

Inhoudsopgave

1	Configuratie	1.1
2	Een Beginscherm!	1.2
3	Een Minigame maken	1.3
4	Een Minigame met lijsten	1.4
5	Procedures	1.5
6	Controleer Alles!	1.6
7	De Score tonen	1.7
8	Afsluiten	1.8

- 1 Wat we behandelen:
 - Hoe we een bestand gebruiken
 - Hoe de speler input geeft
 - Introductie over lijsten
 - Een app die uit te breiden is met minigames
 - Minigames toevoegen (we geven je wat ideeën!)
- 2 Voor je begint, heb je een account voor App Inventor nodig. Gebruik het account dat je bij de Beginner Sushi kaarten van App Inventor hebt aangemaakt. Heb je nog geen account, kijk dan hier: <http://kata.coderdojo.com/images/7/7b/App-inventor-beginner-dutch-nl.pdf>
- 3 **Bij deze stap heb je wellicht een Dojo mentor nodig!** Afhankelijk van wat jijzelf hebt, of wat beschikbaar is bij jouw Dojo, moet je misschien de emulator installeren, je eigen of een Dojo apparaat configureren, of helemaal niets. Welke configuratie je ook moet uitvoeren, klik op de juiste link en volg de instructie.
- 4 Mooi! Nu kun je je eigen Android minigames in App Inventor gaan maken!
- 5 Zet de taal in de App Inventor browser op 'Nederlands' en klik bij 'Projecten' op 'Start een nieuw project'. Geef je project een naam.

- 1 Een **beginscherm** kan informatie aan de spelers tonen. Dit wordt ook wel een start- of hoofdscherm genoemd. Je kunt je eigen beginscherm maken door een nieuw scherm toe te voegen en het een naam te geven (wij kiezen voor 'Beginscherm').
- 2 Wat weet je nog van de Beginner kaarten? Voeg twee labels en één knop toe aan het scherm. Geef het eerste label de naam van je app.

NB: wij vinden het fijn als het beginscherm gecentreerd is en niet alles helemaal links staat. Klik op je scherm in het **Componenten** menu, dan zet je onder **Eigenschappen** de **UitlijningHorizontaal** op **Midden:3**.
- 3 Tijd voor iets nieuws! Je kunt input van de speler krijgen met het **TekstVak** (**Palet > Gebruikersomgeving**). Sleep er één naar je scherm.
- 4 Je laat je speler verschillende minigames spelen, dus je moet gaan bijhouden hoeveel spelletjes ze winnen. Sleep een **Bestand** (**Palet > Opslag**) naar je scherm. De app zal dit gebruiken om de score van de speler bij te houden!
- 5 Maak je beginscherm wat netter. Zo ziet ons beginscherm eruit:



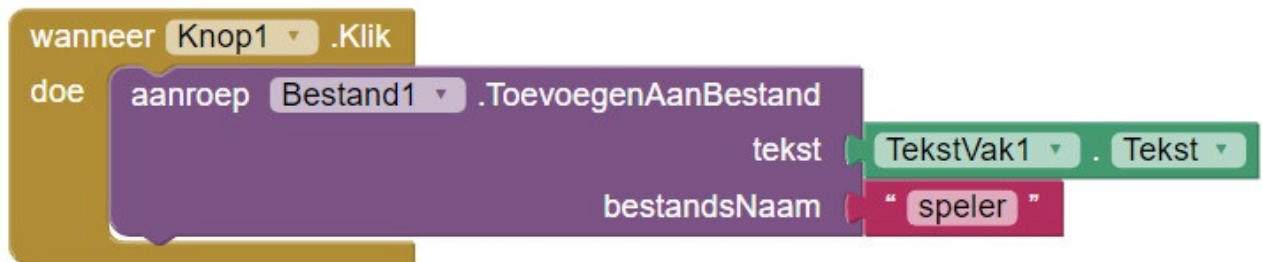
NB: dit is wat wij gedaan hebben, maar je kunt zelf je opmaak bepalen!

