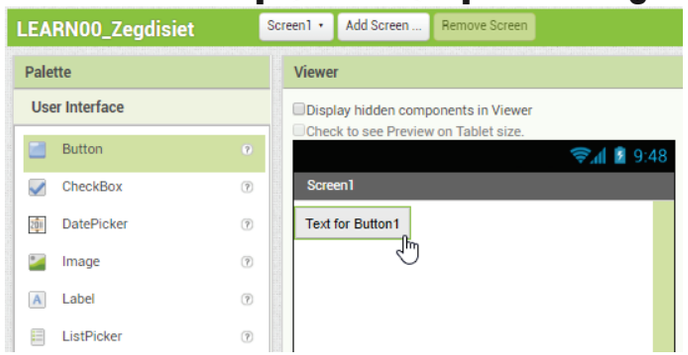
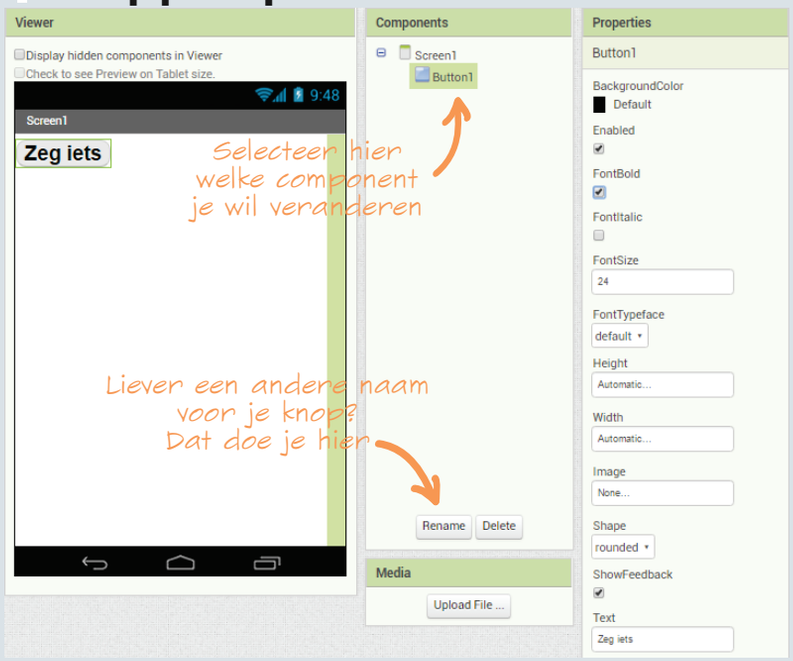
Een app die onzinnige dingen uitkraamt telkens je op een knop drukt

0 Plaats een knop *(Button)* op het designer scherm



In designer-mode teken je het uiterlijk van je app: knoppen, tekstvelden,... voeg je toe door ze vanuit het pallet *(Palette)* op het voorbeeldscherm *(Viewer)* te slepen.

1 Pimp je knop

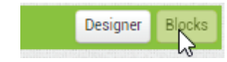


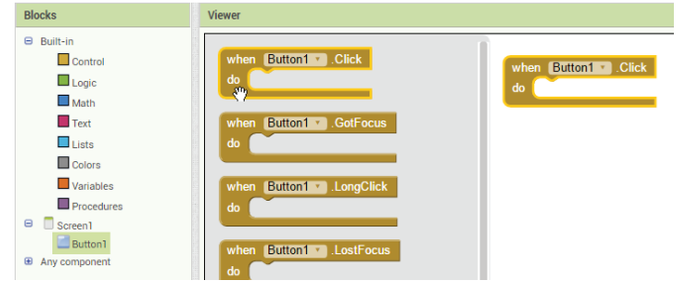
Het is jouw app, dus jij beslist hoe alles eruit ziet.

In het eigenschappen-venster *(Properties)* kan je de kleur, grootte, tekst,... van je knop instellen.

2 Voeg een klik-actie blok toe

Dingen die er goed uitzien maar niets doen zijn meestal niet zo interessant. Om je app iets te laten doen moet je blokken *(Blocks)* toevoegen.

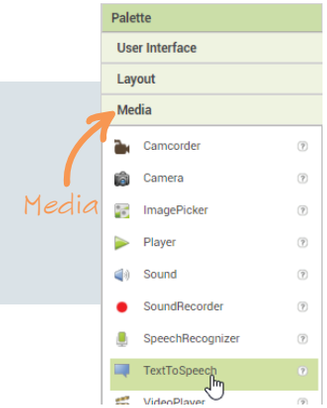




Selecteer **"Button1"** en voeg een **"when Button1.Click do"**-blok toe. Wat de app zal doen moet nog in het blok ingevuld worden. Voorlopig doet de app dus nog niets.

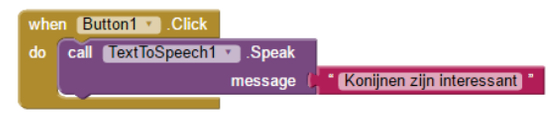
3 Voeg een *TextToSpeech* component toe

Ga terug naar *Designer*-mode en voeg een ***TextToSpeech*** (TekstNaarSpraak) component toe vanuit *Media*. Deze component zal onzichtbaar *(Invisible)* zijn in je app en wordt daarom onder het voorbeeldscherm getoond.



