

Eindverslag Intervaltrainer

Bernhard Degen
Sander Hendrix
Jolein Kluten

29 juni 2013

Inhoudsopgave

1	Voorwoord	2
2	Beschrijving	3
2.1	Inleiding	3
2.2	Productverantwoording	3
2.3	Specificaties	3
2.3.1	Functionele eigenschappen	3
2.3.2	Niet-functionele eigenschappen	5
3	Ontwerp	6
3.1	Globaal ontwerp	6
3.2	Detailontwerp	7
3.3	Ontwerpverantwoording	7
4	Reflectie	8
4.1	Positieve ervaringen	8
4.1.1	MIDI afspelen	8
4.2	Negatieve ervaringen	8
4.2.1	Progressie weergeven	9

1 Voorwoord

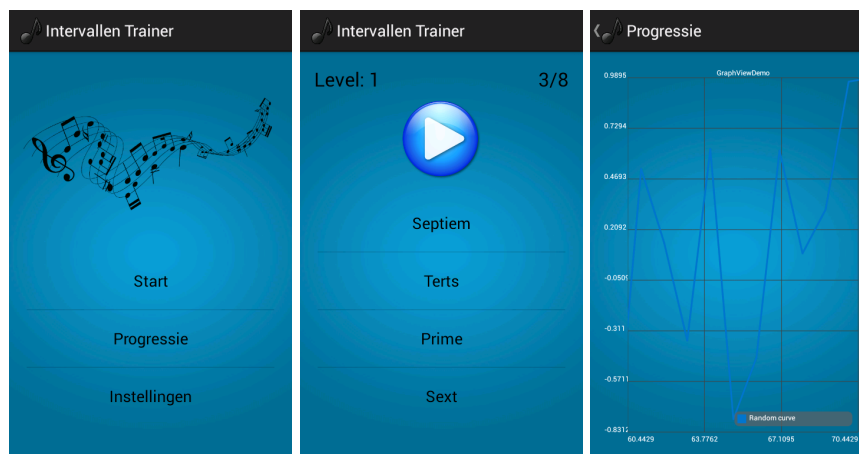
Voor mensen die een muziekinstrument bespelen of op een andere manier met muziek bezig zijn, kan het belangrijk zijn om verschillende intervallen te kunnen herkennen. Een *interval* is in de muziekleer de afstand in toonhoogte tussen twee noten. Het oefenen van intervalherkenning blijkt in de praktijk lastig te zijn. Om dit makkelijker te maken, hebben wij een Android applicatie ontwikkeld.

Dit document is de laatste van een reeks verslagen. Eerst komt een beschrijving van de app aan bod, samen met een verantwoording en specificaties in de vorm van use-cases. Vervolgens wordt er ingegaan op het ontwerp en dit zal ook worden verantwoord. Het laatste onderdeel is de reflectie van het ontwikkeltraject. Twee ervaringen, één positieve en één negatieve, zullen toegelicht worden.

2 Beschrijving

2.1 Inleiding

De app is voornamelijk bedoeld om intervallen te kunnen trainen. Dit wordt gedaan door een interval ten gehore te brengen en een aantal mogelijke antwoorden te tonen. Per interval kiest de gebruiker een antwoord en het zal direct te zien zijn of dat antwoord goed was. Aan het einde van de serie vragen ziet de gebruiker de behaalde score, welke boodschap ook gedeeld kan worden. Als de score voldoende is, bevordert de gebruiker tot een volgend level. Het is de bedoeling dat de gebruiker zijn of haar progressie in een ander scherm kan zien.



Het eerste scherm toont het hoofdmenu, het tweede de weergave tijdens een quiz en het derde een grafiek met daarin de progressie van de gebruiker.

2.2 Productverantwoording

Intervaltrainer is niet de enige app die de gebruiker probeert intervallen bij te brengen. Met dit doel zijn ook andere Android applicaties ontwikkeld. Toch is onze app op een aantal punten verschillend geworden.

De voertaal van onze app is Nederlands. Dit met name handig voor de termen die gebruikt worden om intervallen aan te duiden. Mocht het nodig blijken, dan is het een kleine moeite om de app aan te passen naar een andere taal door het gebruik van een apart bestand voor teksten.