- 6 Hoog tijd om code toe te voegen aan beginscherm. Open de Blokken editor rechtsboven in je scherm.
- 7 Zoek het `wanneer [Knop1]. Klik` blok en zet er een `aanroep [Bestand1]. ToevoegenAanBestand` blok in vanuit **Bestand1**.

- 8 Voeg een blanco **Tekst** blok toe aan de bestandsNaam plaats. Zet "speler" in het **Tekst** blok. Dit bestand zal informatie over de speler bewaren. Om input van de speler te krijgen, zal het blok dat je nodig hebt onder het **Tekst** blok zitten en het ziet er zo uit:



Sleep het naar de tekstplek:



Dat is het! Je hebt zojuist de input van de speler in het "speler"bestand opgeslagen.

- 9 Om het blok af te maken gebruik je het aanroep [TekstVak1]. VerbergToetsenbord blok van **TekstVak1** en vanuit **Controle** het sluit venster met waarde resultaat blok. Zet een tekstblok met "Beginscherm" eraan vast. Zoals dit:



- 10 Prima! Nog één ding, je moet de speler het beginscherm laten zien. Op dit moment laat App Inventor als eerste Screen1 zien. Om het beginscherm te openen als de app opstart kun je deze blokken gebruiken (in het Screen1 blokken gedeelte):



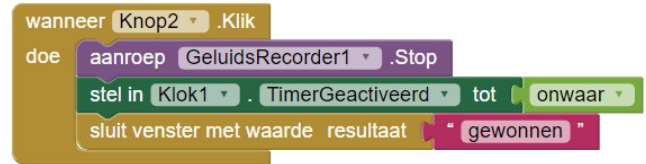
- 11 Als je je app via de emulator of Android apparaat uitvoert, kun je zien hoe je beginscherm eruit ziet.

- 1 Deze app gaat helemaal over minigames! De eerste minigame zal je reflexen testen. Maak een nieuw scherm met de naam "Minigame1".
- 2 Voeg een label toe en verander de tekst naar "Neem snel een geluid op" (zo weet de speler wat er verwacht wordt!). Voeg ook twee knoppen "Opnemen" en "Stop" toe. Zet bij de Stopknop de **zichtbaarheid** uit. Via **Palet > Media** voeg je een **GeluidsRecorder** toe aan je scherm.
- 3 Nu heb je nog een klok (**Palet > Sensoren**) nodig. Klik op de klok in **Componenten** en zet de **TimerInterval** op 3000 in de **Eigenschappen**.

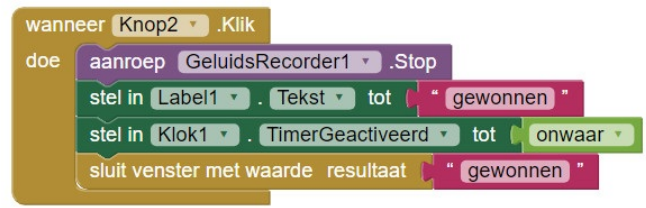


- 4 Ga nu naar de **Blokken** editor. De klok ga je gebruiken om te zien of de speler snel genoeg een geluid opneemt!
- 5 Nu moet de minigame bepalen of de speler gewonnen of verloren heeft. Als de speler een geluid opneemt vóór het klokalarm afgaat wint de speler! Zo niet...dan verliest de speler.
- 6 Sleep twee `wanneer [Knop].Klik` blokken naar het scherm.
- 7 Sleep dan vanuit GeluidsRecorder het `aanroep GeluidsRecorder1.Begin` in je "opnemen" knop. Om de stopknop zichtbaar te maken, pak je de `stel in [Knop2].Zichtbaar tot` vanuit **Knop2** en zet er een `waar` blok aan vast.
- 8 Voor de Stopknop voeg je een `stel in [Klok1].Geactiveerd tot met een onwaar` blok uit **Logica** toe. Gebruik het blok `aanroep [Geluidsrecorder1].Stop` (uit **GeluidsRecorder1**) om de opname te stoppen. Nu hoeft je alleen nog het `sluit venster met waarde resultaat` blok te gebruiken met een **Tekst**blok "gewonnen".

