KU LEUVEN

FACULTEIT INGENIEURSWETENSCHAPPEN

NoiseMApp: Generating Noise Level Maps through Gamification

Philippe De Croock

Thesis submitted for the degree of Master of Science in Engineering: Computer Science, major Distributed Systems

Thesis supervisor:
Prof. dr. ir. Erik Duval

Assessors:
Frans Van Asche
Prof. dr. Danny Hughes

Mentors:
Gonzalo Alberto Parra Chico
Jose Luis Santos Odriozola

Academic year 2012 – 2013

Preface

So, this is it... my Master Thesis. The crown to years of hard work. I'm proud to show you what it has been worth. Of course, this would not have been possible without special people that helped me through this final year.

First, I'd like to say thanks to my promotor professor Erik Duval. He made sure I could choose a subject that I found interesting from the beginning and work it out in my own way.
My assistants Gonzalo Alberto Parra Chico and Jose Luis Santos Odriozola were the best. They were truly interested in what I did and gave me meaningful advice what motivated me to do better.
I'd also like to thank the assessors Frank Van Asche and professor Danny Hughes, who took their time to read my thesis and ask critical questions. Also to colleagues and other people of the department I would like to say thanks for sharing their visions and interpretations.
Special thanks go to friends who stood by me during my thesis and my whole academic life. Katrien and Hendrik are irreplaceable, two of my best friends who were always there for me. Especially their cooking skills have tasted very well while my partner was on Erasmus! Kristof, my buddy at the department who always stood by me with advice, interesting and less interesting discussions in the 'Oase'. Fran, who I know from the beginning at the university and with whom I have shared uncountable hours of team assignments with 'Stacey's Mom' playing in the background. Together we learned what it was to be a programmer and how crazy all-nighters could be.
Gratitude is not enough for my parents. They made all of this possible without asking me for something in return. Although they didn't always understand what I was saying, they were very interested in what I did and will do in the future. Their infinite support, financially and socially, are just some of the things I will always cherish.
And finally, my loving partner Thomas. I wouldn't be where I am now without him. He was always there for me when I needed it. Even while he was on Erasmus in Valencia, we managed to pull through. And still, since August 2006, he makes my world a better place and I wouldn't want to imagine a world without him. I love you.

To all of you, I say "Thank you."

Philippe De Croock

Contents

Abstract

Noise level maps are maps on which noise levels are represented by colours, ranging from green (still) to red (very loud). They are often used to check the noise pollution in order to make environmental decisions. Generating such noise level maps is expensive and labour-intensive. The devices needed to measure sound levels require a large budget.

Today, smart phones exist with lots of sensors: light sensors, accelerometers, gps, WiFi receivers, cameras and microphones. The microphones can be used to record sounds or measure sound levels. Noise mapping applications have already been developed for smart phones on many platforms. Yet, they often lack the engagement of users to go out and record the decibel level regularly at certain locations. This makes it difficult to generate sufficient amounts of up to date data.

This thesis presents a way to make these sorts of applications more fun in order for users to use the application more often than they would otherwise. For this, gamification is applied, which is defined as adding game elements and game techniques to a non-game context. NoiseMApp is the result of adding elements like points, badges and a leaderboard and adding techniques like the Random Record, Sound Battle, Noise Hunt and Sound Checkin to the process of mapping noise levels with a smart phone. The Android application is available on Google Play.

The application has been designed through an iterative process of Rapid Prototyping, which resulted in a paper prototype, two digital prototypes and a final application that all have been evaluated. In order to evaluate the NoiseMApp application, a similar non-gamified application has also been developed, which was tested by an other test population. The NoiseMApp application successfully generated more recordings per user than the non-gamified application and was also perceived as more fun. Gamers and users who do not use application with points, badges and leaderboards tend to be more influenced by gamification.

List of Figures

List of Tables

List of Abbreviations and Symbols

Abbreviations

PBL	Points, Badges and Leaderboards
POI	Points Of Interest

Chapter 1

Introduction

1.1 Problem Statement

1.1.1 Leuven

Leuven is a city in Belgium with 30.847 inhabitants [29]. It is most known for its students, of which more than 43.000 have signed up at KU Leuven anno 2013. Many students studying in Leuven also live there during the week. Numbers are estimated at more than 35.000 students[19] that inhabit a dorm in Leuven.

1.1.2 Noise Pollution

Noise nuisance is considered one of the most hazardous pollution problems of the environment, according to the World Health Organization [39]. Causes of noise pollution can be busy traffic, construction works, music bars, but also groups of people on the streets. In urban environments it can be damaging to human health, well-being and productivity [8]. Also the behaviour of animals can be influenced for the worse. Since January 1st, 2013, new rules are imposed in Belgium concerning noise levels, as proposed by the Flemish Minister for Environment, Nature and Culture Joke Schauvliege [11].

In Leuven, noise nuisance is a hot topic. Even more stringent rules are approved for owners of bars [5]. They all need to have a decibel meter in their establishment that automatically sends readings to the police office. When the sound level standards are exceeded, the bar owners can be fined severely.

Furthermore, it is argued that a big part of noise nuisance in Leuven is due to the students living there [23]. Firstly, there is an immediate consequence of the decibel level of music played on parties or in *fakbars*, which are student bars run by students. For this, the same rules of Minister Schauvliege apply. Secondly, students often hang out on the streets near the *fakbars* or pubs, which causes a lot of rumour after 10 p.m. In Leuven, near Tiensestraat and Oude Markt, people complain about

the noise the most [21].

Therefore, since the start of the academic year 2012-2013, extra measures are taken against noise nuisance in Leuven. These measures include obligations for fakbars to have a curfew at 5 a.m. and for students to show their student id when entering a fakbar. Next to this, police patrols are increased to keep the noise down [23].

The government also tries to raise awareness amoung youth concerning ear damage as a result of loud decibel levels[1].

1.1.3 Noise Pollution Maps

Noise pollution maps are geographical maps on which the sound level distribution existing in a certain region is represented with colours. Green represents low decibel levels, while red expresses high decibel levels. Noise maps give more insight into which regions are more susceptible to noise nuisance. This can be important information for people who are looking for a place to live and study, or to look for some quietness.

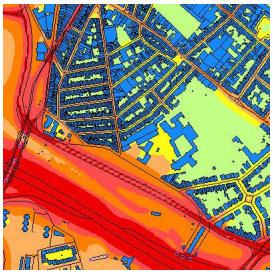


Figure 1.1: A part of the noise map of Antwerp, BE[2].

In Antwerp, Bruges and Ghent such noise pollution maps already exist [12]. An example of a noise map of Antwerp is shown in figure 1.1. It has been stated in the VLAREM[3] [18] that sound level maps should be used in order to identify noise nuisance sources.

1.1.4 Demand for Noise Pollution Maps

In Leuven, such official noise pollution maps are still absent. Jong CD&V Leuven[4] went out into the streets with a decibel meter application installed on their smart phone, to get a first glimpse of the noise nuisance in and round Leuven [31]. If they

[1] A campaign called 'help ze niet naar de tuut'; free translation: 'Do not beep them up'. (http://www.helpzenietnaardetuut.be)

[2] http://www.lne.be/themas/hinder-en-risicos/geluidshinder/beleid/eu-richtlijn/goedgekeurde-geluidskaarten/goedgekeurde-geluidskaarten

[3] Vlaams reglement betreffende de milieuvergunning

[4] Jong CD&V Leuven is a political party for young christen democrats of Leuven

have to go through whole Leuven, at different moments on different days, this job becomes very labour-intensive.

1.2 Goal

The main goal of the thesis is to use gamification techniques (see chapter 2) to facilitate the generation of noise level maps of Leuven. It has been chosen to develop a smart phone application to measure sound levels, which can be shown on a map. This is to be done in a fun way, so people using the application are engaged to measure sound levels throughout Leuven themselves. The application is called NoiseMApp.

The advantage of doing this is threefold. First, a noise map of Leuven can be created at a low cost. The purchase of standardized decibel meters and hiring people to go around Leuven at different times is rather expensive and labour-intensive. Therefore, an application can be created that makes it possible to record decibel levels, which can be done by many people at different times without cost.
Second, the noise level map will have recent data if the application is used frequently. This way, a more accurate and up to date view of the noise pollution and nuisance in Leuven can be obtained. If plenty of data becomes available, more specific conclusions can be drawn about noise pollution, allowing to focus more on environmental decisions and solutions.
Third, while having fun using the application, users can become aware of noise pollution and nuisance in Leuven.

Three main research questions form the basis of this thesis.

1. How can gamification be applied to the problem of generating noise maps?
2. Which game elements and techniques are most contributing to influence the activity of users?
3. How does the presented solution compare to a non-gamified application with the same goal regarding user activity and perception?

1.3 Content

In chapter 2 a background of the term gamification and an explanation of the use of this concept are given. Chapter 3 discusses how gamification is applied to the problem statement in section 1.1. Then, in chapter 4, the development and subsequent evaluation of a paper prototype of the application are explicated. The prototype is then translated into two digital versions, which are explained and evaluated in chapter 5. Considering the results of the prototypes, in chapter 6 the final application is fully presented and the implementation is discussed. The evaluation details of the application can be found in chapter 7. Finally, conclusions of this thesis can be read in chapter 8.

Chapter 2

Literature Study

Gamification found its roots in the beginning of the twentieth century. Yet, there still is few research available on the impact of the use of game elements to boost engagement. Also, there is almost no information available on how to use those game elements to maximise the gain of using gamification. In this chapter, a background will be given about the term gamification and what it exactly entails.

2.1 Gamification

2.1.1 History

Games were already played at 3000 before Christ [1]. Dice games, board games, gladiator fights, etc. are examples of known games in the ages of the Romans. Gaming therefore is certainly not a new concept. It has just changed shape and evolved because of the technological changes and commercialisation since the 18th century [9].