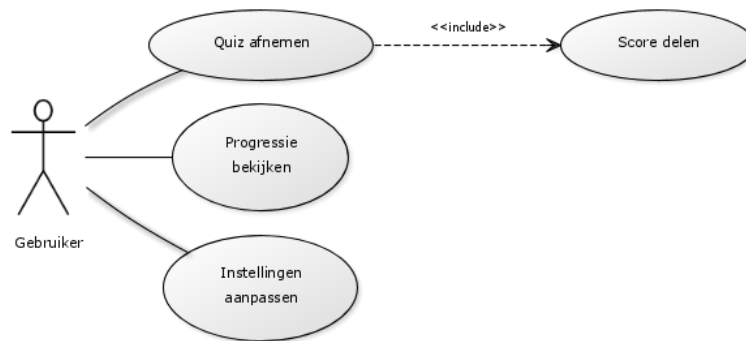
Door de vragen te presenteren in een vorm van een quiz, wordt het voor de gebruiker aantrekkelijker om vaker te oefenen. Waar andere apps vaak een gecompliceerde interface hebben, door bijvoorbeeld een piano te tonen, zijn er bij *Intervaltrainer* ten hoogste vijf knoppen in beeld.

Ten slotte is er de mogelijkheid om de score te delen via alle sociale plugins die op het device, waar de app op draait, geïnstalleerd zijn.

2.3 Specificaties

2.3.1 Functionele eigenschappen

De functionele eigenschappen staan beschreven in een use-case model, gevolgd door een use-case survey.



Quiz afnemen

Een gebruiker start en speelt de quiz.

Progressie bekijken

Een gebruiker kijkt in de grafiek waar zijn/haar progressie in staat afgebeeld.

Score delen

Een gebruiker deelt zijn behaalde score via Facebook of Twitter.

Instellingen aanpassingen

Een gebruiker opent het instellingen-scherm en past daar mogelijk wat instellingen aan.

Eén use-case hiervan, *Quiz afnemen*, is uitgewerkt. Deze use-case is interessant, omdat het de primaire activiteit van de app beschrijft. De uitwerking van deze use-case was dan ook een belangrijke factor bij het implementeren van het uiteindelijke product

Nummer	1
Naam	Quiz afnemen
Korte beschrijving	Een gebruiker start en speelt de quiz.
Primaire actor	Gebruiker
Trigger	Een gebruiker begint met trainen.
Basic flow of events	<ol style="list-style-type: none"> 1. Gebruiker selecteert de optie “Start”. 2. Systeem brengt een interval ten gehore. 3. Systeem toont vier mogelijke antwoorden. 4. Gebruiker selecteert een antwoord. 5. Systeem toont dat het antwoord correct is, door de gekozen knop groen te laten oplichten. 6. Zolang er nog resterende vragen zijn, terug naar stap 2. 7. Systeem toont score. 8. Systeem suggereert om de score te delen. 9. Gebruiker klikt op “Delen”. 10. Verder naar use-case <i>Score delen</i>.
Alternatieve flows	<ol style="list-style-type: none"> 5a. Systeem toont de melding dat het antwoord incorrect is, door de gekozen knop rood te laten oplichten. 6a. Terug naar stap 6.
Precondities	De app wordt niet onderbroken (door bijvoorbeeld een andere app of een laag accuniveau).
Gerelateerde gegevens	-
Aanvullende opmerkingen	Als er niet eerder een level is gekozen, wordt gestart met level 1.
Openstaande punten	-

2.3.2 Niet-functionele eigenschappen

- Levels en teksten zijn makkelijk aanpasbaar
- Schermindeling is geschikt voor zowel telefoons als tablets
- Er is documentatie beschikbaar
- Geringe bestandsgrootte (445.276 bytes)
- Responsief; de reactietijd van een knop is lager tot vergelijkbaar met die bij andere apps

3 Ontwerp

3.1 Globaal ontwerp

Bij het ontwerpen van de applicatie is gekozen om alle onderdelen in te delen in een aantal componenten. Onder de beschrijving van de componenten staat het resulterende klassendiagram.

Quiz

Hoewel het een leerzame app is, betekent dit niet dat het geen leuke app kan zijn. Om het trainen aantrekkelijker te maken, wordt er gebruik gemaakt van activity, *LevelActivity*, in quiz-vorm.

Interval

Het bepalen en afspelen van intervallen is de belangrijkste functionaliteit van de app. De klassen die hierbij horen zijn *DummyLevels* en *Question*.