- 9 Vanuit de **Klok1** blokken pak je het `wanneer [Klok1].Timer` blok. Hier stop je het blok `sluit venster met waarde resultaat` in met een **Tekst** blok "verloren". Om er zeker van te zijn dat de opname gestopt wordt, sleep je hier ook het `aanroep [GeluidsRecorder1].Stop` blok heen. Je blokken zouden er nu zo uit moeten zien:



- 10 Als je de minigame nu wilt uitproberen voeg je twee `stel in [Label1].Tekst tot` blokken toe. Voeg de **Tekst** "gewonnen" en de **Tekst** "verloren" toe aan de juiste blokken. Zo ziet het er bij ons uit:



- 11 Geweldig! Voor je de minigame uitprobeert moet je nog één ding doen. Ga terug naar **Beginscherm** en klik met rechts op het `sluit venster met waarde resultaat` blok. Klik dan op **Schakel Blok uit**. Programmeurs gebruiken dit vaak om de computer een stukje code te laten negeren.



- 12 Want je wil je minigame tonen. Voeg het blok `open een ander scherm schermNaam` toe met een tekstblok "minigame1".

- 13 Wow! Probeer je spel nu uit! De tekst boven in het scherm zal veranderen en je vertellen of je gewonnen of verloren hebt!

1 Cool, je hebt je eerste minigame gemaakt! Door meer minigames te maken houd je je app interessant voor de spelers. Dus nu gaan we een minigame maken dat gebruik maakt van een **Lijst**.

2 Een **Lijst** is eens tel objecten (variabelen) die je bij elkaar opslaat. Denk bijv. aan een boodschappenlijstje. Daarop staan sinaasappels, appels en druiven. Dit zijn objecten, maar ze staan allemaal in dezelfde lijst omdat ze fruit zijn.

3 Voeg een scherm genaamd "Minigame2" toe. Je kunt de **LijstKiezer** (**Palet > Gebruikersomgeving**) gebruiken om de speler een object in een lijst te laten kiezen. Voeg er één toe aan het scherm.

4 Ga naar de **Blokken** editor. Voeg een `Initialiseer globaal [naam] tot` blok toe en verander "naam" in iets beters: "woordenLijst".

5 Er zit een `maak een lijst` blok onder **Lijsten**; zet het in het scherm. Klik op het symbool



en voeg meer objecten toe aan de lijst (wij hebben er vier, maar je mag het aantal zelf bepalen).

6 Gebruik nu `" "` blokken om de lege plekken te vullen. Zet er woorden in. Wij gebruiken diverse huisdieren. Zo ziet het er bij ons uit:



7 Bij de **LijstKiezer** blokken zit een `wanneer [LijstKiezer1].VoorKiezen` blok en een `stel in [LijstKiezer1].[Elementen]` tot blok. Zet ze er samen in. Nu hoeft je alleen nog maar een `krijg []` **Variabelen** blok erin te zetten en is de **LijstKiezer** klaar! Het zal er zo uitzien:



8 Als de speler een woord kiest, moet je kunnen zien of ze het juiste woord hebben gekozen. Je wilt een `wanneer [LijstKiezer1].NaKiezen` blok hebben om na te kijken wel word de speler koos! Sleep het naar je scherm.

9 Kies een woord uit je lijst (wij kozen Vogel uit de lijst). Ga terug naar Ontwerper, voeg een label toe en verander de tekst naar "Kies" en het woord dat je wilt dat de speler kiest (wij hebben "Kies een huisdier").

10 Aangezien we nakijken *of* de speler het juiste woord kiest, kun je een 'als dan' bepaling gebruiken. Klik op het



symbool en voeg een 'else' (= anders) toe aan de 'als'.

