

Tip: Tippity Tap

PWA

Wietse Kuipers (s4317904)
Willem Boumans (s4337166)
Bart Timmermans (s4375785)

27 juni 2014

Voorwoord

Dit is het eindverslag voor de eerstejaars cursus Research & Development. Het doel van deze cursus was om kennis te maken met de ontwikkeling van Android-applicaties en het projectmatig werken met een team om dit te bereiken. Allereerst hebben we, door middel van een practicumopdracht waarbij we een puzzel-applicatie moesten ontwikkelen, kennisgemaakt met Android en de ontwikkeltools. De tweede applicatie die we gemaakt hebben is *Tippity Tap*, een ritmespelletje, deze applicatie is tevens ook ons eindproduct voor deze cursus.

Het verslag is onderverdeeld in drie onderdelen. In de *beschrijving* wordt beschreven hoe onze applicatie eruit ziet en worden de eigenschappen bekeken die onze applicatie heeft. Bij het onderdeel *ontwerp* wordt het ontwikkelingsproces beschreven. Tevens bekijken we de gemaakte keuzes tijdens het ontwikkelen en worden de aparte onderdelen gedetailleerd bekeken. Tot slot kijken we bij het onderdeel *reflectie* terug op de opdracht, hoe het allemaal is verlopen en wat we de volgende keer kunnen we verbeteren.

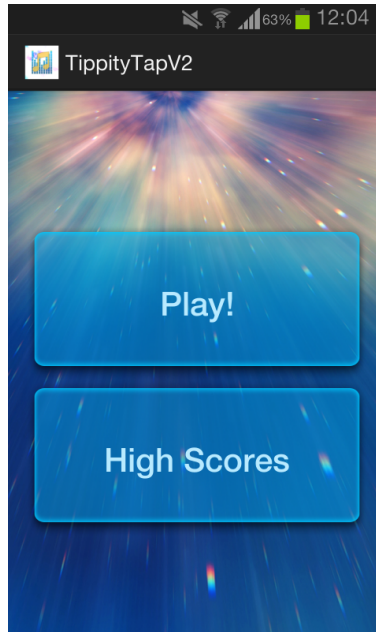
Inhoudsopgave

1	Beschrijving	2
1.1	Inleiding	2
1.2	Productverantwoording	4
1.3	Specificaties	4
1.3.1	Functionele eigenschappen	4
1.3.2	Niet-functionele eigenschappen	6
2	Ontwerp	6
2.1	Globaal ontwerp	6
2.2	Detailontwerp	7
2.2.1	MainMenu	7
2.2.2	Game	7
2.2.3	TipMap	8
2.2.4	Indicator	8
2.2.5	HighScores	9
2.2.6	HighScore	9
2.2.7	HighScoresContainer	9
2.3	Ontwerpverantwoording	9
3	Reflectie	11
3.1	Conclusie	12

1 Beschrijving

1.1 Inleiding

Zodra onze applicatie gestart wordt, komt men in het hoofdmenu terecht. Dit ziet er als volgt uit: (figuur 1)



Figuur 1: Screenshot van het hoofdmenu.

In het hoofdmenu staan twee knoppen, respectievelijk “Play!” en “Highscores”.

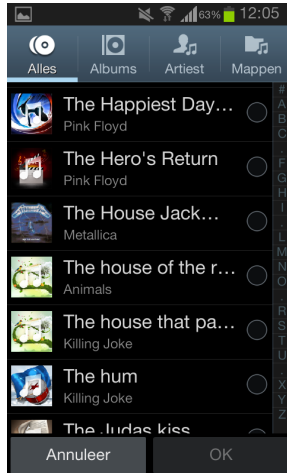
Wanneer men op “Play!” drukt, opent de applicatie de Android-mediaspeler. Hiermee kan een liedje gekozen worden dat al op het device van de gebruiker staat. Dit is waarin onze applicatie zich onderscheid van de applicaties die al te vinden zijn in de Play Store. Deze gebruiken namelijk allemaal vooraf geselecteerde liedjes. Bij onze applicatie heb je een oneindige hoeveelheid liedjes waarmee je kunt spelen, je kunt ze immers uit je eigen collectie kiezen.