Weergave van progressie

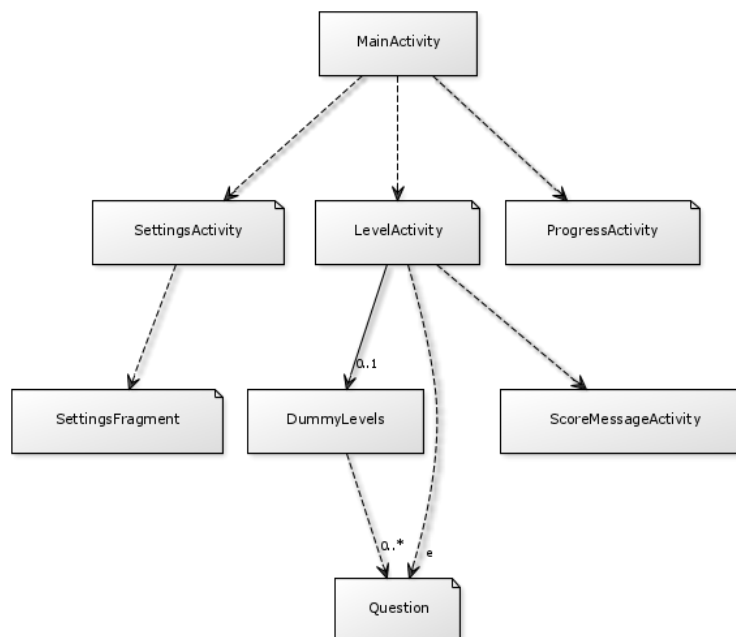
Daarnaast is er een scherm om statistieken van de vooruitgang van de gebruiker weer te geven. Hiertoe is *ProgressActivity* ingericht.

Gegevensopslag

De statistieken en scores worden lokaal opgeslagen. Dit component bestaat uit de klassen *SettingsActivity* en *SettingsFragment*.

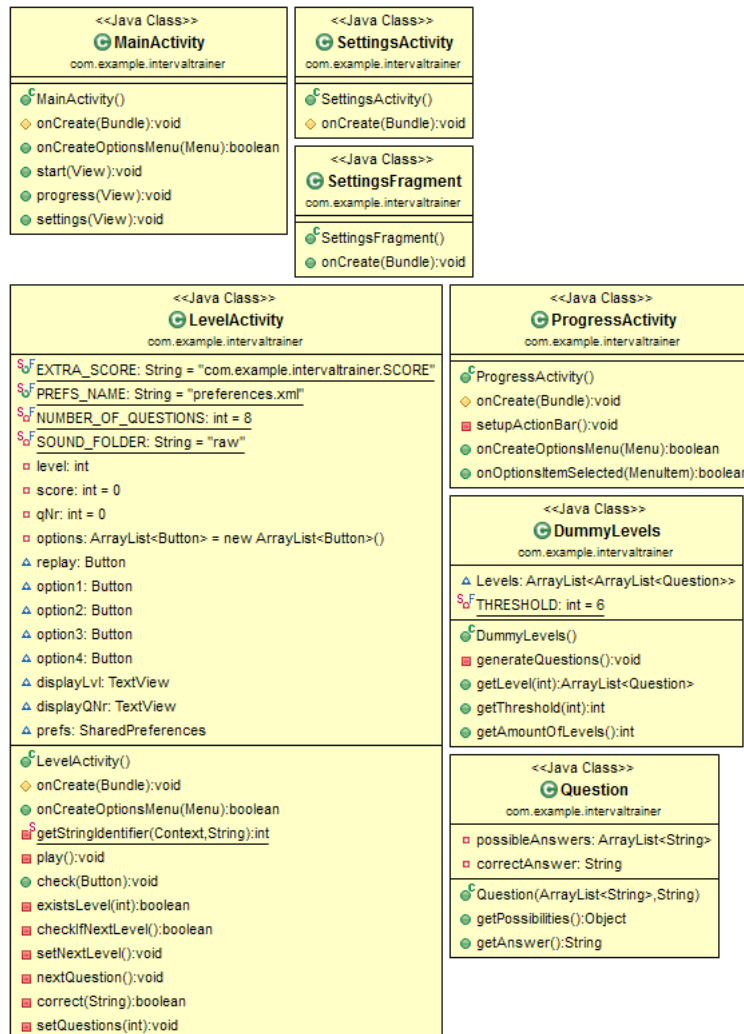
Sociale plugins

Doordat de app op de telefoon of tablet van de gebruiker staat, is het een kleine stap om andere accounts van de gebruiker te raadplegen. Door gebruik te maken van de methode “share” in *ScoreMessageActivity* kunnen scores gemakkelijk worden gedeeld.



3.2 Detailontwerp

Een meer gedetailleerde weergave van bovenstaand klassendiagram:



3.3 Ontwerpverantwoording

Bij het ontwerpen van de applicatie hebben we ervoor gekozen om voor alle use-cases een aparte activity aan te maken. Dit is handig omdat in Android bij iedere activiteiten een user interface gedefinieerd wordt en de gebruiker bij verschillende use-cases op een verschillende manier interacteert met het systeem.