The first simple form of gamification to engage consumers can be traced back to 1912, where a cereal company hid toys into the cereal boxes so people could collect them. Ever since, a lot of companies have used this tool to stimulate people in buying their products.

This day, gamification is more than just putting a toy in a box. It includes systematically thinking about how to make things more game-like and fun. Keeping this in mind, the first mention of the term gamification goes back to 1980. Richard Bartle, an Essex student at the time, co-created the first text-only multi-user game on the internet. The term gamification was used for the work of Bartle, because he made a collaboration platform more game-like. He defined gamification as "turning something not a game into a game" [38]. From then on, studies were done about the use of games to let people study better [14]. At 2002, the Serious Games Initiative[1] was founded by Ben Sawyer and Dave Rejeski. It is a movement that combines academic, private businesses and military work that are using games for training and

[1] http://www.seriousgames.org

simulation. Again, these games are full-blown, which is not what gamification means nowadays.

The first example of the term gamification as it is used now is 'Conundra?'[2]. This was a consultancy organisation to make hardware products more game-like. Although the firm did not last long, it was the first clear example about what gamification is about. In 2005, Bunchball[3] came along to gamify enterprises [38]. Since then, gamification quietly exploded and can be found in a lot of applications, with Foursquare[4] [20] and FarmVille[5] [6] as the main examples.

2.1.2 A definition

The common definition of gamification is:

> the use of game mechanics and game design techniques in non-game contexts [14].

This definition is threefold. First of all, gamification is the use of game mechanics. These game mechanics are the familiar elements that can be seen in almost any social game, like FarmVille. It entails points, badges, leaderboards, (or PBLs), avatars, virtual goods, rewards, progression, quests, etc. [14] to keep the player playing [7]. PBLs can be found in almost any game. Points are used to obtain a measure with which players can compare themselves. To show off some acquired goals (which might be unknown to the player), badges can be earned, too. In the leaderboard, the players themselves can compete to be the first in the ranking. Avatars are used to let the player feel part of the game. Virtual goods are items that can be collected during a game. Points, badges and virtual goods are examples of rewards. Rewards can also come in other shapes, e.g. players can increase their level during game-play. Increasing levels often are unified with progression of the player. By showing this progression, players feel they are trying to reach a goal. Quests are often story told missions that players have to complete, which gives the game more purpose.
Secondly, to gamify an application, it is important to think like a game designer. This means making a game that people want to play. It is not enough to only use game elements in random places. That is why the use of game design techniques is also part of the definition.
Third, the gamified application must reside in a non-game context. A non-game context can be a task in the real world like running, being at a specific location, complimenting a colleague, learning a course, etc. Such a task in the real world has to be the starting point in order to develop an appliciation that gamifies this task. The objective of playing a gamified version of such a task must be an objective related to that task. For example, if the non-context task is running, then playing a gamified version of running should make one want to win by running harder. The

[2] http://www.nanodome.com/conundra.co.uk
[3] http://www.bunchball.com
[4] http://www.foursquare.com
[5] http://www.farmville.com

most vivid example is Nike+[6], which tries to make running more encouraging [38]. Generally, the goal of gamification is to engage people to change or maintain some kind of behaviour [30]: a business that sells better school results, running faster, increasing work environment, etc.

Although the definition given above tries to capture the whole meaning of the word gamification, it should be handled with care. Gamification is certainly not a strictly framed concept and it leaves a lot of room for interpretation. How gamification can be applied is still an active field of research, which has not gained a lot of results yet.

2.1.3 Games vs. gamification

As presented in the paper of Sebastian Deterding [13], gamification demarcates a distinction between game and play. Play or playfulness has to be thought of as a larger, looser category that contains, but is different from, games. This distinction is the result of Callois' concept of *paidia* and *ludus*[4]. *Paidia* is a more free, improvisational recombination of behaviors and meanings, whereas *ludus* is the activity of playing, bound by rules and a competitive strife towards goals. To give an example, *paidia* can be seen as the possibility to wander around in a game, just discovering the game world and *ludus* as the need to accomplish missions in games.

Gamification definitely relates to the latter. As said by Jane McGonigal [28], gamification leads to the suggested term *gamefulness* (or *ludus*) which is the opposite of the term *playfulness* (or *paidia*).

Gamification can be compared to games and serious games, toys and playful design by two dimensions [13]. The first dimension is the playing/gaming dimension, which has been explained in the previous subsection. The second dimension is the whole/part dimension, which can be derived from the definition of gamification, which states the use of game elements. When those two dimensions are drawn, one could situate gamification as shown in figure 2.1.

2.1.4 Gamification and fun

Gamification does not evolve in engagement directly. Gamifying a task by adding the ability to earn badges and points, might not make this task more fun. If it is not fun, people will not play it.

Thus, fun is also an important factor. People using a gamified application want to enjoy it. Only then the application can have a noticeable impact on the player. There are four keys to fun according to Nicole Lazzarro [22]:

- Simple fun is the fun people are having by just experiencing the game. It is the fun that people get from wandering about in a game and discovering everything.

[6] http://nikeplus.nike.com

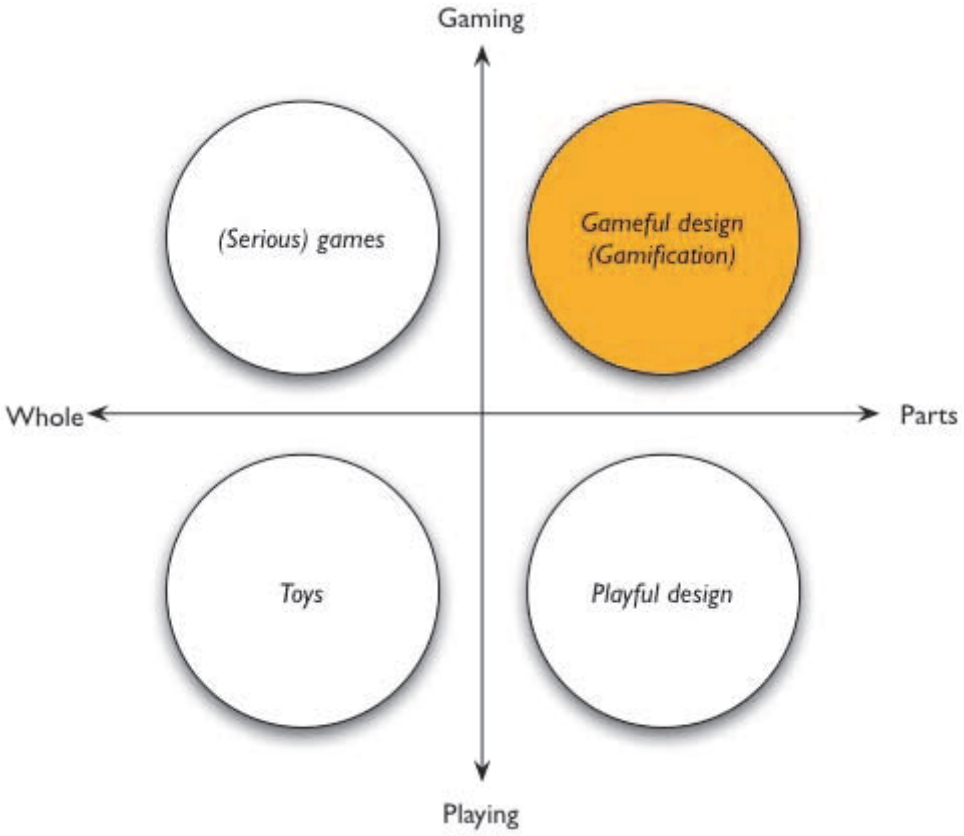


Figure 2.1: "Gamification" situated between game and play, whole and parts [14].

- Hard fun is the fun people have when they accomplish something. This can be a difficult level in a game or a new record.
- People fun is the fun experienced while playing with people: reasoning, collaborating, sharing.
- Serious fun is the fun people encounter when they do something for the better. An example of this is changing running behaviour.

Although this is a wide categorisation, it gives some insights in different sorts of fun that can be achieved by the player. It might help thinking about what gamification can be applied to.

In order to make a game fun, professor Kevin Werbach presents three techniques [2] that keep the player interested, and by extension engaged:

- Onboarding is about letting the player discover the options and let him play as quickly as possible. The learning curve should be balanced; not too steep, not too flat. Elements like hints, guidelines and sparkling options are manifestations of this.
- Scaffolding is the intention of keeping the user at play. E.g. it is not fun when

a game is too hard, because the player lost all his money. A donation of some money when the player passes 'Start' is a good example of scaffolding.
- Pathways to mastery must be created, so players feel that they make some progress. Player will not keep playing, if they think it is all for nothing. They need to have a (or multiple) goal(s).

2.2 Related Work

Applications already exist that try to generate noise maps. A brief summary of such applications is listed in section 2.2.1. All applications discussed are available for download at Google Play[7].
Research has also been done in trying to gamify this sort of applications, which is discussed in section 2.2.2.

2.2.1 Noise Mapping Applications

Eye on Earth is a Shared Environmental Information System (SEIS), which is an initiative of the European Environment Agency (EEA)[8]. They developed a mobile-based application, called NoiseWatch. With this application people can go out and measure noise levels wherever they want. These measures are recorded and stored on a server, which later can be viewed on a map on a website[9].

NoiseTube[10] is another example of a very similar project. It is a project coordinated by the VUB BrusSense Group[11]. They also created an application to map noise levels in a similar way to NoiseWatch. Additionally, they offer an API so other applications can send their measurements to the server of NoiseTube.

NoiseDroid[12] is yet another example of a noise mapping application. Besides the noise mapping functionality, a score is also given to the quality of the recording. The developers of this application have been contacted about the code, which they were happy to share. Unfortunately, the code was not maintained and deemed not usable.