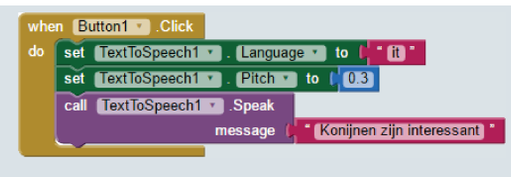
4 Maak de puzzel verder af

Voor elke component die je in *Designer*-mode toevoegt vind je nieuwe blokken terug. Met een **"call TextToSpeech1.Speak message"**-blok ("TekstNaarSpraak1.Spreek bericht"-blok) kan je app echt praten. Voeg nog een tekstblok toe, anders heeft je app geen idee wat hij moet zeggen.

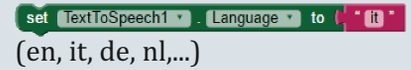


5 Speeltijd

De grootste uitvindingen zijn gedaan door wat te spelen en prullen met schijnbaar nutteloze dingen. Speeltijd nemen is dus enorm belangrijk :-)

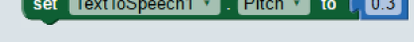


Probeer eens een andere taal



(en, it, de, nl,...)

of andere toonhoogte

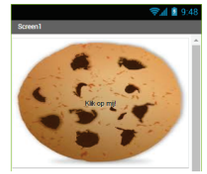


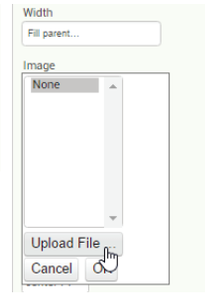
of...

author: @kurtdevocht

Een remake van één van de spannendste games uit de recente geschiedenis: tellen hoe vaak je op een koekje klikt!

0 Eerst een koekje

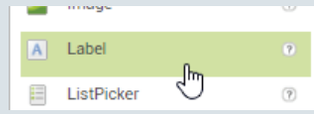


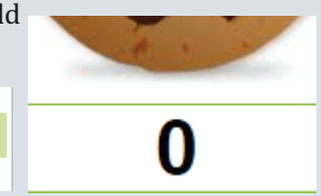


Zoek of teken een afbeelding van een koekje. Klik je liever op iets anders, zoek dan een afbeelding van iets anders. Doe vooral uw goesting :-) Plaats een knop *(Button)* op het scherm. In de eigenschappen van de knop kan je de koekjesafbeelding uploaden.

1 Voeg een label toe

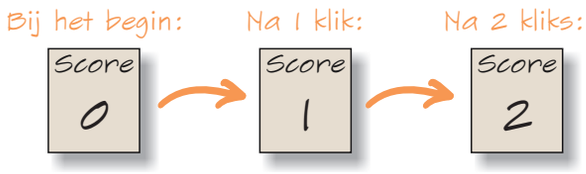
Om informatie (zoals je score) te tonen kan je een tekstveld *(label)* gebruiken. Voeg een tekstveld toe en verander de kleur, grootte,... naar eigen smaak.





3 Score onthouden met een papiertje

Je dient ergens bij te houden hoe vaak je op het koekje klikt. In de echte wereld zou je daarvoor een papiertje kunnen gebruiken: om te beginnen schrijf je met een potlood een '0' op het papiertje. Om te onthouden wat het getal betekent zet je een titel bovenaan, bijvoorbeeld **'score'**. Telkens er iemand klikt, gom je uit wat er op het papiertje stond en schrijf je er een getal op dat 1 meer is.



4 Score onthouden met een variabele

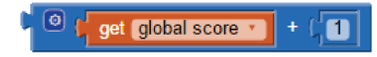
Potlood en papier in je app stoppen is natuurlijk geen optie. Om 'iets' tijdelijk te onthouden in een computerprogramma gebruik je een **variabele**. Een variabele heeft een **waarde** (wat er op het papiertje staat) en een **naam** (de titel van het papiertje).

- Voeg een *'initialize global (name) to'*-blok toe. Dit blok maakt een nieuwe variabele en stopt er meteen een beginwaarde in;
- Verander de naam naar 'score'. Als je liever 'banaan' als naam hebt dan mag dat uiteraard ook, maar dan is het iets minder duidelijk dat deze variabele gebruikt zal worden om de score bij te houden ;-)
- Hang de beginwaarde '0' aan het blok.

3 Score verhogen

Om de waarde van een variabele te lezen gebruik je een *get*-blok, om te schrijven een *set*-blok. Gebruik een *+*-blok om een nieuwe waarde te berekenen door de huidige score en 1 op te tellen.