Wanneer er vervolgens op “Ok” gedrukt wordt, zijn er twee scenario’s:

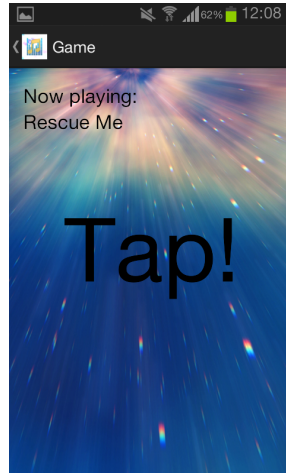
Ten eerste: er is geen TipMap voor het liedje gemaakt.

In dit scenario zal de gebruiker eerst gevraagd worden om een TipMap te maken voor het nieuwe liedje, natuurlijk geeft de applicatie ook instructies over het maken van een TipMap. De precieze werking van de TipMaps wordt verder toegelicht bij *ontwerp*.

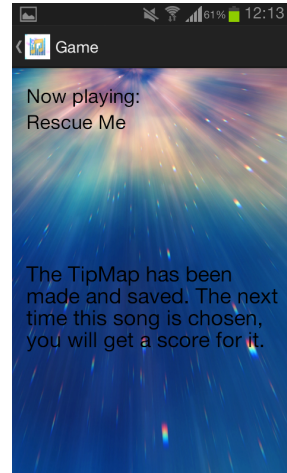
Screenshots voor als er geen TipMap voor het liedje gemaakt is.



Figuur 2: Een liedje kiezen.



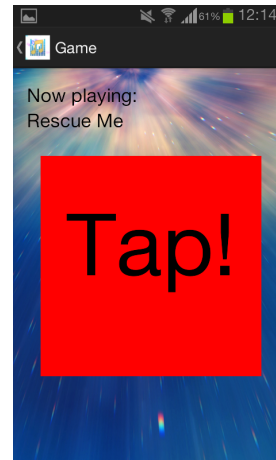
Figuur 3: Maken van een TipMap.



Figuur 4: Na een TipMap gemaakt is.

Ten tweede: er is al een TipMap aanwezig voor het gekozen liedje.

Als er al een TipMap aanwezig is voor het liedje, is het de bedoeling dat de gebruiker zo dicht mogelijk bij het oorspronkelijk ingegeven ritme komt. Hierbij krijgt de gebruiker een visuele hint in de vorm van een knop die van rood naar groen verandert op het moment dat er getapt moet worden. De gebruiker herhaalt dit totdat het liedje af is. Als het liedje is afgelopen krijgt de gebruiker zijn/haar behaalde score te zien, deze wordt ook opgeslagen in het "Highscore"-menu. Deze highscores houden de applicatie ook leuk. De gebruiker blijft geboeid doordat hij constant kan proberen een betere highscore te behalen.



Wanneer men vanuit het menu naar het venster met "Highscores" gaat, dan krijgt men een lijst te zien met scores die de speler tot nu toe behaald heeft. Voor ieder liedje wordt een aparte score bijgehouden. In het highscores-venster staat dus het tuple <Liedje, score>.