2.2.2 Gamified Noise Mapping Applications

Some research has been done trying to gamify the process of generating noise level maps. Researchers at the Jaume I University have developed two prototypes [27]: Noise Battle and Noise Quest.
In Noise Battle [26], maps are divided into squares. Players have to compete to conquer as many squares as they can. They are rewarded with things they find

[7] Google Play is the online app platform for Android (http://play.google.com)
[8] The EEA is an agency of the European Union to provide people with environmental information.
[9] http://www.eyeonearth.org
[10] http://www.noisetube.net
[11] http://www.brussense.be
[12] http://play.google.com/store/apps/details?id=de.noisedroid&hl=nl

whenever they record a square. Squares can be overtaken by enemies that make recordings with better quality, or more recent. The main goal of Noise Battle is to encourage players to generate accurate and up to date data.
Noise Quest is another application prototype, in which players have to follow a story line. In this story they are asked to record at different points of interests (POI) through quests, missions and challenges they have to complete. The general idea of this prototype is to have a uniformly distributed noise level map around the city, throughout time.
Still, both prototypes have not yet been evaluated fully.

Researchers of the Telecooperation Lab at Technical University of Darmstadt[13] developed a real noise mapping application, called NoiseMap[14] [33]. It was also published on Google Play. The data of the measurements is also stored on a server and displayed on a website[15]. They introduce points that can be earned when users map noise levels. Also, a leaderboard with the points earned can be consulted. Because of this, this application is the closest example of a working application that is gamified to map noise levels at the time.

Conclusion

The thesis presented here, will use the definition of gamification as presented in section 2.1.2. However, gamification remains a rather unfamiliar concept and still a lot of research has to be done about how it can be applied. In this thesis, a way is presented to gamify the recording of noise levels in order to create a noise level map of Leuven. The focus mainly lies on the gamification of the process rather than on the generation of the noise level maps.

[13] http://www.tk.informatik.tu-darmstadt.de/de/telecooperation-group
[14] https://play.google.com/store/apps/details?id=de.tudarmstadt.tk.noisemap
[15] http://www.da-sense.de

Chapter 3

User Scenarios

3.1 Goals

The main goal of the NoiseMApp application is to get as many noise level recordings as possible, by making the application fun, using concepts of gamification.

It is easy to see that when the application is being used massively, more data will be acquired. Even when the application is used by only one user who is extremely motivated, the data obtained might be useful enough to create a noise level map. Of course, it is always better to have more people taking recordings of the surroundings. Firstly, the reliability of the sound recordings increase, because data is then being generated by different sources, which will result in mitigating outliers. Secondly, friends of users might notice the activity and join the sound recording community, which in turn will deliver more data. The more fun people have with the application, the more accurate and complete the data captured will be.

Engagement is the key to getting a lot of people to record the noise level at several locations. To achieve this engagement, concepts of gamification, discussed in chapter 2 are introduced. This way, the users of the application experience the recording of the sound level as a fun thing to do, while still providing useful information. All the scenarios discussed in this chapter are implemented in the prototypes discussed in following chapters.

3.2 Key Concepts

Gamifying an application requires, but is not restricted to, adding game elements to the application. These elements have already been discussed in chapter 2. The application of these game elements are being discussed in this section.

3.2.1 Points

Users can earn points by measuring the noise level. These points do not have any material value, but are introduced to compare the productivity of different users. This productivity can be measured by how much noise recordings they take, how fast they do it and how accurate these noise recordings are. The points are merely introduced to motivate the users to be as productive as possible.
For example, when a player has recorded the noise, he could receive 5 points: 3 points for making the noise recording and 2 points because the quality of the recording seemed good.
Also bonus points can be earned:

- when a badge is earned, which is discussed in subsection 3.2.2;
- when a player becomes the mayor of a particular area or street;
- when a player records for the first time in that area;
- when a player is the first of his friends to record in that area.

These points are totalled, and added to the users total score.

3.2.2 Achievements

Points can be earned, but players might loose interest if that is the only thing that can be gained in the application. To answer this problem, achievements are used to support the point system. Players can accomplish two types of achievements: temporary and permanent achievements.

When players record the most in a particular area for a specific time interval, they can become the mayor of that area. Mayors will gather extra points when they record in the area of which they are the mayor. Such mayorships are temporary: at any time, someone else might take the mayorship from them, by doing more recordings in that area at that time.

Players can collect badges as well, but these are held permanently. The same badge can only be acquired once, since it is linked to a specific target.

Targets include the following:

- first, fifth, fiftieth, hundredth, ... time doing a noise recording;
- first, fifth, fiftieth, hundredth, ... time winning a Sound Battle (see section 3.3.2);
- winning five, ten, twenty, thirty, fifty, ... Sound Battles in a row;
- every time a Noise Hunt (see section 3.3.3) is done;
- checking in for the first, fifth, fiftieth, hundredth, ... time;
- becoming the mayor of five, ten, twenty, fifty, ... areas.

Every badge also comes with additional points, but only at that time when it is received. For example, when players do a recording for the first time, they receive the Newbie Badge. Together with the badge, they acquire bonus points for that

recording. Next time they do a recording, the might not get a badge, because no specific target has been reached then.
The use of badges is three-fold:

- it will add an element of surprise to the game since the targets are unknown to the players;
- the players will feel rewarded for their work;
- players can easily compare themselves to each other.

3.2.3 Access

The concept of access can be seen in almost any game. A typical manifestation of this is the use of levels where players have to first complete level 1 before they can play level 2.

In NoiseMApp, such restrictions have also been implemented in the Noise Hunt, discussed in section 3.3.3. In the Noise Hunt scenario, players can only play a Noise Hunt once all previous hunts have been successfully executed and enough points are gathered.

3.2.4 User Profiles

Users playing the game have a personal profile. It gives a clear overview of the status of the player. Information is shown about acquired points, badges and possible mayorships. This way, users can track their progress.

Also, a leaderboard can be accessed through the profile to engage competitiveness. Friends can then take a look at each others profile.

3.3 Scenarios

In order to establish a real game experience with game elements and strategies discussed above, different scenarios have been defined. These scenarios include four game scenarios, one scenario to view the user profile and one scenario to view the map with all recorded noise. In chapter 4 the scenarios are being made concrete.

3.3.1 Random Record

The Random Record scenario is one of the four game-like parts that can be played in the application. It is the most simple one: players only have to go outside and record sound wherever and whenever they want.

For each recording, players can acquire points, badges and mayorships as discussed in section 3.2. This scenario makes it possible for players to increase their score easily. It represents the easy fun one could have with an application, discussed in section 2.1.4.

User Story Rachel is walking towards her work in Leuven. Every morning, it is the same dull route. She would like to make this walk more interesting. Then she remembers to have downloaded NoiseMApp. She chooses the Random Record, because she cannot loose too much time. On the way to work, she makes some recordings. Now her score has been improved.

Goal This scenario makes it possible to acquire a lot of noise level data everywhere in Leuven. Because players can easily earn points with this scenario by going out recording in Leuven, a lot of recording data will be generated through this scenario.

3.3.2 Sound Battle

The Sound Battle scenario tries to bring more fun into the application. With this scenario a real game experience should be perceived by the players. Two players have to compete against each other to make the best, most accurate noise recordings as fast as possible. The locations where to record are set, in contrast to the Random Record scenario.

User Story Finn has some time on his hands and wants to go outside, because the weather is good. He likes competing against other people, so he chooses to do the Sound Battle. Because his Friend Sam also has the NoiseMApp installed, Finn chooses to battle against Sam. Now, three Sound Battle locations are shown on the map. Finn tries to cover these locations as fast as possible to win the battle.
Meanwhile, Sam has received a notification that Finn has challenged him in a Sound Battle. He now tries to go to these locations as fast as possible to go recording. Once both have finished the game, they can consult their score. It seems Finn has won and he earned his First Sound Battle Won badge.

Goal Every time a Sound Battle is played, three locations are being recorded twice; one time by each player. Having more recordings at the same location increases the accuracy of the noise levels measured at that location. This is the main idea behind the Sound Battle scenario. By letting players interact with each other, recording noise levels might be more fun and engaging.

3.3.3 Noise Hunt

The Noise Hunt scenario can be played alone. It creates the possibility for the players to compete to themselves. The Noise Hunt challenges players to go *hunting* for specific locations.

User Story Quinn loves the NoiseMApp, but she thinks she can do better. She does not want to play against someone else, but on her own. She checks out the Noise Hunt. It seems she did not do any challenge, yet. She can choose the first one: 'Walk in the Park'. For this challenge, she has to go recording at three locations in the city park. The locations have to be separated by 50 metres. After recording at

these locations, she receives points for recording. Additionally, she also gets a badge for completing the 'Walk in the Park' Noise Hunt. Now, she can try the next Noise Hunt: 'Blitzkrieg'.

Goal Challenges will move a lot of players to the same location at different times. One could see the similarity with the goal of the Sound Battle. Creating such challenges yield the benefit that players can be manipulated to go recording near specific noise sources. This way, more control about which locations have to be recorded is possible.

3.3.4 Sound Checkin

The Sound Checkin scenario resembles to Foursquare [10]. Players can check in to places. By doing so, friends of players can see where that player is. In addition to Foursquare, a sound checkin is a checkin accompanied by recording the noise level at that place. Again, points, badges and mayorships can be earned by doing this.

User Story Kurt is at a discotheque and wants to let his friends know that he is already there. Brittany still has to leave home, but sees the checkin of Kurt. Because Kurt had to record the noise level, Brittany already sees that the noise level exceeds the 90 dB level. She can now decide to bring ear plugs, because she does not like dancing when the sound is that loud. Kurt was very informative by checking in, so of course he gains some points.

Goal Except only letting your friends know where you are, it is also interesting to give them sound level information of that location, too. The main goal of this scenario is to make people aware of noise levels. It is certainly not the goal to check upon pub owners who are not respecting noise levels. Friends that receive this information, can decide whether they should bring along ear plugs if they feel the sound level is too high.