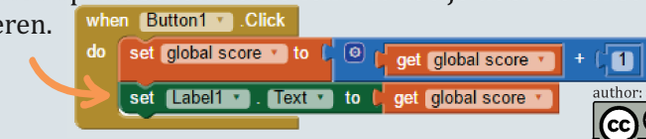
Dit blok telt 2 dingen op: de waarde van 'score' en '1'



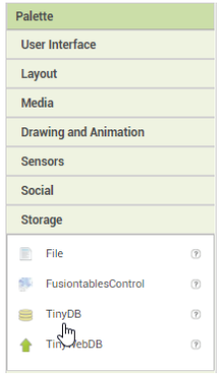
Met een *set*-blok stop je het resultaat van deze uiterst gesofisticeerde berekening terug in de score-variabele . Dit doe je telkens er op de koekjesknop geklikt wordt.

4 Score tonen

Voeg blokken toe om de score op het scherm te tonen. Dit doe je door de tekst van het label telkes te veranderen.



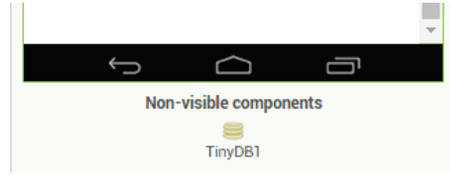
0 Klikwerk weggooien is jammer...



V1.0 van koekjesklikker bewaart de score in een variabele. Als je de app afsluit dan wordt de waarde van de score-variabele 'vergeten'. Je dient dus telkens terug vanaf 0 te beginnen.

Als je eeuwig de score wil bijhouden dan kan je deze opslaan in het interne geheugen van je Android toestel, bijvoorbeeld door gebruik te maken van een **database**. In een **database** kan je verschillende waardes (getallen, tekst,...) bewaren die je 'eeuwig' wil bijhouden. Er zijn bijzonder veel soorten databases. *TinyDB* is een database die je in App Inventor kan gebruiken om waardes te bewaren op je Android toestel zelf.

- Plaats een koekjesknop *(Button)* en een tekstveld *(Label)*, net zoals in V1.0 van de koekjesklikker app;
- Plaats een *TinyDB*-component op het scherm.



1 Lees de score uit TinyDB

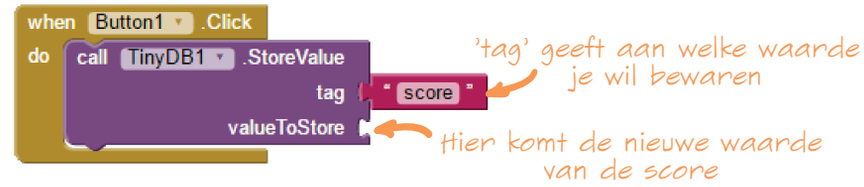
In *TinyDB* kunnen verschillende waardes (getallen, teksten,...) bewaard worden. Om aan te geven welke waarde je wil lezen of schrijven gebruik je een ***tag***. De waarde van de tag kies je zelf, net zoals je de naam van een variabele zelf kan kiezen. Als je alles duidelijk wil houden kies je best een logische naam. 'score' om een score te bewaren bijvoorbeeld ;-)

Met een '*call TinyDB.GetValue*'-blok kan je een waarde uit TinyDB uitlezen. Gebruik dit om bij het opstarten van de app de score te tonen in het label.

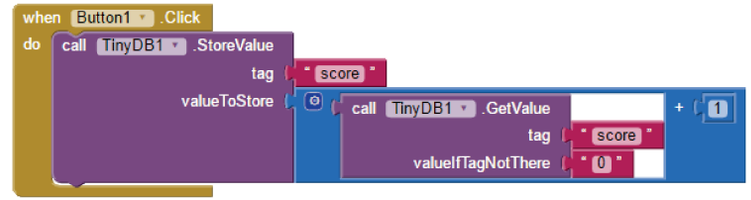
De eerste keer je de app start zullen er nog geen waardes in TinyDB zitten. Met 'valueIfTagNotThere' geef je aan welke waarde je wil als een (nog) niet bestaande tag gevraagd wordt.

3 Score verhogen

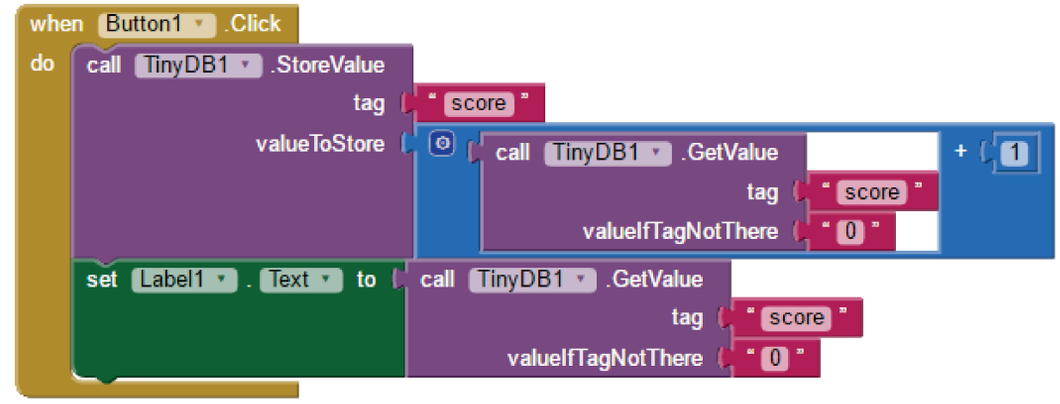
Om een nieuwe score te bewaren gebruik je een '*call TinyDB.StoreValue*'-blok.