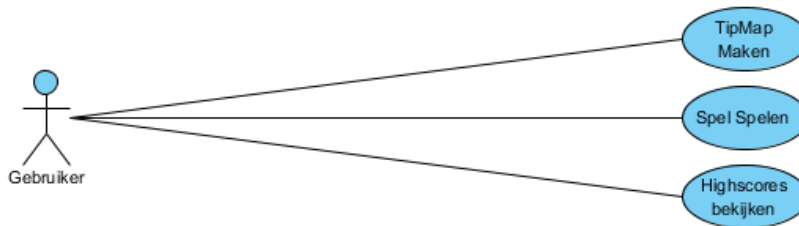
1.2 Productverantwoording

Er bestaan al legio ritmespelletjes. Denk aan Guitar Hero, Rock Band, Audiosurf en TapTap. Deze spellen zijn grotendeels voor de pc/spelcomputer gemaakt. TapTap onder andere heeft wel een Android-versie. Er bestaan dus al ritmespelletjes voor Android, die gebruik maken van muziek. Deze muziek is alleen wel altijd vooraf bepaald. Je moet het doen met de liedjes die de developers meegeleverd hebben. Er is geen mogelijkheid om je eigen muziek te gebruiken voor het spel. Onze applicatie kan wel gebruik maken van de muziek van de gebruiker. Op deze manier heeft de gebruiker een eindeloos aantal liedjes tot zijn beschikking. Tippity Tap onderscheidt zich ook van de al bestaande applicaties door het TipMap-systeem. Bij de andere spellen/applicaties moet je altijd een vooraf bepaald ritme volgen. Tippity Tap maakt het mogelijk om je eigen ritme in te geven bij ieder liedje. Dit geeft de gebruiker nog meer vrijheid. Het maken van de applicatie was zeker de moeite waard, omdat Tippity Tap het grootste kritiekpunt op de bestaande spelletjes aanpakt, namelijk het zeer beperkte aantal liedjes dat je kunt spelen.

1.3 Specificaties

1.3.1 Functionele eigenschappen

Tippity Tap is een applicatie die ontwikkeld is om functioneel, doch aantrekkelijk te zijn. Omdat het een spelletje zonder ingewikkelde optiemenu's is, zijn er niet veel use cases te verzinnen bij deze applicatie. Dit kan men terugvinden in ons use case model (figuur 5), waarin slechts drie use cases staan: het maken van een TipMap, een spelletje spelen en de highscores bekijken. Twee van deze use cases zullen hieronder helemaal uitgewerkt worden, om een beeld van de functionele eigenschappen te creëren. De niet-functionele eigenschappen zullen als lijst gegeven worden.



Figuur 5: Use case model.

Use case	Spel spelen
Beschrijving	In deze use case wordt het spelen van het spel behandeld.
Actoren	Degene die het spel Tippiy Tap speelt, hierna gebruiker genoemd.
Trigger	De gebruiker selecteert de “Play!”-knop in het hoofdmenu en kiest een liedje uit.
Basic course of events	<ol style="list-style-type: none"> 1 Het systeem laadt het liedje met bijbehorende TipMap en speelt de muziek af. 2 De gebruiker drukt op de “Tap!”-knop. 3 .1 Als op dit moment de “Tap!”-knop groen is, wordt de score verhoogd. 3 .2 Als op dit moment de “Tap!”-knop rood is, wordt de score verlaagd. 4 Als het liedje is afgelopen laat het systeem de score van de gebruiker zien, en slaat deze op in een extern bestand.
Alternate paths	-
Exception paths	<p>E1 De gebruiker gaat terug naar het hoofdmenu met de terugknop.</p> <p>E2 Het systeem laat het hoofdmenu zien, de score is niet opgeslagen.</p>
Assumpties	De gebruiker gebruikt een Android device met een compatible API-level.
Precondities	Er is een TipMap gemaakt voor het liedje dat de gebruiker heeft gekozen.
Postcondities	-

Use case	Highscores bekijken
Beschrijving	In deze use case wordt het bekijken van de lijst met highscores behandeld.
Actoren	Degene die het spel Tippity Tap speelt, hierna gebruiker genoemd.
Trigger	De gebruiker selecteert de “Highscores”-knop in het hoofdmenu.
Basic course of events	<ol style="list-style-type: none"> 1 Het systeem laadt het highscores-bestand in en laat dit aan de gebruiker zien. 2 De gebruiker bekijkt de highscores.
Alternate paths	-
Exception paths	<p>E1 De gebruiker gaat terug naar het hoofdmenu met de terugknop.</p> <p>E2 Het systeem laat het hoofdmenu zien, de score is niet opgeslagen.</p>
Assumpties	De gebruiker gebruikt een Android device met een compatible API-level.
Precondities	-
Postcondities	-

1.3.2 Niet-functionele eigenschappen

Beschikbaarheid Tippity Tap maakt geen gebruik van enige vorm van (draadloze) communicatie en is daardoor altijd beschikbaar voor de gebruiker.