11 Voor als gebruik je een `[] = []` blok uit **Logica**, een **Tekst**blok en het `[LijstKiezer1].[Selectie]` (het spelersblok) uit **LijstKiezer**. Als de speler juist kiest, wint hij; zo niet, dan verliest hij.

12 Zo zou het er nu uit moeten zien:



13 OM te zien of het werkt kun je hetzelfde doen als in de eerste Minigame. Voeg twee `stel in [Label1].[Tekst]` tot blokken toe en zet er `gewonnen` en `verloren` bij.

14 Je moet ook het scherm laten veranderen als er op de "Play" knop gedrukt wordt. Probeer het! Dan kun je je nieuwe minigame spelen.

- 1 De beste manier om deze app te maken is met behulp van een scherm dat alles controleert. Open Screen1.
- 2 Dit scherm heeft ook een **Bestand** nodig, voeg er één toe.
- 3 Ga naar de Blokken editor. Je gaat je eerste procedure maken! Een procedure is een stukje code dat je kunt gebruiken met het "roep aan" blok. Meestal zullen programmeurs deze blokken gebruiken zodat ze niet steeds dezelfde code moeten maken. Deze blokken gaan we gebruiken:



- 4 Klik op **Procedures** aan de linkerkant en sleep het `tot [procedure]` blok in je scherm. Je gaat minigames openen met deze procedure, dus om je te helpen onthouden wat deze procedure doet, noem je het "openMinigame".
- 5 Aangezien je een scherm opent, sleep je het **Controle** blok `open een ander scherm schermNaam` naar het **Procedure** blok.
- 6 Deze procedure gaat voor al jouw minigames werken. Sleep het blok `initialiseer globaal [naam]` uit de **Variabelen** naar het scherm. Verander "naam" naar "aantalMinigames" (zo weet je waar de variabele voor is). Sleep het `[0]` blok erin en verander de 0 in 2 (het aantal minigames dat je tot nu toe hebt) en dan is deze variabele klaar. Het moet er zo uit zien:



- 7 De laatste stap betreft het kiezen van een scherm! Weet je de namen van je eerste twee minigames nog? Minigame1 en Minigame2. Hierdoor kun je willekeurig een scherm kiezen. Door willekeur toe te voegen voelt je app elke keer anders aan.
- 8 Om willekeurig een minigame te kiezen heb je vier blokken nodig: een `samenvoegen` blok uit **Tekst** (dit voegt twee teksten samen), `willekeurig getal tussen [1] en [100]` vanuit **Wiskunde**, een `krijg []` blok uit **Variabelen** en een **Tekst** blok met de tekst "Minigame". Zet ze allemaal als volgt in elkaar:



- 1 Je hebt nu alle onderdelen om de app te laten werken! Met deze onderdelen gaan we de app leuk maken.
- 2 Om controle over de app te houden, hoeft dit scherm alleen maar na te kijken welk scherm net afgesloten is. Dus, sleep het `wanneer [Screen1].AnderSchermGesloten` blok van **Screen1** in je editor.
- 3 Wanneer een minigame afgesloten wordt, dan geeft dat een waarde. Je kunt deze waarde binnen het `wanneer [Screen1].AnderSchermGesloten` blok zetten.
- 4 Om de score van de speler bij te houden, kun je het `initialiseer globaal [score].tot` blok gebruiken (in **Variabelen**). Verander "naam" naar "score" zodat je het makkelijk onthoudt. Sleep het `o` blok uit **Wiskunde** naar het `initialiseer globaal [score] tot` blok. Zo moet het eruit zien:



- 5 Sleep het `als dan` blok met het symbool



naar je editor. Klik op het radertje en voeg twee `anders als` blokken toe zodat het er zo uit ziet:

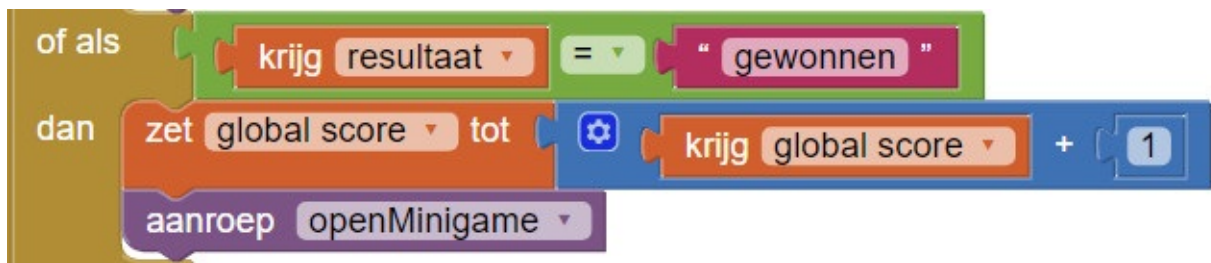


- 6 Het eerste dat nagekeken moet worden is of een minigame of het beginscherm is afgesloten. Gebruik het `[] = []` blok uit **Logica**. Kijk of het `resultaat` overeenkomt met de tekst "Beginscherm".
 - o NB: dit blok heb je aan het begin toegevoegd en toen uitgeschakeld. Schakel het weer aan zodat het beginscherm sluit met het resultaat "Beginscherm".

- 7 Voor het dan gedeelte wil je een minigame spelen, omdat de enige manier waarop een speler van het beginscherm naar **Screen1** komt is door op de Play knop te drukken. Weet je nog de procedure die je gemaakt hebt? openMinigame? Dat kun je nu gebruiken. In **Procedures** zit nu een `aanroep [openMinigame]` blok, gebruik dat! Je kunt ook het `zet [global score] tot` blok op 0 zetten, zodat de score steeds gereset wordt! Zorg ervoor dat alles klopt:



- 8 De twee `of als` zullen nakijken of de speler het laatste spel gewonnen of verloren heeft. Onthoud: als een minigame sluit dan resulteert dat in de **Tekst** "gewonnen" of "verloren". Als ze gewonnen hebben, zou de app 1 moeten optellen bij de score van de speler en een andere minigame moeten openen. Je kunt het `[] = []` blok uit **Wiskunde** gebruiken met de scorevariabele om 1 erbij op te tellen.



- 9 Als ze verloren hebben, slaat het de score op in het bestand en opent het beginscherm. Om de score op te tellen bij het spelersbestand gebruik je het `aanroep [Bestand1].ToevoegenAanBestand` blok (uit **Bestand1**). Als de score is opgeslagen opent het beginscherm. Zo zien de blokken er nu uit:



- 10 Je kunt de app nu uitproberen, maar de score werkt nog niet...dat gaan we op de volgende kaart oplossen!



1 Nu moeten we alleen nog de score van de speler tonen die je games gespeeld heeft!

2 Om de score te tonen moet je een aantal blokken van Screen1 aanpassen.

3 In het `aanroep [Bestand1].ToevoegenAanBestand` blok haal je het `krijg [global score]` blok weg en vervang je het door een **Tekst** `samenvoegen` blok.

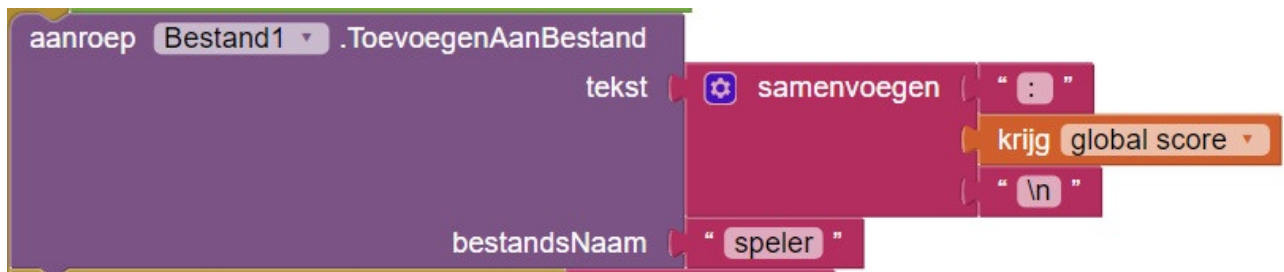
4 Klik op het



symbool en sleep er een `tekst` blok bij.