De nieuwe waarde van de score wordt berekend door bij de oude waarde 1 op te tellen. Gebruik hiervoor een *+*-blok. De oude waarde lees je uit met een '*call TinyDB.GetValue*'-blok.



Om de gewijzigde score te tonen dienen er nog blokken toegevoegd te worden om de tekst van het label te wijzigen telkens er op de knop geklikt wordt.



Ga verder op de achterkant

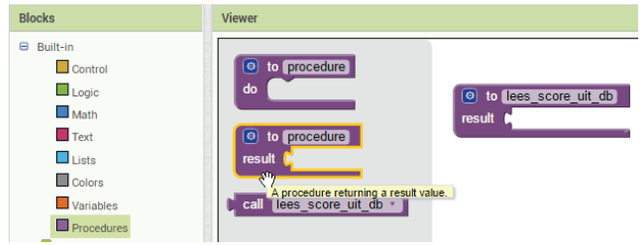
author: @kurtdevocht

4 Herhaal jezelf niet steeds...

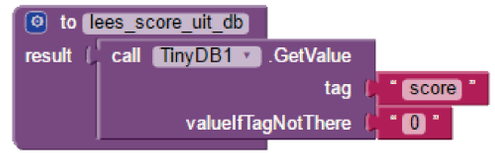
Je hebt misschien gemerkt dat zowel bij het opstarten van de app (*'when Screen.Initialize'*-blok) als bij het klikken (*'when Button.Click'*-blok) de waarde van de score uit TinyDB gelezen wordt en je hiervoor steeds dezelfde blokken gebruikte. Als je merkt dat op verschillende plaatsen in je app dezelfde blokken herhaald worden kan je hiervan een **procedure** maken.

Er zijn 2 soorten procedures. Procedures die iets doen zonder een waarde terug te geven maak je met een 'to *procedure* ***result***'-blok. Procedures die een waarde teruggeven maak je met een 'to *procedure* ***result***'-blok.

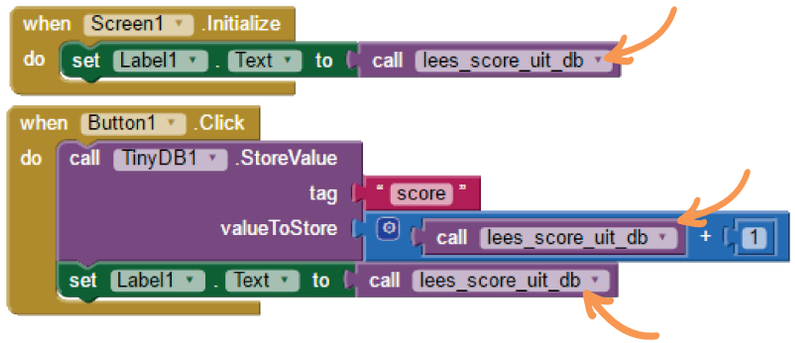
Plaats een 'to *procedure result*'-blok en geef het een duidelijke naam, bijvoorbeeld 'lees_score_uit_db'



App Inventor heeft geen flauw benul wat je met 'lees_score_uit_db' bedoelt zolang je geen blokken plaatst in het procedure-blok. Plaats in de procedure de blokken om de score uit TinyDB te lezen.

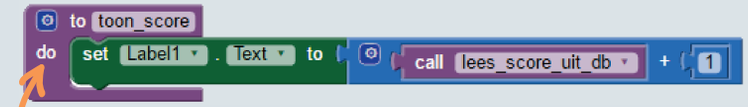


Op de plaatsen waar de score uit de database gelezen wordt kan je nu de procedure oproepen. Dat maakt je programma duidelijker en makkelijker aan te passen.



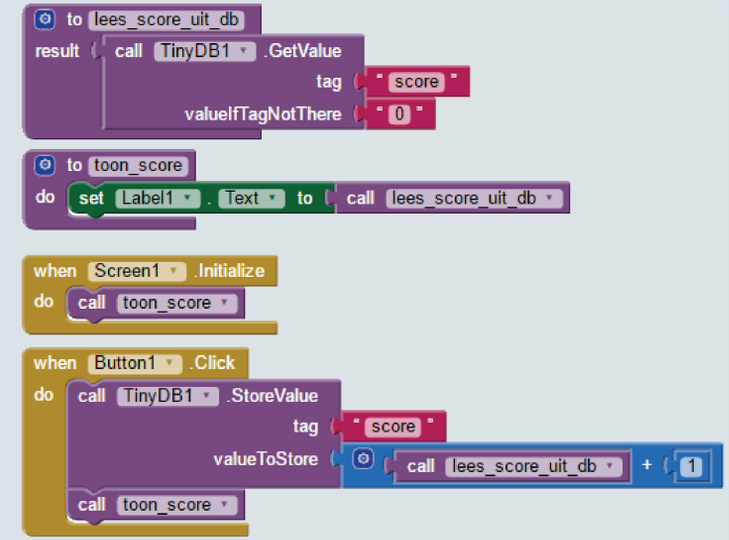
5 Herhaal jezelf niet steeds...