Error tolerantie Onze applicatie is momenteel zo geprogrammeerd, dat alle mogelijke errors opgevangen worden. Het zal dus niet voorkomen dat de gebruiker een error-bericht te zien krijgt.

Usability Tippity Tap is opgemaakt met grote, duidelijke knoppen en heeft geen menu's met vele opties, waardoor is de usability erg goed.

Compatability Momenteel is Tippity Tap compatibel met alle Androidversies >Android 4.0. Het draait op alle devices die deze versie, of een hogere, als besturingssysteem gebruiken.

2 Ontwerp

2.1 Globaal ontwerp

De applicatie kan grofweg worden opgedeeld in twee stukken: Het spelen van het spel en het uitlezen van je highscores. Het kunnen spelen en bijhouden van behaalde resultaten zijn de twee belangrijkste aspecten van een spel en onze twee componenten zorgen hiervoor. De rol van het spel spreekt al voor zich, de

rol is namelijk het spel zelf. Het spel houdt tijdens het spelen ook de scores bij, deze kunnen daarna worden uitgelezen met de highscore functionaliteit. Beide functionaliteiten zijn bereikbaar via het hoofdmenu, hetgeen zichtbaar is zodra de applicatie opgestart wordt.

2.2 Detailontwerp

De applicatie is opgedeeld in een aantal klassen, de belangrijkste klassen hiervan zijn als volgt:

- MainMenu
- Game
- TipMap
- Indicator
- HighScores
- HighScore
- HighScoresContainer

De overige klassen zijn ontworpen als ondersteuning voor deze klassen. De bovengenoemde klassen worden hieronder nog in detail behandeld.

2.2.1 MainMenu

Deze klasse heeft niet veel uitleg nodig, dit is het hoofdmenu. Via deze activity kun je naar de andere twee activities, namelijk het spel en de highscores. Buiten dat heeft het hoofdmenu geen functionaliteit.

2.2.2 Game

Dit is die activity waarin het spel zich afspeelt, hij laat het spel lopen. Deze klasse is afhankelijk van de TipMap klasse en heeft dus ook een TipMap als attribuut. Aan het einde van het spel heeft deze klasse ook de HighScore klasse nodig, de behaalde score moet namelijk doorgegeven worden. De meeste methoden van Game reageren op Android events zoals aanrakingen. Alle listeners die voor het spel relevant zijn zitten in deze klasse. De code wordt hierdoor wat rommeliger, maar alle evenementen die voor het spel belangrijk zijn zitten nu wel in n klasse, wat de overzichtelijkheid weer verbeterd. De methoden bestaan dan ook uit een hoop listeners en een priv methode die een highscore toevoegt. Deze is apart gemaakt zodat de functies niet te lang worden.

2.2.3 TipMap

Een klasse die de titel en het ritme van een liedje opslaat. Het ritme wordt opgeslagen als een lijst van tijdstippen waarop er op de knop is gedrukt. Dat wil zeggen dat het ritme zelf aangegeven moet worden door op de knop te drukken en dat deze vervolgens als lijst van tijdstippen wordt opgeslagen. Verder houdt de TipMap ook nog bij of een TipMap gemaakt wordt en kan de TipMap kijken of een gegeven tijdstip in zijn eigen lijst van tijdstippen zit. Deze klasse is serializable.

De klasse bevat de volgende methodes:

- **public TipMap(String song_name)**
Maakt een TipMap aan met de ingevoerde naam.
- **public boolean contains_tip(long tip)**
Kijkt of een tijdstip in zijn lijst van tijdstippen staat. Er wordt gecontroleerd binnen een interval van +- 100 ms om het spel niet te moeilijk te maken. Het tijdstip waarop oorspronkelijk gedrukt was bij het maken van de TipMap hoeft dus nu niet exact gelijk te zijn.
- **public int max_score()**
Geeft de hoogst haalbare score.
- **public boolean isBeing_created()**
Checkt of de TipMap nu gemaakt wordt.
- **public void addTip(Long time)**
Voegt een tijdstip toe aan zijn lijst.
- **public LinkedList<Long>getTaps()**
Geeft de lijst van tijdstippen.
- **public void finished()**
Zet de boolean being_created op false.
- **public String getSong_name()**
Geef de naam van het liedje waarvan deze TipMap is.