3.3.5 View Profile

Earning points, badges and mayorships is fun, but there should be some room to let players brag about it. A profile is a typical way to do this. The profile exists of personal information with the acquired points, badges and mayorships and a leaderboard in which the players are situated between their friends.

User story Tina already played the NoiseMApp many times. Because she lost track of how many points she already earned, she checks her profile. It looks like she has held down two mayorships! Now, she takes a look at the leaderboard and sees that her friend Santana has more points and resides on first place! Because she wants to be number one, she decides to play more.

Goal By showing players how good they are doing, they might get triggered for more engagement. The leaderboard is a powerful technique to let players compete with each other.

3.3.6 View Map

This scenario shows the players what they are actually doing. On a map, all data is shown so players can have an idea of the noise level in Leuven.

User story Mike is a student in Leuven and lives at his dorm in the first part of the Tiensestraat. He did not know when he came to Leuven that it could be this loud over there with all the parties at the *fakbars*. Thanks to NoiseMApp, he now has information about the noise levels in Leuven. He sees that it is much quieter in the second part of the Tiensestraat.

Goal The most important part of the application is to gather noise level data. This data exists of sound levels at a specific location. Players cannot loose the idea behind NoiseMApp. They have to know that recording noise levels might help them in finding quiet spots to live. This way, players can be made aware about the noise level in Leuven.

Conclusion

A way is proposed to gamify the generation of noise level maps. Key concepts of games are analysed and are used in order to gamify the application. Points, badges, user profiles and the use of access restrictions will be incorporated in the application. Also 6 different scenarios are presented, including 4 game-like scenarios (Random Record, Sound Battle, Noise Hunt and Sound Checkin) and 2 others (View Profile and View Map). These scenarios represent the game techniques that will contribute to the gamification of the noise mapping application.

Chapter 4

Paper Prototype

This chapter describes the implementation of a prototype on paper of the scenarios discussed in chapter 3. The paper prototype is evaluated by real users, of which the results can be found in section 4.3.

4.1 Goals

When beginning the design phase of an application, it is good practice to start with a paper prototype. Paper prototyping has the advantage [34] to quickly sketch the look of the application, while eliminating technology variables. Buttons, icons and text of the application can be drawn, but also the flow of the different screens have to be defined. Without writing any line of code, an application can be simulated as if it were a real working one.

The evaluation of a paper prototype by real test users allows an early detection of possible issues concerning the usability of the application. Throughout several iterations, problems can be discovered and solved quickly with almost no effort.

4.2 Design

This section describes the design of the paper prototype for the NoiseMApp application. First, the creation of the prototype in general is depicted. Then, the several flows of the NoiseMApp application applied to the paper prototype are described. The figures are ordered in such a way that the scenario flows have to be read from left to right, top to bottom. The flow defined in this section is the normal flow of the application and how it should be executed. In subsection 4.3 the evaluation by test users of these flows is described.

4.2.1 Creation

The paper prototype made for this thesis is designed on a computer, since it is more clear than drawing. Although drawing a paper prototype is one of the big advantages, it is more time consuming. When buttons that appear on every screen are changed, all screens with these buttons have to be redrawn. Using a program like Paint.NET [15] makes this updating process more user-friendly.

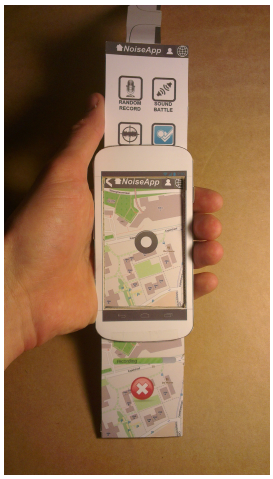


Figure 4.1: The paper prototype.

It feels important to make the prototype as true to a real application on a smart phone as possible, considering people using the application will have to run around with it. For this reason, a cardboard model of an Android phone has been made, of which the template is made available by the Android Developers website [1]. People evaluating the paper prototype, can grasp the model as it were a real smart phone. The cardboard model also serves as a casing for the paper screens of the application. The different screens of one flow are taped together at top and bottom, making it possible for the application flow to be slidden through the casing as shown in figure 4.1.

Almost the entire paper prototype is made in black and white. Colours might be too distracting in order for the goal to be achieved.
Generally, the design of the application is inspired by other applications, like Foursquare [10]. A minimalistic design has been chosen where screens are simple, not overcrowded with images and text.

4.2.2 Main screen

All scenarios have to be started from the main screen shown in figure 4.2, except the View Profile and View Map, which can be started from any screen. Four buttons

[1] http://developer.android.com/design/downloads/index.html

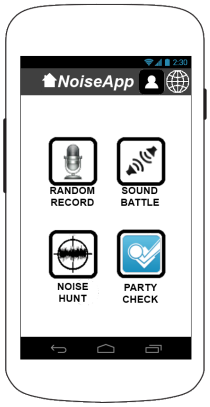


Figure 4.2: The main screen of NoiseMApp.

stand out on the screen. These buttons are used to start the four different game-like scenarios. The top-left button starts the Random Record scenario, the top-right begins the Sound Battle. The bottom-left and bottom-right represent respectively the Noise Hunt and the Sound Checkin. For clarity, the names of the scenarios are also shown underneath the buttons.

In the top bar, the name of the application is shown, which was NoiseApp at the time. Also two icons can be found in the top-right corner. The first one represents a user, and will show the profile of the player. The second icon is a globe, which will turn the player to a map with an overlay of all recorded noise.

It should be noticed that the top bar also acts as a home button. If a player pushes the name of the app, they return to the home screen.

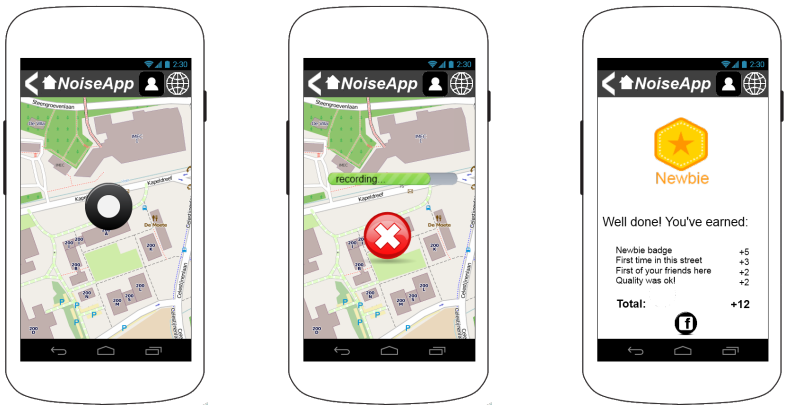
4.2.3 Random Record

Flow

1. Once the Random Record has been chosen in the main screen, a screen with a map and a record button appears (figure 4.3a). Players can now start recording whenever they want, by pushing the big button in the middle of the screen. The record button also functions as a location marker, i.e. it shows where the player is. This way, the record button will follow the player on the map.
2. While recording, the progress is shown. Also a cancel button can be used to cancel the recording, returning the player to the record screen (figure 4.3b).
3. If the recording is done, a points screen appears. This points screen holds information about acquired badges, the points gathered and possible mayorships earned (figure 4.3c).

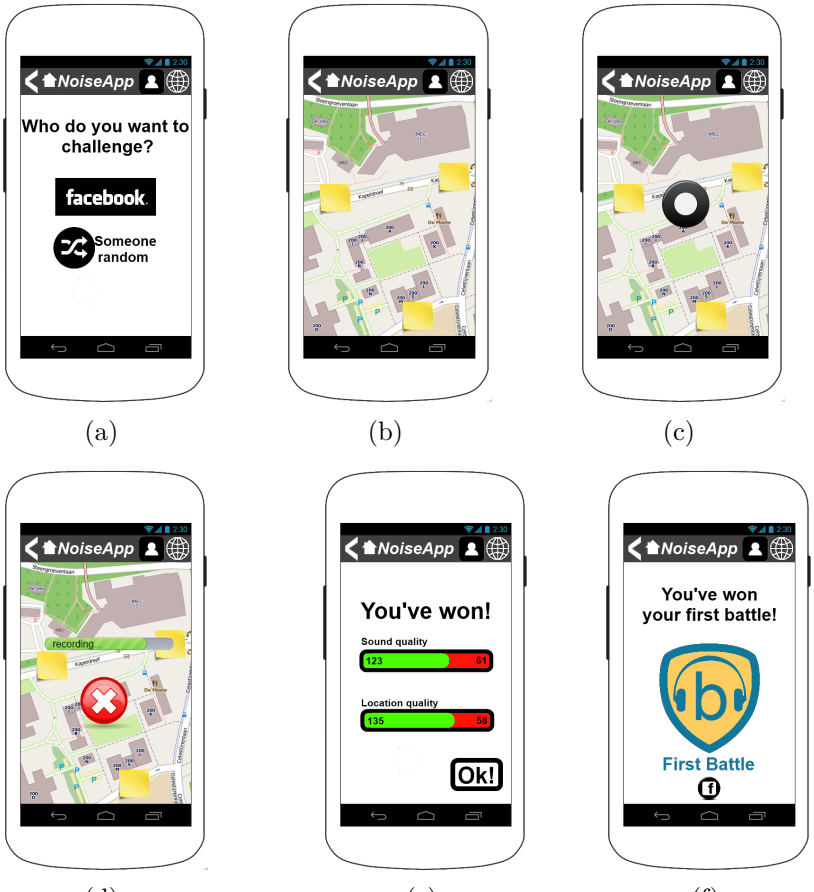
4.2.4 Sound Battle

Flow



(a) (b) (c)

Figure 4.3: The flow of the Random Record scenario.