Heb je gemerkt dat er op 2 plaatsen een *'set Label.Text'*-blok gebruikt wordt om de score te tonen? Als je wil kan je ook dit in een procedure gieten.



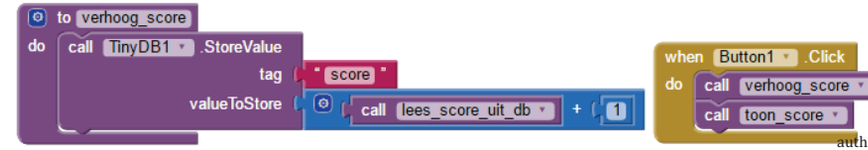
Omdat deze procedure geen waarde teruggeeft gebruik je een 'to procedure *do*'-blok

Je volledige programma ziet er dan als volgt uit. Netjes toch?



6 Nog meer procedures?

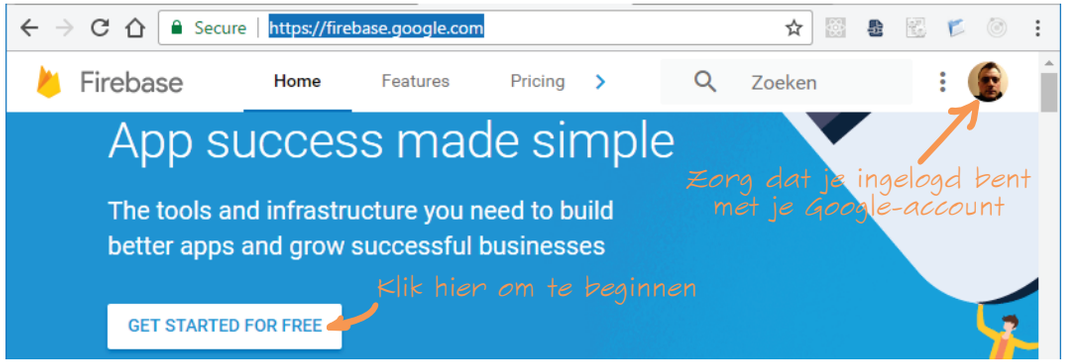
Procedures kan je ook gebruiken om je programma duidelijker te maken. Je zou bijvoorbeeld een procedure kunnen maken om de score te verhogen, ook al wordt dit maar op 1 plaats gedaan. Zo wordt de code in je *'when Button.click'*-blok superduidelijk.



0 Samen klikken is plezanter

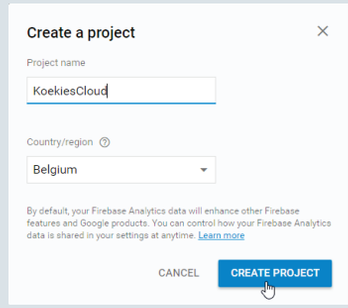
V2.0 van koekjesklikker bewaarde de score op de interne opslag van jouw smartphone zelf. Als iemand anders op haar smartphone jouw app gebruikt dan zal zij haar eigen score hebben, beginnend vanaf 0. Iedereen heeft dus z'n eigen klik-score.

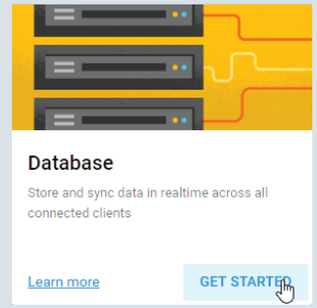
Zou het niet veel plezanter zijn om samen op een koekje te klikken en een gemeenschappelijke score te hebben? Hiervoor kunnen we de score niet langer op het interne geheugen van een toestel bewaren maar hebben we opslag nodig 'in de cloud' (op het internet dus). Er zijn massaal veel manieren om informatie op het internet te bewaren. Voor deze app zullen we gebruik maken van **Firebase**, een gratis cloud-database van Google. Surf naar **https://firebase.google.com/** en klik op **„GET STARTED FOR FREE“**



1 Maak de database aan

- Maak een nieuw Firebase project. Kies een naam en klik op **'CREATE PROJECT'**;
- Klik bij **'Database'** op **'GET STARTED'**. Er zal een nieuwe cloud-database aangemaakt worden.