2.2.4 Indicator

Dit is een extension van de SurfaceView klasse. Deze klasse wordt gebruikt om de gebruiker te laten zien wanneer hij/zij tijdens het spel moet drukken. Heeft een TipMap nodig en maakt zijn eigen IndicatorThread aan om zichzelf daarmee te besturen.

De belangrijkste methodes van de klasse:

- **public void setTipMap(TipMap tipMap)**
Geef de TipMap aan waarop de indicator zijn indicaties moet baseren.

- **public void start(long start)**
Geef de tijd op waarop het spel is gestart. Belangrijk voor synchronisatie.
- **public void doDraw(Canvas canvas)**
Methode die de indicator tekent. Wordt constant aangeroepen door de IndicatorThread.

2.2.5 HighScores

De activity die de highscores laat zien. Is vanzelfsprekend afhankelijk van de HighScoresContainer klassen daarmee ook van de HighScore klasse. Heeft verder geen bijzondere methoden of attributen.

2.2.6 HighScore

Een simpele klasse die een titel, score en maximale score bijhoudt. De klasse HighScoresContainer is niets meer dan een serializable lijst van HighScore objecten.

De belangrijke methodes:

- **public HighScore(String song_name, int score, int max_score)**
Maakt een nieuwe HighScore met de naam van het liedje, de score en de maximale score.
- **public String toString()**
Geeft een String representatie van de Highscore die direct in de lijst van highscores kan.
- **public int getScore()**
Haalt de score op.
- **public String getSong_name()**
Haalt de titel van het liedje op.

2.2.7 HighScoresContainer

Een serializable klasse die een lijst van alle highscores bijhoudt. Hoewel tegen OO principes is er hier voor gekozen om de lijst van HighScores publiek te maken, omdat deze vaak aangepast en uitgelezen moet worden en de code daarmee sterk vereenvoudigt.

2.3 Ontwerpverantwoording

Ontwerpbeslissing 1: de TipMap

Vanwege de uitwerking die we hebben gekozen (gebruiker geeft zelf een ritme op voor het liedje) was er een klasse nodig die dit ritme opslaat. Aangezien het ritme te maken heeft met tijd, was het meteen logisch om het ritme op te

slaan als een lijst van relatieve tijdstippen (het begin van het liedje is tijdstip 0).

De interne klok van Android werkt met milliseconden met als datatype Long, dus het was makkelijk om Longs te gebruiken om omzetswerk te voorkomen. De methoden van de klasse zijn nogal logische getters en setters, behalve de constructor en `contains.tip(long tip)`. `contains.tip` is een essentiële functie voor het spel, aangezien deze voor de puntentelling zorgt.

Er was echter nog een probleem, op de milliseconde precies een druk herhalen is vrijwel onmogelijk, dus er moest met een interval gewerkt worden. Door middel van trial-and-error is er uiteindelijk voor een interval van 200ms gekozen. 200ms is groot genoeg om punten te kunnen scoren, maar ook kort genoeg om te voorkomen dat willekeurig drukken ook succesvol is. Door het gebruik van het interval en het werken in milliseconden, worden erg snelle liedjes soms wel erg moeilijk of makkelijk, bijvoorbeeld doordat je zo snel na elkaar tikt dat het interval elkaar overlapt. Helaas is dit ook moeilijk te vermijden.

Ontwerpbeslissing 2: de Indicator

Om het spel op een speelbaar niveau te maken was er nog een element noodzakelijk, dat er voor zorgt dat het zichtbaar is wanneer er op de knop gedrukt moet worden. Het onthouden van een eerder ingegeven ritme is zorgt namelijk voor een te moeilijk spel.

Aangezien er met een interval van 200ms wordt gewerkt is het belangrijk dat de indicator snel kon veranderen, zonder dat de app te zwaar belast wordt. Om dit te realiseren hebben we een tip van de studentassistenten gebruikt, namelijk dat een SurfaceView hier erg geschikt voor is. De klasse Indicator is dan ook een uitbreiding op de SurfaceView van Android. Vermoedelijk had het gebruik van een losstaand Canvas de app te traag gemaakt, maar met een canvas binnen een SurfaceView is het geen enkel probleem.