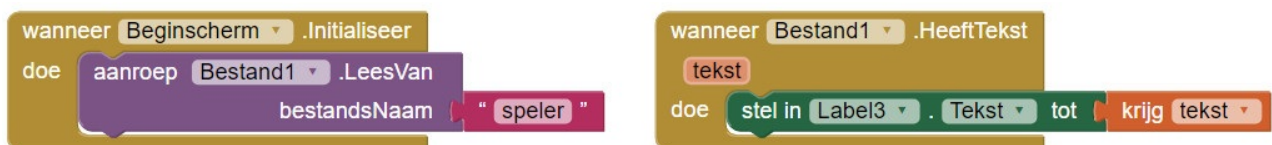
5 Zet het `krijg [global score]` blok in het midden en twee lege **Tekst** blokken bovenin en onderin.

6 Het bovenste **Tekst** blok is om een spatie te plaatsen tussen de naam van de speler en de score. Ik heb `" "` gebruikt. Zet `"\n"` in het laatste **Tekst** blok. `"\n"` is hoe programmeurs aan de computer vertellen om op "Enter" te klikken als het tekst laat zien. Zo ziet het blok er nu uit:



7 In de Ontwerper van het Beginscherm voeg je nog een label toe en haal je daar alle tekst weg. Ga weer naar Blokken, voeg een `wanneer [Beginscherm].Initialiseer` blok toe. Vanuit **Bestand1** sleep het het `aanroep [Bestand1].LeesVan` blok toe en zet je er een **Tekst** blok in. De tekst moet zijn "speler".

8 Goed! Je hebt de tekst gelezen. Nu moet je de tekst laten zien. Sleep het `wanneer [Bestand1].HeeftTekst` blok uit **Bestand1** naar je scherm. Vanuit **Label3** sleep je het `stel in [Label3].Tekst` tot blok daarin en zet je `krijg [tekst]` aan het eind. Zo moeten ze er uit zien:



9 Geweldig! Je game is helemaal klaar om gebruikt te worden! Probeer het uit via de emulator of een Android apparaat.

- 10 Het kan zijn dat je een fout tegenkomt als je een tijdje speelt...de scores verdwijnen van het scherm! Je kunt oude scores weghalen door een "verwijder score" knop aan het beginscherm toe te voegen. Als de speler daarop drukt, wordt het bestand gewist. Als de "verwijder score" knop ingedrukt is, gebruik je het `aanroep [Bestand1].Verwijder` blok onder je **Knop**. Zo ziet het eruit:



- 1 Jij hebt alle controle over de werking van je app! Nu is het heel makkelijk om je eigen minigames toe te voegen. We geven je nog een voorbeeld en daarna ga je zelf aan de slag.
- 2 Maak een "Minigame3" scherm.
- 3 Dit spel laat de speler raden naar een optelsom. Dus je hebt twee labels, één tekstblok en een knop nodig in de **Ontwerper**. Zo ziet die van ons eruit:

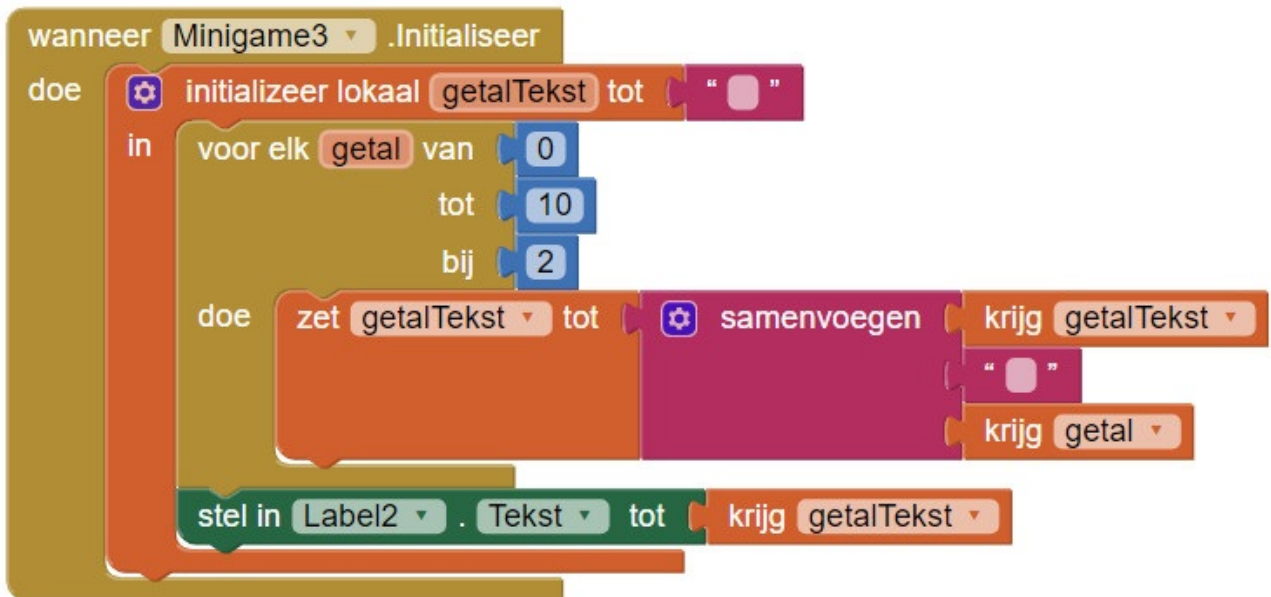


- 4 Ga naar Blokken. Gebruik een **voor elk lus** om het programmeren van het spel makkelijker te maken. Elke 'voor elk' lus herhaalt dezelfde code tot het stopgezet wordt. Dit is handig voor luie programmeurs!
- 5 Gebruik een `wanneer [Minigame3].Initialiseer` blok en zet daar een `Initialiseer lokaal [naam] tot` blok in. Verander "naam" in "getalTekst" en voeg een leeg " " blok toe.
- 6 Voeg nu een `voor elk [getal] van` blok toe (in **Controle**). Dit blok heeft drie delen. Het eerste is het getal waarmee begonnen moet worden; het tweede hoe lang de lus blijft tellen tot het stopt; en de laatste is hoeveel de lus elke keer optelt (dit is wat de speler moet raden). Zet het eerste blok op 0, het tweede op 10, het laatste op 2:



- 7 Binnen de lus kun je nu iets gaafs doen! Gebruik het `zet [getalTekst] tot` blok. Je gaat het zetten op de vorige getalTekst en een spatie (" ") + het getal (dit komt uit de voor lus).

8 Nu moet je tekst van het label naar `krijg [getalTekst]` zetten en moet alles er zo uit zien:



9 Deze stap heb je al eerder gedaan, dus kijk of je begrijpt waarom deze blokken in de code moeten staan:



10 Ga terug naar Screen1 en zorg dat het `Initialiseer globaal [aantalMinigames] tot` blok nu op 3 staat (je hebt nu immers drie spellen!)

11 Zie je hoe makkelijk het was om een spel toe te voegen? Probeer deze ideeën eens uit:

- o Bedenk zelf een spel (dat is altijd leuker).
- o Verander de lijst minigame. Laat het spel willekeurig een object uit de lijst kiezen, zeg de speler erop te klikken en kijk na of de speler de juiste keuze maakte.
- o Probeer andere **Media** opties uit. Je kunt de speler vragen een foto van zichzelf te maken met de camerablokken.
- o Kijk of het de speler lukt om in één seconde vijf keer op een knop te drukken.
- o Probeer van alles uit, gebruik je fantasie en probeer alle blokken!