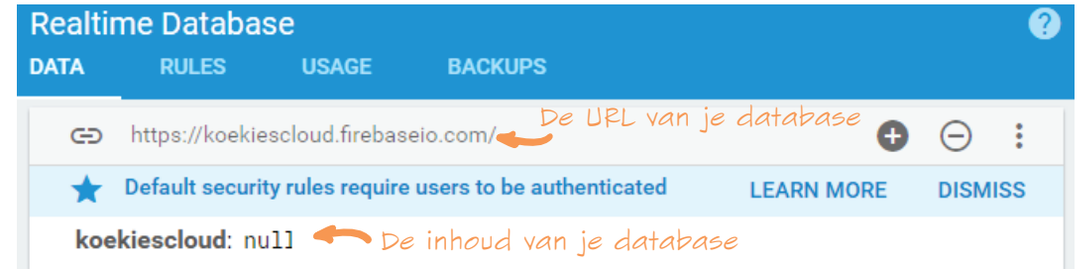




2 Database dashboard

Je komt in het dashboard van je database terecht:

- Een URL is een adres op het web. Ook je Firebase database heeft een URL: deze vind je bovenaan. Zonder de URL weet je app niet met welke database hij moet verbinden;
- In het dashboard zie je ook de inhoud van je database. Een nieuwe database is leeg, dat wordt met de tekst 'null' aangeduid.



3 Iedereen welkom

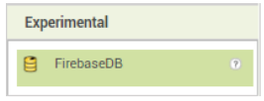
De standaard Firebase beveiligingsregels laten niet zomaar iedereen toe om in database te lezen of te schrijven. Om dit voorbeeld eenvoudig te houden gaan we de beveiliging uitschakelen zodat iedereen erin kan lezen en schrijven. **Iedereen die de URL weet zal dus gegevens kunnen veranderen!** Voor onze speel-app is dat nog wel ok.

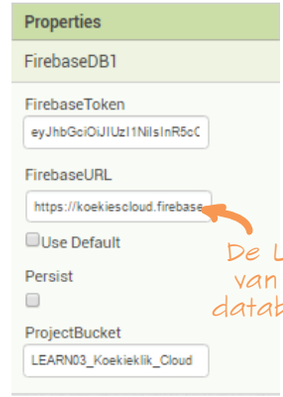
Ga verder op de achterkant

author: @kurtdevocht

4 Verbind je app met Firebase

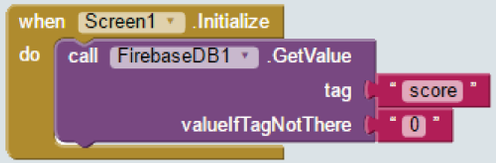
- Plaats een koekjesknop en scorelabel;
- Plaats een FirebaseDB-component. Deze vind je in de categorie 'Experimental';
- Selecteer de FirebaseDB-component en vul de URL van je database in.

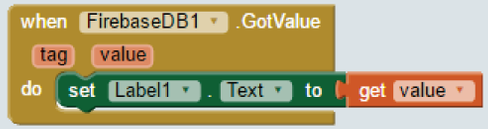




5 Lees de score als je app opstart

- Gebruik een *'call FirebaseDB.GetValue'*-blok om de tag 'score' uit de Firebase database te lezen. Anders dan bij TinyDB geeft dit blok de uitgelezen waarde niet meteen terug;
- Zodra de waarde ontvangen werd zal het *'when FirebaseDB.GotValue do'*-blok opgeroepen worden. Dat gaat meestal supersnel - dat hangt af van de verbindingssnelheid met de Firebase-server. In dit blok krijg je 2 variabelen: **tag** (de tag-naam van de gevraagde waarde) en **value** (de eigenlijke waarde).

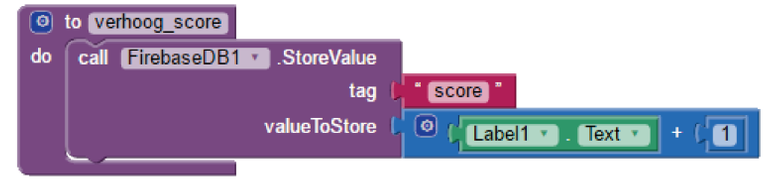




Het patroon waarbij commando's om data op te vragen op de achtergrond uitgevoerd worden en een melding geven zodra ze klaar zijn noemt men **asynchroon programmeren**, een concept dat in quasi alle moderne programmeertalen onderteund wordt.

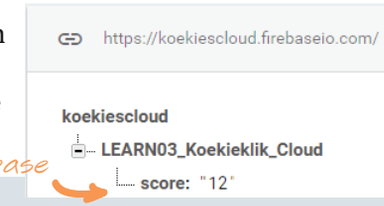
6 Score verhogen

Gebruik een *'call FirebaseDB.storeValue'*-blok om een nieuwe score in de database te stoppen.



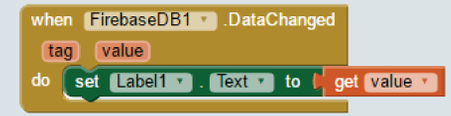
Telkens je op de knop klikt zal er een nieuwe score in de database gestopt worden. Dat zie je ook in het database-dashboard! In je app zie je de nieuwe score nog niet.

Telkens je op de knop drukt verandert de waarde in de database



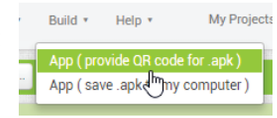
7 Nieuwe score tonen

Plaats een *'when FirebaseDB.DataChanged do'*-blok. Dit zal uitgevoerd worden zodra er iets in de database verandert. Het maakt niet uit wie of wat de database verandert heeft! Ook als je manueel in het dashboard een nieuwe score ingeeft zal je label een nieuwe waarde tonen.