Voorts hebben we bij het ontwerp de MVC-architectuur (Model-View-Controller) in acht genomen en waar mogelijk gebruik gemaakt van bestaande *design patterns* in *Java*.

De resultaten van het eerste gebruikersonderzoek, zijn in sectie 4 ondergebracht.

4 Reflectie

4.1 Positieve ervaringen

Over de volgende punten zijn wij tevreden:

- De implementatie van MIDI
- De quiz heeft meerdere levels en de mogelijkheid dit verder uit te breiden
- Het gemak van user interfaces beschrijven met XML
- Het leren werken met L^AT_EX
- Het ontwikkelen van een Android app met behulp van de SDK
- Het ophalen van teksten in een apart bestand
- Opties opslaan met behulp van `SharedPreferences`

4.1.1 MIDI afspelen

MIDI is een acroniem voor "Musical Instrument Digital Interface". Het is een protocol voor muzikale informatie. MIDI werkt met commando's, waarin verschillende informatie voor het af te spelen geluid staat. Denk aan het instrument, de toonsoort, de toonhoogte en variaties hierop. Het kan gezien worden als een soort elektronisch 'notenschrift', dat door een apparaat genterpreteerd word en omgezet naar tonen.

1. Oorspronkelijk was het idee deze commandos dynamisch samen te stellen en vervolgens te genereren en meteen af te spelen. Dit bleek echter te hoog gegrepen te zijn.
2. Het alternatief was echter ook niet aantrekkelijk. Dit zou namelijk het opnemen van geluidsfragmenten zijn geweest, wat weer allerlei problemen met zich mee zou brengen. Niet de minste is een enorme toename in bestandsgrootte van de gehele app.
3. We hebben gexperimenteerd en zijn uiteindelijk bij een tussenvariant uitgekomen. We werken niet met dynamische commando's, maar met vooraf aangemaakte commando's. MIDI-commando's kunnen namelijk ook worden opgeslagen als bestanden. In deze zogenaamde standaard MIDI-bestanden (met de extensie .MID) kunnen de commandos worden vastgelegd. We hebben deze MIDI-files aangemaakt met een extern programma, om ze vervolgens aan de app toe te voegen. Het voordeel is dat deze bestanden zeer klein zijn, ze bevatten immers geen geluidsfragment maar een commando.

4.2 Negatieve ervaringen

Hierover zijn we minder tevreden:

- De vragen in de quiz zijn niet willekeurig
- Er zijn geen vector-afbeeldingen gebruikt
- Het bijhouden van progressie werkt (nog) niet

4.2.1 Progressie weergeven

Het opslaan en weergeven van de progressie van de gebruiker is niet meer gelukt. Het gebrek aan tijd en het niet-werkend hebben van een `SQLite`-database zijn hiervan de oorzaken. De grafiek is wel geïmplementeerd, zoals te zien op de screenshots van de user interface. Het opzetten van een interactieve grafiek hebben we aangepakt met de theoretische onderzoeksmethode.

1. We hebben gekeken naar een aantal bestaande oplossingen.
2. Daarna hebben we een aantal criteria waar de grafiek aan moest voldoen opgesteld:
 - De grootte van de grafiek moet tijdens run-time aan te passen zijn
 - Er moet de mogelijkheid zijn om een veld (op een leeg stuk van het vlak) weer te geven, zoals een legenda
 - De grafiek moet interactief zijn; aan de hand van het datapunt waarop de gebruiker tikt, moet de invulling van het veld kunnen worden bepaald
 - De afstand tussen twee lijnen in het rooster moet in te stellen zijn
3. Punt 1 en 4 zijn te verklaren met de eis dat de applicatie geschikt moet zijn voor zowel telefoons als tablets en dat de gebruiker tijdens het gebruik de mogelijkheid moet hebben om van scherm-oriëntatie te veranderen. De rationale achter punt 2 en 3 is dat het voor de gebruiker interessanter is om ook gedetailleerde informatie over een bepaalde score of dag te kunnen bekijken; de meest gebruiksvriendelijke manier hiervoor leek ons om dit te integreren in de grafiek.
4. Uit de mogelijke oplossingen hebben we *GraphView* gekozen. Deze API voldoet aan alle eisen en oogde het meest gemakkelijk en flexibel in gebruik. Daarnaast heeft het een aantal extra functies die mogelijk van pas zouden komen, zoals de ondersteuning van gebaren voor het scrollen over en zoomen van de grafiek.