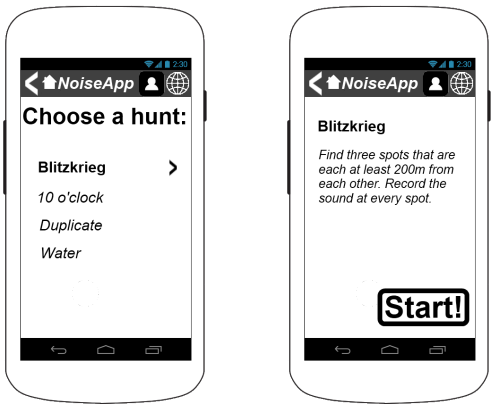
(a) (b) (c)

(d) (e) (f)

Figure 4.4: The flow of the Sound Battle scenario.

1. Once the Sound Battle has been chosen in the main screen, players must choose whether they want to battle a random player or a friend (figure 4.4a). The assumption has been made that the player chose the random option, which makes the flow easier.
2. A map appears like the one in the Random Record scenario appears, but no record button is shown. Also, three sound battle locations are represented on the map with Post-Its (figure 4.4b).
3. Players now have to move towards one of the unrecorded locations. If they are close enough to such a Sound Battle location, the record button appears (figure 4.4c). By pushing the button, the recording starts.
4. The screen with the progress and the cancel button is shown (figure 4.4d).
5. After the recording, the respective Sound Battle location is checked off and the players return to the recording screen. Steps 2 to 4 are repeated until all three Sound Battle locations have been recorded.
6. Afterwards, a summary of the points earned with this scenario is shown (figure 4.4e). This summary includes two parameters:
   - sound quality: this is the quality of the sound. As described in chapter 2, quality of sound can be measured. Movement of the phone and wind in the microphone are parameters that lessens the quality of the recorded sound.
   - location quality: this is the accuracy of how close the players get to the given Sound Battle location. The closer players get to the location, the more points they earn.

4.2.5 Noise Hunt



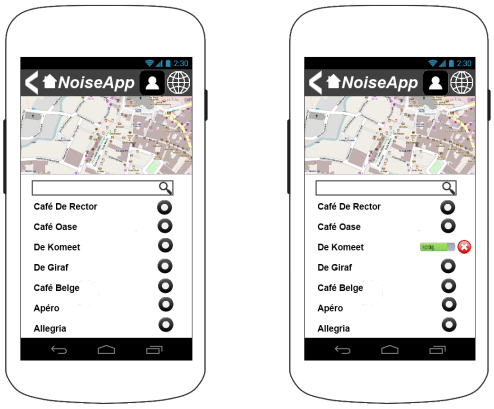
(a) (b)

Figure 4.5: The flow of the Noise Hunt scenario.

Flow

1. Once the Noise Hunt has been chosen in the main screen, players can see all the completed, available and unavailable Noise Hunts (figure 4.5a). Of course, one of the available Noise Hunts has to be chosen, which is 'Blitzkrieg' in case of this flow.
2. A screen appears with more information about the specific Noise Hunt (figure 4.5b). Players can start playing by pushing the 'Go' button.
3. A recording screen appears just like the one that appears in the Random Record scenario (figure 4.3a).
4. The player tries to finish the given noise hunt.
5. A point screen appears like the one in the Random Record scenario (figure 4.3c).

4.2.6 Sound Checkin



(a) (b)

Figure 4.6: The flow of the Sound Checkin scenario.

Flow

1. Once the Sound Checkin has been chosen (in the paper prototype it was called Party Check at that time), a screen appears with all bars, restaurants and discotheques close by (figure 4.6a). Players have to choose where they are and push the appropriate record button.
2. When recording, the progress bar appears with a cancel button (figure 4.6b).
3. Finally, when the recording is done, a point screen appears like the one in the Random Record Scenario (figure 4.3c).

4.2.7 View Profile

Flow

1. The profile can be checked by pushing the user shaped button in the top bar. This button can be accessed from any screen in the application. The profile

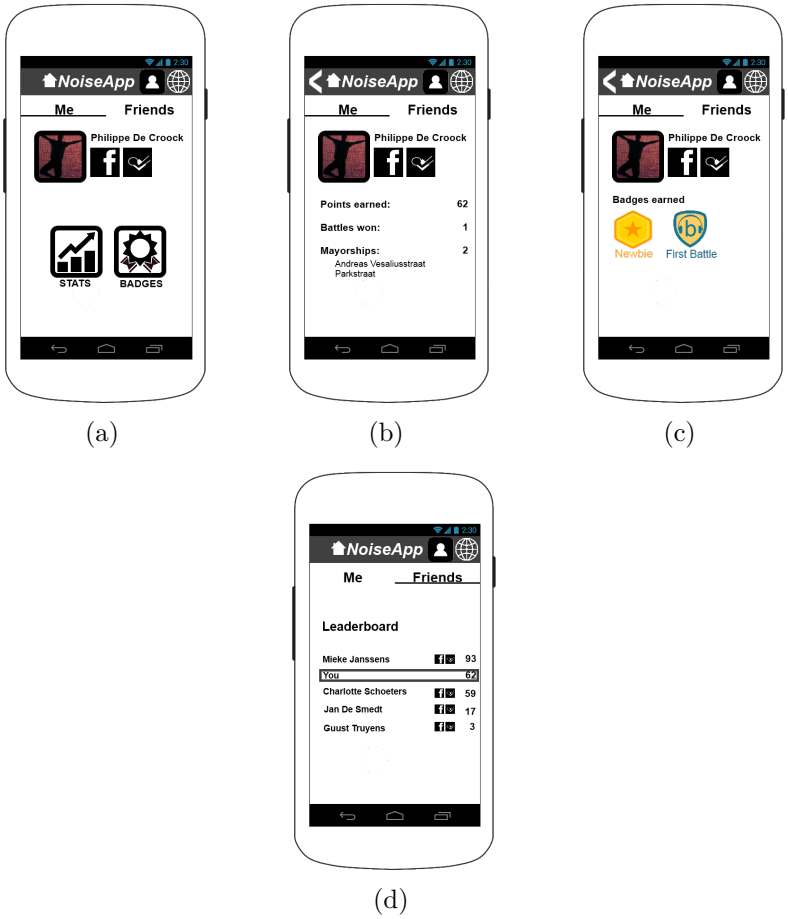


Figure 4.7: The flow of the View Profile scenario.

consists of two tabs: 'Me' and 'Friends'. On the 'Me' tab there are two buttons to choose from (figure 4.7a).

- When the players want to see the statistics, they have to push the statistics button, which will direct them to the statistics screen (figure 4.7b).
- When players want to take a look at the badges they collected, they should push the badges button (figure 4.7c).

2. Profiles of friends can be seen through the 'Friends' tab (figure 4.7d). A leaderboard appears in which friends are sorted by the points they have gathered.

4.2.8 View Map

A paper prototype version of the View Map scenario has not been implemented. Reason for this is the overlap with other scenarios that have a map, which makes it redundant. Still, the flow of the scenario is given below.

Flow

1. Once players push the globe icon in the top right corner of every screen it will turn them to the map screen. The map has an overlayed of colored dots, which represent the different sound levels of the recordings measured at that location.
2. When tapped on a dot, information appears about the exact sound level.

4.3 Evaluation

In this section, the evaluation of the paper prototype is explained. First, characteristics of the test population is shown. Then, the setup is described of the evaluation and the questionnaire. Finally, the results are depicted in section 4.3.2.

4.3.1 Setup

Demographics

Eight people have tested the paper prototype, averagely aged 22 years old. Four of them had an Android phone and one of them did not have a smart phone at all. Half of the people were tested at the Department of Computer Science in Heverlee. The other half was tested at the library of Psychogology. Both sessions took place at a table inside the building.

Scenarios

To evaluate the paper prototype, the test users were asked to perform three scenarios. Only three scenarios were chosen to keep the testers alert. They never had to spend more than 15 minutes on the evaluation.
Every scenario was introduced by telling them a little story:

- Random Record:

  > You are walking outside and you are bored. You have just installed your new NoiseApp application. Now you want to record sound, right where you are. How are you going to do this?

- Sound Battle:

  > So, that were easy points! Although you do not gain a lot of points with Random Record. You can earn more points with competing against someone else. I would like you to play a game against a random player.

- View Profile:

  > Now you want to check how many points you have earned. Can you try to find a leaderboard to situate yourself between your friends?

Two of the three scenarios involved recording sounds, which is of course the most important flow of the application. Users were asked to think out loud [37] during the execution of the flows. Consequently, a lot of useful information can be revealed about the look and feel of the application and the application flow.

System Usability Scale (SUS) questionnaire

Afterwards, the test users had to fill in a System Usability Scale (SUS) questionnaire [3]. This questionnaire consists of 10 statements that had to be answered using a 5-Likert scale. The test users have to answer with a grade from 1 (strongly disagree) to 5 (strongly agree). Thanks to this scale, a general score can be extracted from all the answers.

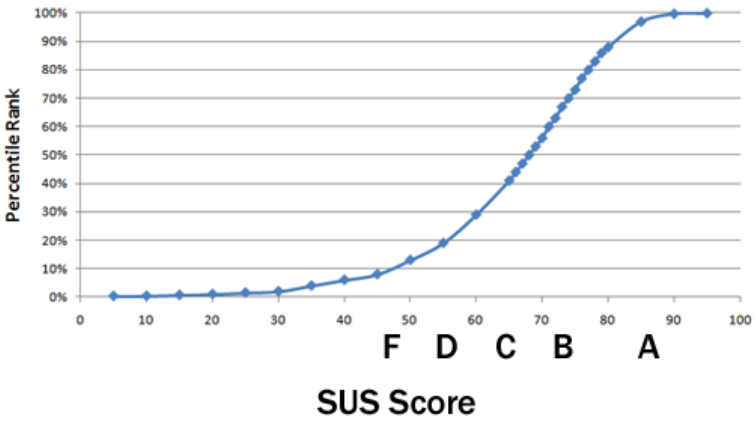


Figure 4.8: Percentile ranks associated with SUS scores and letter grades [32].