8 Test met een 2de toestel

- Maak eerst een '.apk'-bestand van je app. Klik in het App Inventor menu op 'Build' > 'App (provide QR code for .apk)';
- Na enkele seconden wordt een QR-code getoond. Deze bevat een link naar het .apk bestand. Open deze link op een 2de toestel en installeer de app;
- Je kan nu op beide toestellen vollen bak koekjes klikken :-)



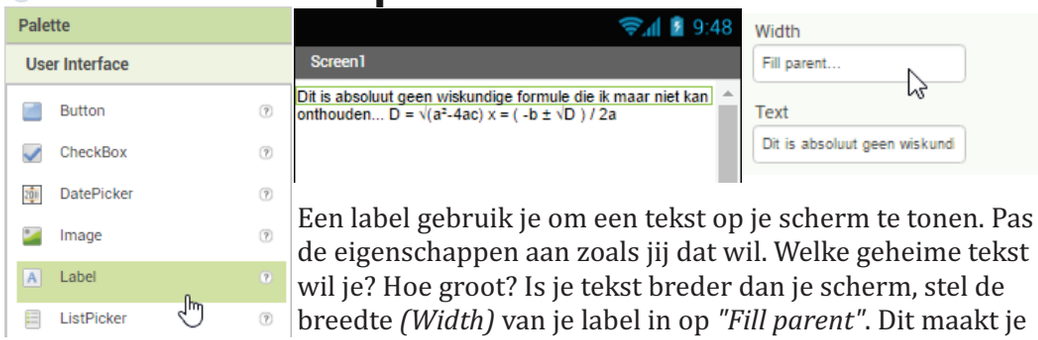
9 Nog niet perfect... (maar al wel plezant)

Wat gebeurt er als je op beide toestellen quasi tegelijkertijd op de knop klikt? Score + 2? Of toch + 1? Denk daar maar eens over na hoe dat komt ;-) Dit oplossen met de huidige App Inventor componenten is bijzonder lastig....

author: @kurtdevocht

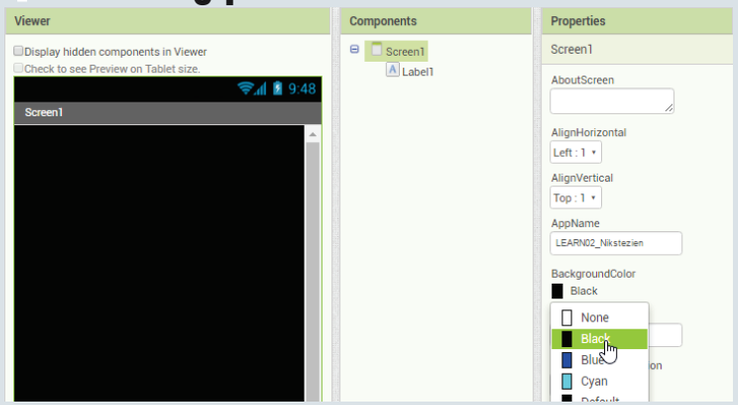
Een app die een geheime tekst op het scherm toont.
Of toch niet? Gebruik tijdens de examens is op eigen risico :-)

0 Plaats een *Label* op het scherm

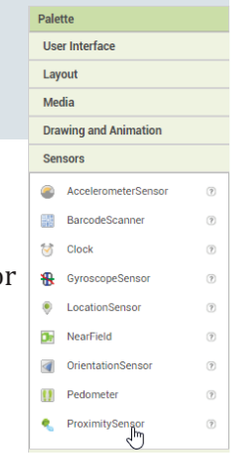


Een label gebruik je om een tekst op je scherm te tonen. Pas de eigenschappen aan zoals jij dat wil. Welke geheime tekst wil je? Hoe groot? Is je tekst breder dan je scherm, stel de breedte *(Width)* van je label in op *"Fill parent"*. Dit maakt je label even breed als het scherm.

1 Verberg je tekst



Maak de achtergrondkleur *(BackgroundColor)* van het scherm zwart *(Black)*. Je geheime tekst is nu helemaal onzichtbaar.



2 Voeg een klik-actie blok toe

We gaan de tekst terug laten verschijnen als je je hand dicht bij je toestel houdt. Hiervoor maken we gebruik van de nabijheidssensor die (hopelijk) in je toestel ingebouwd zit. Sleep een *ProximitySensor*-component op het scherm.

3 Voeg een *ProximityChanged* blok toe

De app moet iets doen telkens de nabijheidssensor van waarde verandert. Dat is exact waarvoor het *ProximityChanged*-blok dient. Alles wat hiertussen geplaatst wordt zal uitgevoerd worden telkens de ingebouwde sensor een andere waarde meet.

