het opvullen van de cirkel staat standaard aan, maar je kan dit natuurlijk wijzigen naar 'false'.

Drag

Op een vergelijkbare manier ga je nu een lijn laten tekenen op het moment dat je met je vinger over het scherm sleept. Het verschil met aanraken is dat er bij slepen maar liefst zes argumenten meegegeven worden aan het uitvoeren.

De conditional die je daarvoor nodig hebt heet 'when TekenCanvas.Dragged'. De actie die je erin sleept heet 'call TekenCanvas.DrawLine'. Vervolgens moet je de huidige positie (x2 en y2) en de daaraan voorafgaande positie (x1 en y1) van de vinger opnemen in het de actie. Zo wordt tussen de eerste en de tweede positie van je vinger, voor en na het slepen, een lijn getrokken.

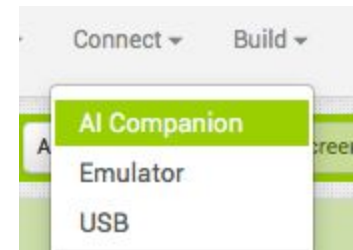


3. Testen

Na al dat programmeren wordt het natuurlijk tijd om je app te testen. Er zijn verschillende manieren om dat te doen. Deze vind je alledrie onder de menuknop 'Connect'.

AI Companion

In de Google Play store kun je de Android app 'MIT AI2 Companion' app vinden. Installeer deze om via een QR code een verbinding te maken tussen het project in de browser en jouw eigen telefoon. Je kan je app dan testen en alle wijzigingen worden direct zichtbaar op je telefoon. Een uitgebreidere beschrijving hiervan vind je [op deze website](#).



Emulator

Je kan, als je geen Android apparaat hebt, ook een emulator gebruiken. Dit is een programma dat een telefoon nabootst op je computer. Het programma heet AIStarter en kan [op deze website](#) gedownload worden.

USB

Je kan je toestel ook via een USB-kabel verbinden met App Inventor. De instructies daarvoor vind je op [deze website](#).

Afronding

Je hebt in deze les geleerd hoe je 'action handlers' voor knoppen programmeert. Je kan de

layout van menuknoppen aanpassen met een tabel, en je hebt conditionals geprogrammeerd die bepalen wat er moet gebeuren bij een druk op een knop. Je hebt twee manieren van interactie met de telefoon gecodeerd, het aanraken van en het slepen over het scherm.

In de volgende les ga je een foto maken met de camera van je toestel die je vervolgens als achtergrond van TekenCanvas kan instellen. Zo kun je lijnen tekenen op zelf gemaakte foto's.

Evaluatie

Wat ging goed?

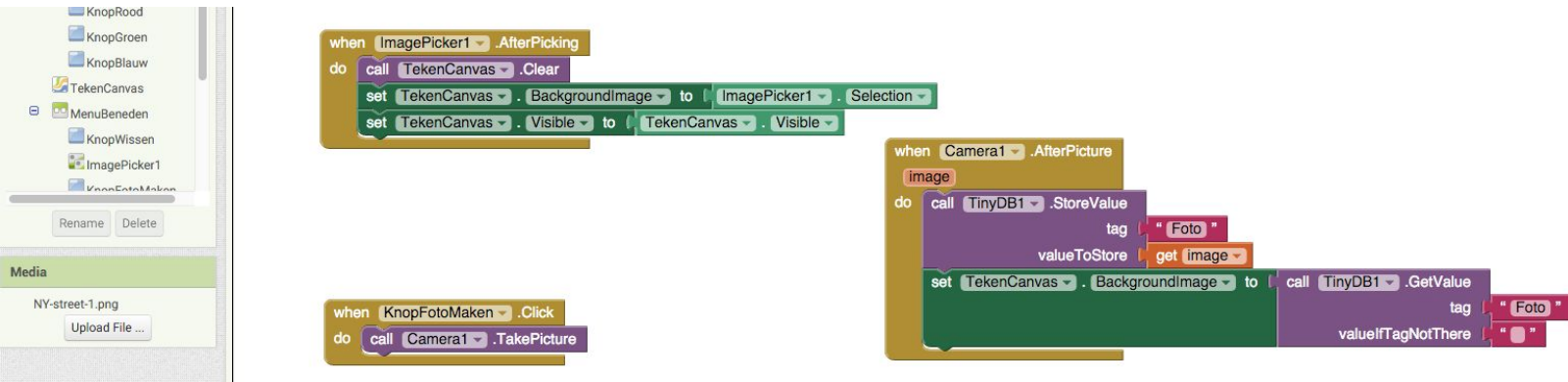
Waar had je hulp bij nodig?

Hoe zou deze les verbeterd kunnen worden?

Deze tutorial is gebaseerd op <http://appinventor.mit.edu/explore/ai2/paintpot-part1.html>.

Les 4: Camera integratie

Het maken van foto's integreren in de app



Doelen

In deze les leer je hoe je het kiezen en maken van foto's met je Android toestel kan programmeren en hoe je dit integreert in de Perspectief app.