Some background should be given to interpret this score. A study [32] has been done in which more than 5000 people took the SUS questionnaire for 500 evaluations. The average SUS score of applications tested with this questionnaire yielded 68. Obviously, a score above 68 would be above average. These results are shown in a graph in figure 4.8. From the figure it is clear that a score of 80 can be perceived as a good SUS score, since it has only been reached by 10% of all 5000 evaluations.

The amount of people taking part in the SUS questionnaire, also plays a role in the conclusion that can be drawn of the results. Researchers at the Fidelity Center for Applied Technology have evaluated two websites with different sorts of questionnaires [36]. They did this with different group sizes of 6, 8, 10, 12 and 14 members. The members afterwards had to prefer a website. In figure 4.9 the percentage of the correct conclusions of different questionnaires are compared. A conclusion is considered correct if the outcome of the questionnaire indicates the preference the test user has. One may conclude that SUS is a relatively good and practical questionnaire that does not need many users to reach a correct conclusion.

4.3.2 Results

In this subsection, the results of the evaluation are being discussed. Generally, no big problems were discovered throughout the whole evaluation. Everyone seemed to enjoy themselves using the prototype and were interested in what the application

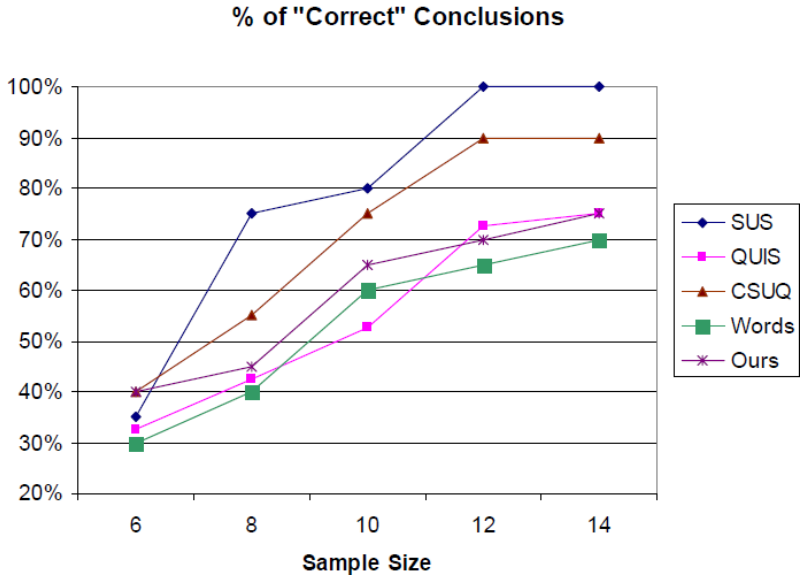


Figure 4.9: Percentage of correct conclusions according to group sizes [36].

does. Yet, some issues have been discovered by the evaluation of the paper prototype. First, problems that occurred during the execution of the scenarios are exposed. Second, the results of the SUS questionnaire are depicted.

Flow Problems

Record button The record button in the middle of the screen did not seem clear at all (figure 4.3a). Some users did not know what had to be done when they first saw the screen. Also, it was not clear that the record button also had the function of a location marker. Of course, since the application was still in a paper prototype phase, the location marker could not be made clearer. This would demand the use of dynamic images, which is impossible to do on paper.
Even more confusion was introduced when playing the Sound Battle, where there is no record button or location marker to begin with.

Progress screen Also the progress screen raised some issues (figure 4.3b). It seemed that the cancel button caused some confusion. One person even said that it felt like something went wrong, because of the big red cross.

Sound Battle locations Sound Battle locations were represented by post-its on the screen (figure 4.4c). Using post-its was simply not the best method to do this. Many people did not get that the post-its represented Sound Battle locations. Some even thought they were other players, which is of course not the case. The opponent is not represented in the game.

Location quality When a Sound Battle has been completed, the overview of the points earned shows up. Meanwhile they were asked if everyone was clear in the overview. At that time, most people did not know what location accuracy actually meant. It was definitely not the best term to use.

Friends When people were asked to go to the leaderboard, most of them did not get to the right screen immediately. For most people, it was not clear that the profile view existed of two tabs. This could be caused by the bad representation of the tabs on the paper prototype.
Also, one test user found it confusing to look for the leaderboard at the Friends tab.

Top bar The bar on top of the application was not that handy as first expected. Most of the test users always went back to the main screen to go to the profile, which is not the fastest way to do this. The people that did find the fastest way, knew this because they had previously erroneously pushed the button and had to be told that that was wrong in that case. For example when asking the test users to go to the leaderboard, they often tapped on the globe icon.
In addition, nobody found out that pushing the title would return them to the main screen.

SUS Questionnaire

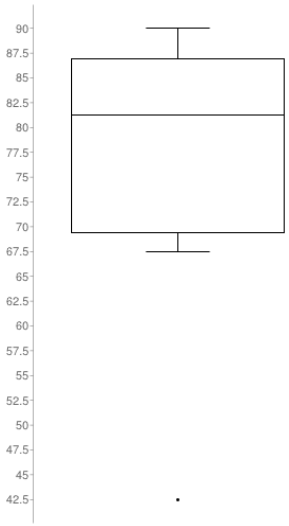


Figure 4.10: Boxplot of the SUS-scores.

Every test user had to fill in a SUS questionnaire. A boxplot of the score is given in figure 4.10.

As shown in the figure, the result of the test on the paper prototype of the NoiseMApp application averaged to 76.25. If the outlier at the bottom is neglected,

the average raises up to 81.07. Neglecting the outlier makes sense, since the test user responsible for that score did not know how to use a smart phone at all. After all, the application would reside on a smart phone and basic knowledge of using smart phones can be expected.
The median is 81.25.

One might say that evaluating a paper prototype through a SUS questionnaire is not that helpful. Of course, the usability of the application cannot be tested fully. Evaluating the paper prototype this way, though, already gives a hint of how the application is being perceived by the test users. Therefore, the obtained result might serve as a a lower bound.

Conclusion

In this chapter, the paper prototype has been explained and evaluated. The cardboard casing tried to give test user the feeling that they were holding a real device, which might give a more realistic view of the result obtained by the evaluation.
Overall, the paper prototype scored well. There were no critical flow issues, only the look of buttons and what they did were sometimes unclear and should be changed.

Because the paper prototype scored this well, the development of a first digital prototype has been initiated. The digital prototypes are discussed in chapter 5

Chapter 5

Digital Prototypes

This chapter will handle details concerning the digital prototypes. For the full details of the implementation, the reader is referred to chapter 6.

Two prototypes have been made and are explained in next sections. Both prototypes have incorporated changes that are elicited from evaluations of the paper prototype. They are also both evaluated to discover more issues regarding the usability and usefulness of the application. Some implementation details are discussed that are needed to understand how the prototype works.

5.1 First Digital Prototype

5.1.1 Goal

In the context of Rapid Prototyping, digital prototyping is a well-known part of the development process [24]. This has the advantage to easily present dynamic interfaces to test users. Alhtough paper prototyping an application helps to quickly create a user centred lay-out, the NoiseMApp application is highly dynamic and location-aware, which is hard to make clear to the user by only using paper prototypes. Also, evaluating a prototype of the application on a real hand-held device makes it more real to the test subjects to imagine a fully working application.

The first digital prototype is implemented with two of all six scenarios that are discussed in chapter 4: Random Record and Sound Battle. Reason for this is that these two scenarios are the ones most shaping the application. Consequently, it is possible to quickly evaluate the first prototype to see if the design, usability and flow of the prototype is going in the right direction. An important aspect of rapid prototyping is to let the test users think aloud [37]. This immediately brings problems to light, which could be forgotten if asked afterwards. Hence, no extra time is lost implementing all scenarios in future iterations.

It is important that the prototype looks as if it really works. They need to have the possibility to push buttons and interact with the prototype as if it was a

real smart phone application. However, no real noise recordings, point and badge calculations or Sound Battles are made, which makes it a prototype in any sense. Every point or badge the test user seems to receive is hard coded, which means that every time the scenario is executed, the same points and badges are earned. The Sound Battle is purely an illusion of playing a game against someone else.

On the main screen, all six features are shown, but only the two features stated above can actually be chosen (see figure 5.6).

5.1.2 Changes

Due to the problems discovered in the paper prototype, discussed in section 4.3.2, the prototype has been changed. The changes to the prototype can be read in following sections. For a full overview of the flow of the scenario, the interested reader is referred to appendix C.1.

Record Button

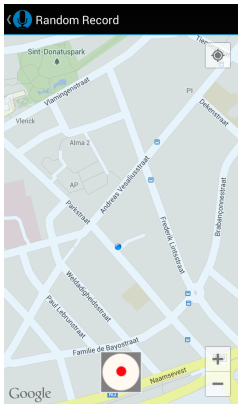


Figure 5.1: The record button has been changed in the first NoiseMApp prototype.

The record button has been moved to the bottom of the screen (see figure 5.1). This is done to solve the confusion regarding the record button in the paper prototype. Before, the record button had two functions: it was a location marker and a record button. Now, those two functions are separated. The location marker now is the default blue dot or triangle of Google Maps.
The button has been altered so it should be more clear that it is a record button. It will also remain on the screen the whole time, in contrast to the paper prototype.
Additionally, the recording only starts when a GPS fix is found. If the record button is pressed while the device is still searching for a GPS signal, a message appears that the user has to wait.

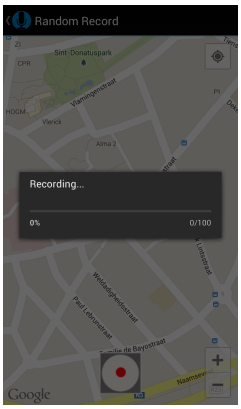


Figure 5.2: The cancel button has been changed in the first NoiseMApp prototype.