Om de Indicator aan te sturen krijgt deze eerst de TipMap (dus tijdstippen waarop getapt moet worden), waarna een thread wordt gestart om constant het Canvas opnieuw te tekenen en te checken of de speler moet gaan tappen. De speler ziet 100ms van tevoren de knop oplichten als teken dat hij/zij moet gaan tappen. Deze tijd is opzettelijk voor het moment gekozen waarop gedrukt moet worden, dit zorgt ervoor dat er ook met de reactietijd van de speler rekening wordt gehouden.

3 Reflectie

We zijn erg tevreden met het eindresultaat, alhoewel we, net als de meeste groepen, in het begin nog moeite hadden met Android, hebben we toch een goed functionerende app geproduceerd. Het feit dat de Setting Sun applicatie ook al een spel was heeft wel geholpen bij het ontwikkelen van deze applicatie. Zo hoefde je bijvoorbeeld de highscores en het canvas niet helemaal opnieuw te gaan ontdekken.

In de door ons ontwikkelde app is ook alles aanwezig wat we oorspronkelijk wilde. Zo kan je je eigen muziek kiezen en een ritmisch spelletje spelen. Hoewel we oorspronkelijk liever een oplossing hadden gevonden waarbij het door de gebruiker gekozen liedje automatisch geanalyseerd en omgezet kon worden naar een TipMap, vinden we dat de oplossing om zelf een TipMap te maken een goed alternatief is.

Ook hebben we bij deze applicatie, in tegenstelling tot de Setting Sun applicatie, gebruik gemaakt van de combinatie Android Studio en GitHub. Android Studio is ontwikkeld door Google en hier hadden we persoonlijk een betere ervaring mee dan met Eclipse. GitHub is vooral onze samenwerking ten goede gekomen, hierdoor kon iedereen ten alle tijde bij de laatste versie en was het voor iedereen mogelijk om thuis ook te werken. Hierdoor hoefde we niet heel vaak af te spreken om toch gecoördineerd te kunnen werken.

De negatieve punten waren bijvoorbeeld dat je soms geen goede oplossing door Android al gegeven kreeg. Zo hebben we bijvoorbeeld al relatief veel moeite gehad met de muziek. Android heeft weliswaar al een oplossing hier zelf voor, maar achteraf bleek dan bijvoorbeeld weer dat hij niet op ieder merk de oplossing fatsoenlijk implementeerde.

Op technisch aspect heb je altijd nog enkele mankementen waar je tijdens het programmeren tegenaan loopt, zeer grote problemen, die zich niet snel op lieten lossen, (buiten het analyseren van de muziek, dat we met het maken van TipMaps hebben opgelost) zijn we niet tegengekomen.

Natuurlijk zijn er nog sommige punten van bijvoorbeeld het design waar we, als we meer tijd hadden gehad, graag nog even aan hadden gesleuteld. Dit zijn vrijwel alleen visuele punten, zo kon bijvoorbeeld de “Tap!”-knop nog wat mooier opgemaakt worden of de highscores wat mooier worden weergegeven.

3.1 Conclusie

Al met al zijn we tot een goed resultaat gekomen en zijn we allemaal tevreden over de app. Alhoewel er nog sommige verbeteringspunten blijven, zijn we van mening dat app goed functioneert. De samenwerking is goed verlopen, dit onder andere door de taakverdeling waarbij iedereen vooral aan een onderdeel werkte dat hem goed lag. Projecten in de toekomst met Android zullen ook zeker een stuk gemakkelijker verlopen door de ervaring die we hebben opgedaan bij deze cursus. Het gebruik van de tools zoals GitHub en Android Studio is ook uitstekend bevallen en deze zullen we in de toekomst ook vaker gaan gebruiken bij projecten zoals deze. De werkwijze zal dus, doordat deze goed is bevallen, niet veel verschillen bij andere projecten.