4 Als ... dan ...

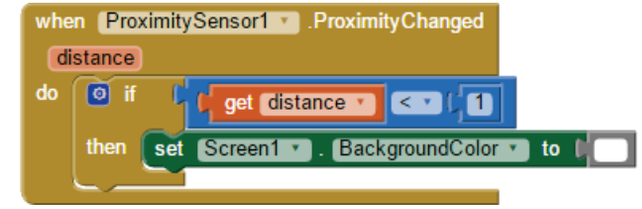
<u>Als</u> je hand dicht genoeg bij je smartphone is **dan** moet de tekst zichtbaar zijn.
Hiervoor wordt een als-dan-blok *(If-then)* blok gebruikt. Aan het *if*-stuk plak je iets dat **waar** of **niet waar** kan zijn. Wat je in het *then*-stuk plakt zal enkel uitgevoerd worden als het *if*-stuk waar is.

3 Is je hand dichtbij?

Het *distance* (afstand) veld van het *ProximityChanged*-blok bevat steeds de afstand van je hand. Meestal geeft dit veldje de waarde 1 als je hand veraf is, en 0 als je hand dichtbij is. In het *If*-stuk kunnen we dus testen of de waarde van *distance* kleiner is dan 1. Hiervoor gebruik je blokken die je bij *Math* (wiskunde) terugvindt.

4 Maak de tekst weer zichtbaar...

...door de achtergrondkleur weer wit te maken.



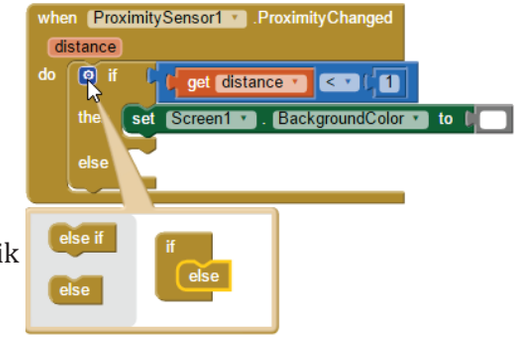
De tekst zal nu terug zichtbaar worden als je hand dichtbij is. Doe je je hand weer weg, dan blijft de tekst nog steeds zichtbaar. Nog niet helemaal spiekbestendig dus...

Ga verder op de achterkant ➡

author: @kurtdevocht

5 Als ... dan ... anders ...

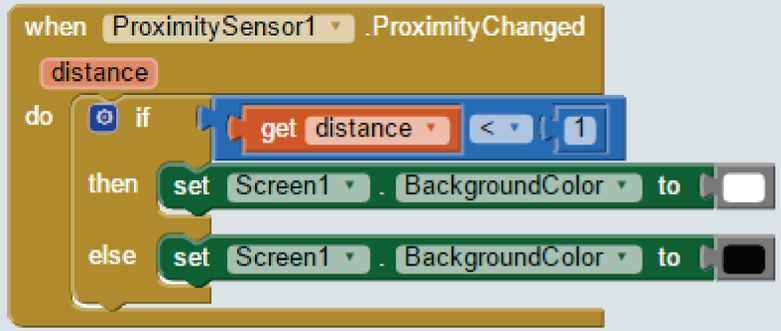
Er ontbreekt nog een stukje logica in onze app. **Als** je hand dichtbij is **dan** moet de tekst zichtbaar zijn, **anders** moet de tekst onzichtbaar zijn.



Hiervoor dienen we nog een anders-stuk *(else)* toe te voegen aan het *if-then*-blok. Klik hiervoor op het blauw-witte icoontje met het tandwiel.

6 Verberg je tekst

In het *if*-stuk wordt gekeken of de gemeten afstand *(distance)* kleiner is dan 1. Als dit **niet waar** is (dus als je hand veraf is) dan zal uitgevoerd worden wat in het *else*-stuk staat. Hierin kan je dus blokken plaatsen om de tekst weer onzichtbaar te maken.



5 Speeltijd

Er zitten wellicht nog tal van sensoren in je Android-toestel. Vis uit welke. Neus rond in de categorie *Sensors* en zoek uit wat de App Inventor componenten doen

"We don't stop playing because we grow old;
We grow old because we stop playing."

-- George Bernard Shaw

TextToSpeech zegt niks...

- Uiteraard heb je al nagekeken of het volume van je toestel niet uitstaat, toch?
- Ga naar de Play Store en kijk na of de 'Google Text-to-speech (Google Tekst-naar-spraak)'-app geïnstalleerd is. Normaal gezien wordt dit voorgeïnstalleerd met Android, maar dit is niet het geval bij alle toestellen;
- Test met een andere app (bijvoorbeeld Google Translate) of de tekst-naar-spraak functionaliteit werkt op je toestel;
- Telkens je een nieuwe taal voor de eerste keer gebruikt dan zal een nieuw taalpakket vanaf het internet geïnstalleerd worden. Dit werkt niet als je toestel niet met het internet verbonden is. Kies dus een andere taal, of verbind je Android-toestel met het internet.

Mijn sensor doet niks...

- Niet alle toestellen hebben dezelfde sensoren. Installeer de app **"Sensors Multitool"** vanuit de Play Store. Deze app laat zien welke sensoren jouw toestel heeft en toont alle meetwaarden van de sensoren.