Cancel Button

The cancel button that appeared during a recording in the paper prototype, has been deleted (see figure 5.2). Some people thought that an error had occurred when they saw this cancel button in the paper prototype. To cancel a recording, it now suffices to press outside of the progress screen, or on the back button.

Sound Battle Locations

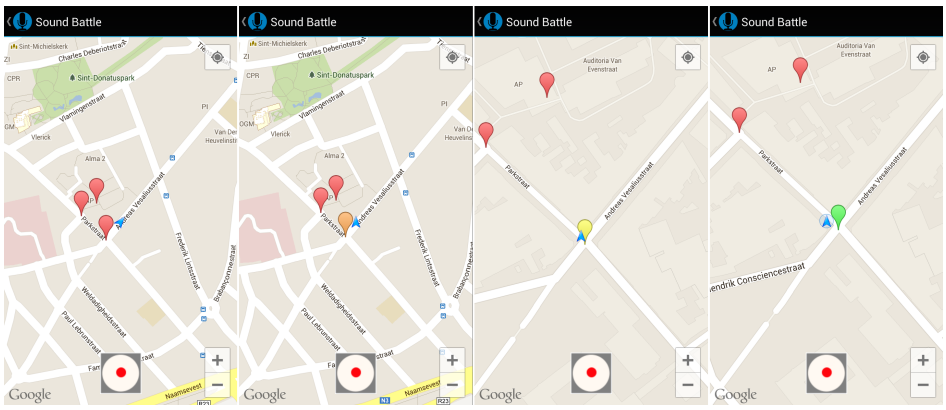


Figure 5.3: The Sound Battle locations have been made dynamic in the first NoiseMApp prototype.

A digital prototype makes it possible to have dynamic Sound Battle locations on the screen (see figure 5.3). In the paper prototype, Sound Battle locations were represented with post-its (see figure 4.4b). The dynamic Sound Battle locations have a colour code: red when the user is too far away from the location, orange when the user is closer and yellow when the user is close enough to the location to start the

recording for a Sound Battle. When a recording is done, the marker turns green to show that the sound at that location has been successfully recorded.

Location Quality

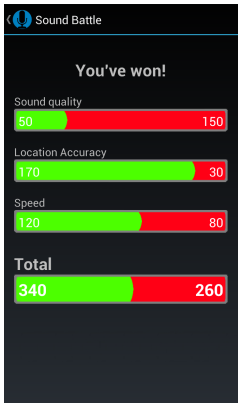


Figure 5.4: The term location quality has changed to location accuracy in the first NoiseMApp prototype.

The term 'location quality' has been changed to 'location accuracy' in the points screen of the Sound Battle (see figure 5.5). In the evaluation of the paper prototype, a lot of the test users did not really understand what location accuracy meant.

Speed

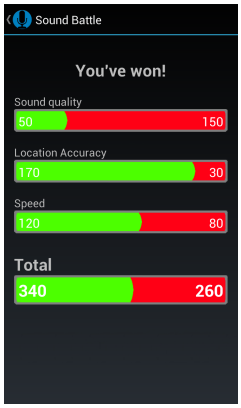


Figure 5.5: The term location quality has changed to location accuracy in the first NoiseMApp prototype.

Speed is added to the parameters to win (or lose) a Sound Battle (see figure 5.5). The time passing between the first and the last recording is measured. If this time interval is shorter, more points can be gained.

Top Bar Icons

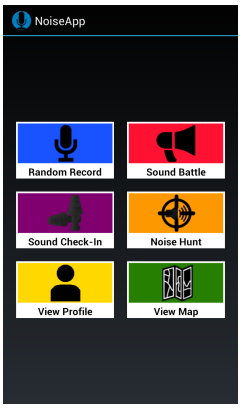


Figure 5.6: The View Profile and Show Map button have been moved in the first NoiseMApp prototype.

The button to view the profile and view the map with all noise recordings has been moved from the top bar to the main screen (see figure 5.6). Almost every test user returned to the main screen when testing the paper prototype to go to the profile, indicating the illogical position of these icons. It also did not seem to make sense to check the profile or the noise map during game play. The alert reader might have noticed that the name of the View Map scenario has been changed to Show Map.
Also, it seemed impossible to implement the launcher icon as a home button in the digital prototype, although this was suggested in the paper prototype.

5.1.3 Implementation Details

In order to understand better the descisions made, some necessary implementation details are explained. A summary of all the implementation details can be read in chapter 6.

Google Maps API Although the application still is quite static (no dynamic content is created), the map on which the location of the test user is presented, is necessary for the test users to perceive the prototype as a real application. The Google Maps API has been used to support this feature. The API makes it easy to implement a map screen into the application, which looks familiar to a lot of Android users using Google Maps.

Sound Battle Locations In Sound Battle, random Sound Battle locations in Leuven are generated. These locations have to be on streets that are accessible to the public, so both players can reach them. Google Maps API[1] seems the obvious

[1] https://developers.google.com/maps

choice to do this, but this was not as easy as expected. One can request Google to return coordinates of places, like restaurants, but this is not sufficient. Coordinates of a whole street are needed, not only of parts of streets. For this reason, an OpenStreetMap[2] has been downloaded of Leuven. All possible roads, streets and little pathways are on these maps. Using Quantum GIS[3], all coordinates of these streets can be extracted. The NoiseMApp prototype picks three of these locations, within a range of 200 meters. These values haven been chosen arbitrarily. In Leuven there are 9998 coordinates, which results in a large file to load. It is also more labour intensive to generate random locations in such a big file. To overcome this problem, Leuven is divided into 35 quadrants, of which the coordinates are stored in separate text files. This effects the loading time drastically.
The disadvantage of this method is that the Sound Battle cannot be played outside of Leuven. Thus, the application is not very scalable in the present implementation.

5.1.4 Evaluation

The evaluation of the digital prototype is similar to the evaluation of the paper prototype. Now, the testing is done with one HTC One X[4] device, on which Android 4.2[5] (Jelly Bean) is installed.
The two implemented scenarios are tested: the Random Record scenario and the Sound Battle scenario. Again, the users were asked to perform these scenarios and to think out loud during the test. During the test, they were asked questions about the look and feel and about things that might be unclear to them. They were also asked to fill in an elaborate questionnaire on Google Drive of which the full summary of the results can be found in appendix **??**. This questionnaire was split up in six parts:

1. personal information;
2. smart phone information;
3. game experience;
4. fun experience;
5. SUS questionnaire;
6. look and feel.

Population

Five people in total have tested the prototype. Two of them are Masters in Computer Science, one is majoring in Geography, one is studying Informatics and one, who has a background of Chemical Engineering, is already working for four years. The ages reach between 18 and 27. Four of them own a smart phone, of which two have Android installed on their device and the other two have an iPhone with iOS. These four people also play games on the smart phone at least sometimes, which meant once a week or more. All test users are new: they have not tested the paper prototype.

[2] http://www.openstreetmap.org
[3] http://www.qgis.org
[4] http://www.htc.com/uk/smartphones/htc-one-x
[5] http://www.android.com/about/jelly-bean

Results

Here, the observations and results of the user test and questionnaire are presented.

User Test

What When test users were starting the random record scenario, they did not really know what to do. Some of them asked: "Ok, what do I do now?". One person thought that the recording started as soon as he entered the screen with the Google Maps.

When As mentioned earlier, the recording can only start when the GPS has a fix. Otherwise, a message appears remembering the test user has to wait until there is a GPS fix. Android OS shows a blinking GPS icon in the notification bar when searching for GPS and it stops blinking when such a fix is found. Though, this did not seem clear enough to most users, even to one of the Android users.

Sound Battle Locations The colouring of the markers that show how far the user is removed from the location, caused most users to keep on searching to get closer to the given location when the marker was already yellow. It seemed the colour yellow was not clear enough to represent the possibility to start the recording. Four out of five users said green would be a better colour. Yet, the marker changes colour to green when the test user recorded a sound successfully.
Also, only one test user pressed the location marker, which shows the longitude and latitude and a message stating that the user has to come closer to the location or is ready to record.

Recording Done When a recording has been executed successfully, the Sound Battle location dot turned green, but only after the test user had moved location. This caused three users to keep on waiting and the other two to ask what had to be done afterwards.

Sound Quality Although location accuracy on the points screen seemed to be understood better, more questions were asked about how sound quality was measured.

Questionnaire All results that seem of interest are analysed here.

On a scale from 1 (strongly disagree) to 5 (strongly agree), users had to give a score to following statements: "I often have problems concerning noise pollution and noise nuisance" and "I think it makes sense to generate noise pollution maps". Although only one test user seems to deal with noise pollution regularly, all the test users think the application is useful. This seems to be an indication that the

application can fulfil the goal of making a whole noise map of Leuven.

Also, users had to give a score on a scale from 1 (strongly disagree) to 5 (strongly agree) whether they thought the application would be fun, once it was finished. Four out of five think it would be fun indeed, but an equal amount of test users think it is not very challenging. Referring to the types of different fun in section 2.1.4, it seems that the application rather fits the category of easy fun.

There were no objections concerning the separation of 200 metres between Sound Battle locations. Most of of them found it important that the Sound Battle locations are the same for them as for the opponents. The amount of locations, which is three, also did not raise any problems. The recording of the sound, which takes 10 seconds, did not seem to last too long neither.

SUS Questionnaire The SUS questionnaire and its meaning has already been explained in section 4.3.1. Since only five people have tested the application, the SUS questionnaire is not very reliable. As shown in [36], out of only less than 35% of the results, the correct conclusions can be drawn. Still, the SUS score gives an indication of the usability of the application and might shine some light on problems encountered by the participants.

In general, the mean score given to the application is 81 and the median is 80. This is the same score as the paper prototype, which might seem rather disappointing. The background of the population might be an explanation for this. Still, 81 is a very good score and indicates that the design of the application is still moving into the right direction.