Tijdsduur

Deze les duurt ongeveer 45 minuten.

Je hebt nodig

Een computer met een browser en internetverbinding. Een Android telefoon of tablet is niet noodzakelijk maar wel wenselijk.

Opdracht in het kort

Je hebt een app gebouwd waarmee je kan tekenen op een canvas. Om perspectieflijnen op een zelfgemaakte foto te kunnen maken, ga je in deze les het cameracomponent toevoegen en programmeren.

Perspectief

Ga naar <http://appinventor.mit.edu>. Kies rechtsbovenaan in het menu voor de optie 'Create apps!'. Je wordt naar een pagina geleid waar gevraagd wordt in te loggen met een Google Account. Open via de menuknoppen Projects – My projects vervolgens het project 'Perspectief' van de vorige les.

Je gaat nieuwe elementen toevoegen aan je app waardoor ze niet meer allemaal op het scherm passen zonder te scrollen. Klik rechts onder Components op Screen1. In de lijst met eigenschappen (Properties) plaats je het vinkje bij 'Scrollable'. Dit zorgt ervoor dat je omhoog en omlaag kan scrollen en daardoor alle elementen zichtbaar worden.

Je gaat twee opties toevoegen: Een foto kiezen uit het album en een nieuwe foto maken. Voor beide handelingen maak je aparte knoppen aan.

1. Foto kiezen uit album

Design

Om het kiezen van een foto te triggeren, moet je een knop toevoegen aan de user interface. Ga naar Designer Mode als je daar nog niet bent.

In het Palette onder de categorie Media staat 'ImagePicker'. Sleep deze naar de Viewer en plaats hem onder de button 'KnopWissen'. Verander de eigenschap 'Text' van deze ImagePicker naar bijvoorbeeld 'Foto kiezen'.



Net zoals bij de kleurenknoppen ga je de ImagePicker en de button KnopWissen ook horizontaal uitlijnen met behulp van een HorizontalArrangement. Je vindt deze in de categorie Layout van het Palette. Denk eraan dat je de 'width' van deze ook weer op 'Fill parent' instelt.

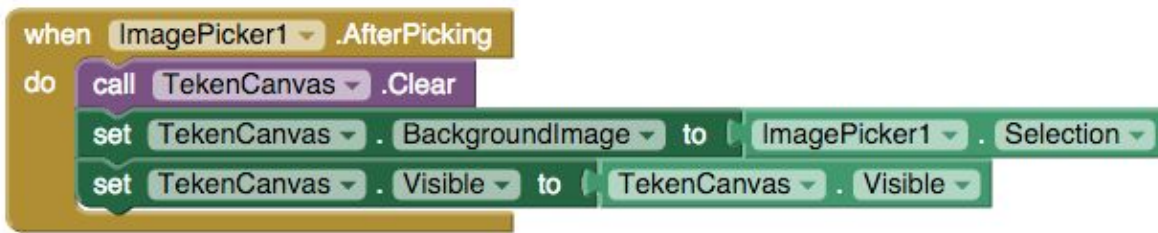
Code

Ga naar de Blocks Editor. Wanneer je op ImagePicker1 klikt, verschijnt de lade met daarin onder meer de blok 'when ImagePicker1.AfterPicking do'. Wanneer de gebruiker klikt op 'Foto maken' wordt het album met bestaande foto's geopend en kun je een foto selecteren. Met behulp van deze conditional ga je de app vertellen dat hij die selectie als achtergrond van TekenCanvas moet instellen.



Voordat de geselecteerde foto wordt geplaatst, moet het TekenCanvas gewist worden. Sleep de actie 'call TekenCanvas.clear' uit de lade van TekenCanvas naar de conditional.

Daarna ga je de achtergrond van het canvas instellen. Hiervoor gebruik je de actie 'set TekenCanvas.BackgroundImage to'. Om het geheel compleet te krijgen moet je aan deze laatste actie iets toevoegen, namelijk de foto die je zojuist gemaakt of geselecteerd hebt uit de bibliotheek. In de lade van de ImagePicker1 selecteer je het element 'ImagePicker1.Selection' en voegt deze toe aan de actie. Deze verwijst namelijk naar de foto die je hebt gekozen.



De laatste actie die toegevoegd moet worden is het zichtbaar (Visible) maken van het TekenCanvas voor het geval dat niet standaard zo is. De blokken hiervoor vind je in de lade van TekenCanvas.

2. Nieuwe foto maken

Design

In de Designer mode sleep je uit de lade Media van het Palette de camera-component naar de viewer. Je zal in de viewer zelf niets zien verschijnen maar daaronder komt wel een melding dat er een non-visible component is toegevoegd. Het camera-component heeft dus geen zichtbare onderdelen maar je kan er in de Blocks Mode wel mee programmeren.

Natuurlijk heb je ook een button nodig om op te klikken wanneer je een foto wil maken. Sleep uit het Palette een nieuwe button naar de viewer en noem deze KnopFotoMaken.

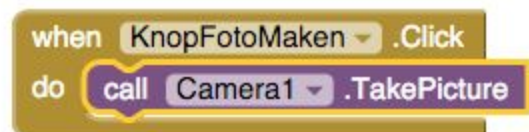


Wanneer de foto is gemaakt, moet deze tijdelijk opgeslagen worden. Dit gebeurt met behulp van een TinyDB-component. Dat vind je in het Palette onder Storage. Net zoals de camera is dit een non-visible component.

Code

Je wil een foto laten maken met een druk op KnopFotoMaken. Daarvoor gebruik je de event handler 'when KnopFotoMaken.Click do'. Als actie plaats je daarin 'call Camera1.TakePicture' uit de lade van de camera-component.

Vervolgens ga je de event handler programmeren die bepaalt wat er moet gebeuren na het maken van de foto. Deze vind je in de lade van Camera1. Je gaat de waarde van de gemaakte foto opslaan



in de database TinyDB1. Kies daarom uit die lade de actie 'call TinyDB1.StoreValue'.

Je hebt bij het doorgeven van de positie van de vinger gezien dat je een doorgegeven argument kan gebruiken in de event handler. Ga met je muis boven het oranje onderdeel 'image' en kies voor 'get image'. Deze sleep je naar de opening van 'valueToStore'.

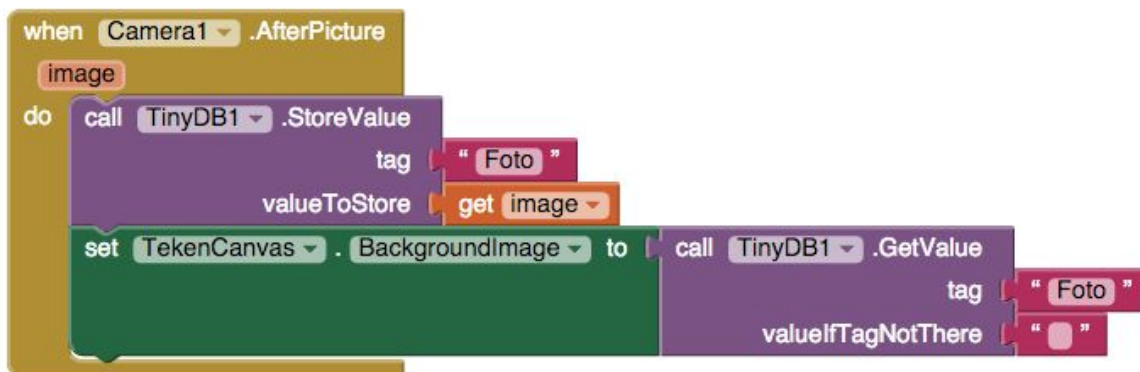


Uit de categorie Built-in kies je voor Text.

Deze sleep je naar de opening bij 'tag'.

Wat je hier als tag invult mag je zelf bepalen. Je gebruikt deze naam zometeen wel weer om de foto uit de database te halen en als achtergrond van het TekenCanvas in te stellen.

In dezelfde event handler ga je ook weer de waarde uit de database ophalen en instellen als achtergrond van TekenCanvas. Uit de lade van dat component sleep je 'set TekenCanvas.BackgroundImage to'. Daar plak je vervolgens uit de lade van TinyDB1 aan vast: 'call TinyDB1.GetValue'. Als tag gebruik je natuurlijk een tekstblok met daarin 'Foto'.



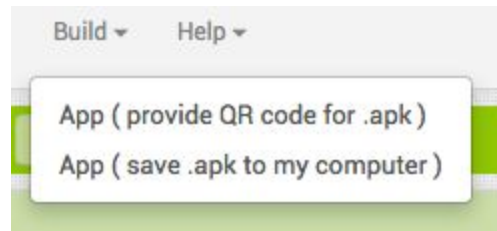
Je bent nu klaar met de layout en de code van de app. Het enige dat nog resteert is het god testen van alle functies en het exporteren zodat iedereen jouw app kan installeren.

Exporteren & installeren

Nu je app helemaal klaar is, moet je hem natuurlijk eerst goed testen. In de vorige les heb je geleerd hoe je dat doet.

Weet je zeker dat hij goed werkt, dan kun je hem exporteren met behulp van de menuknop 'Build - App'. App Inventor maakt er dan een installatiebestand (.apk) van dat je op elk Android toestel kan openen.

Iedereen met wie je dit bestand deelt, kan jouw app dus installeren.



Afronding

Gefeliciteerd met het afronden van je eerste app!

Wil je verder oefenen met App Inventor, dan zijn er talloze tutorials beschikbaar op de website van MIT maar ook elders op internet. Zoek maar eens op Youtube, bijvoorbeeld.

Succes!

Evaluatie

Wat ging goed?

Waar had je hulp bij nodig?

Hoe zou deze les verbeterd kunnen worden?

Deze tutorial is gebaseerd op <http://appinventor.mit.edu/explore/ai2/paintpot-part1.html>.