5.2 Final Digital Prototype

In this section, the full digital prototype is described. This prototype has undergone some changes, due to the evaluation of the first digital prototype.

5.2.1 Goal

In the final digital prototype, all scenarios are implemented, including Noise Hunt, Sound Checkin, View Profile and Show Map. Still, it remains a prototype, which means no real recording functionality is implemented, yet.

5.2.2 Changes

Considering the problems in section 5.1.4, some changes have been made to the lay-out of the digital prototype. These changes are explained here.

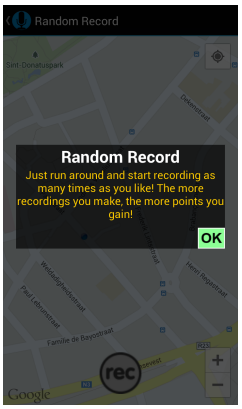


Figure 5.7: An example of an explanation popup in the full NoiseMApp prototype.

Explanation Pop-ups

As mentioned earlier, when performing the Random Record scenario, some of the test users did not really know what to do. Now, whenever any recording scenario is started, an explanation of that specific scenario is given (see figure 5.7). Such an explanation popup was already existing in the Sound Battle scenario, for which most users did not need any help to understand it. The users have the choice to never let the popup show again.

Record Button

Figure 5.8: The changed record button in the full NoiseMApp prototype.

The record button has been changed to an icon with "rec" on it. When pressing the button, it turns red. This should increase the clarity of the button and its function.
Since recording can only begin once a sufficient GPS signal is found, a message appears to make this clear to the user.

Sound Battle Location Markers

In the full digital prototype, the icons of the Sound Battle locations are changed to filled coloured circles with crossing swords in it (see figure 5.9). The markers change colour, considering the distance of the user to the Sound Battle location. The colour yellow to indicate that the user is close enough to start recording has been changed to green. The green coloured marker is changed to a green check mark.

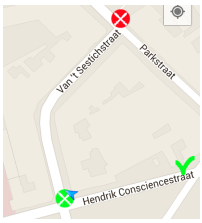


Figure 5.9: The changed Sound Battle location markers in the full NoiseMApp prototype.

Sound Checkin

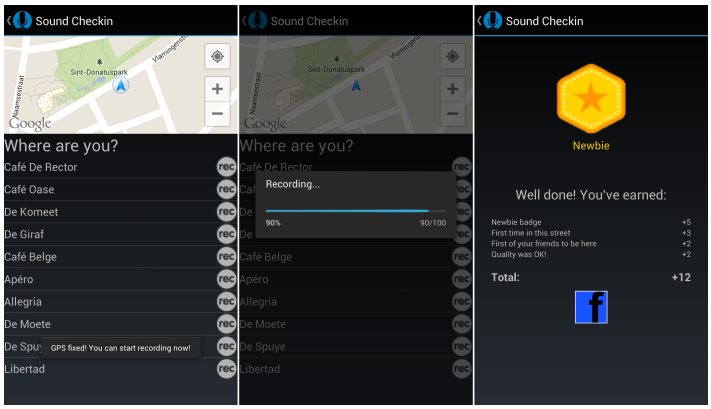


Figure 5.10: The Sound Checkin scenario of the full digital NoiseMApp prototype.

Now, the Sound Checkin has also been implemented in the prototype. With Sound Checkin, one can check in to a location just like they would on Foursquare [10]. When clicking the Sound Checkin button, a screen appears where places of the Google Places database in the neighbourhood are listed (see screen 1 in figure 5.10). Places include restaurants, bars, discotheques, but also parks and cultural buildings. When tapping such a place, the recording starts (see screen 2 in figure 5.10). After the recording, a similar points screen to the one of the Random Record scenario appears (see screen 3 in figure 5.10).

Noise Hunt

It is also possible for the test user to check out the Noise Hunt feature of the prototype. When tapping the Noise Hunt button in the main screen, a list of Noise Hunts appears (see screen 1 in figure 5.11). Only Walk in the Park has been implemented for the test users. The challenge is to get to the city park and record at 3 locations in the park separated by at least 50 metres. Once the Noise Hunt has been completed, screen 1 will show that the first Noise Hunt has been done by changing the colour of the button to green. From now on, the next Noise Hunt, which is Blitzkrieg, can be done. Yet, this Noise Hunt has not been implemented in the prototype.

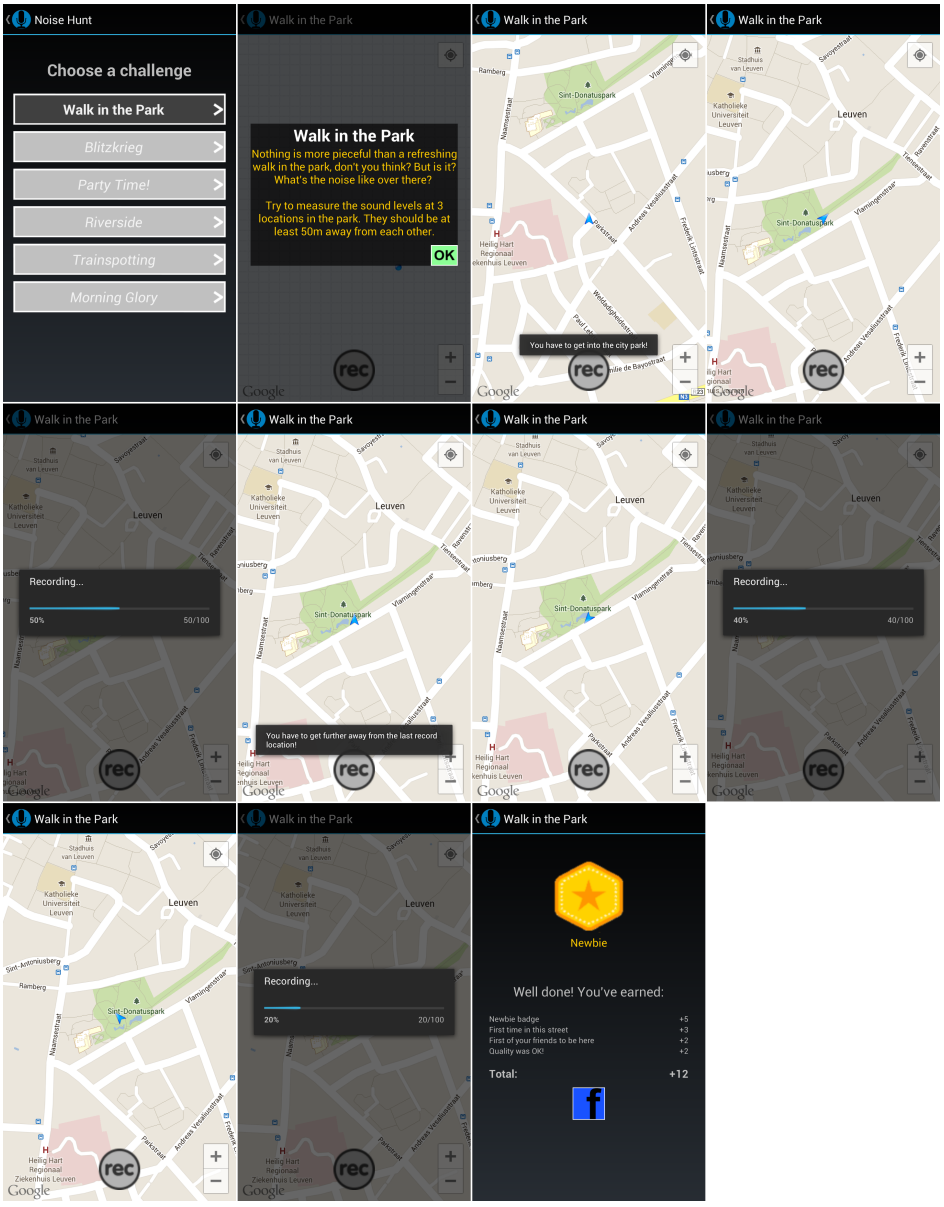


Figure 5.11: The Noise Hunt scenario of the full digital NoiseMApp prototype.

View Profile

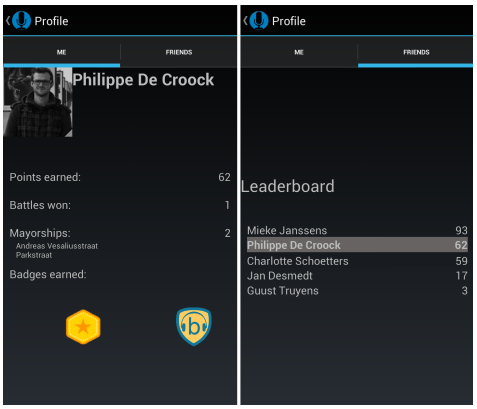


Figure 5.12: The View Profile scenario of the full digital NoiseMApp prototype.

The View Profile scenario can also be executed in the prototype. Because this scenario is exactly the same as in the paper prototype, the interested reader is referred to 3.3.5.

Show Map

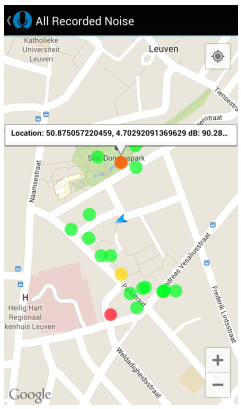


Figure 5.13: The Show Map scenario of the full digital NoiseMApp prototype.

The Show Map scenario has been fully implemented. Now all made recordings can be viewed on the map. They are represented with coloured dots: green for recordings below 55 dB, yellow for recordings between 55 and 70 dB, orange for recordings between 70 and 80 dB and red for recordings that exceed 80 dB. These colour codes are also used on the noise pollution maps of Antwerp, discussed in chapter 1.

5.2.3 Implementation Details

Since the first digital prototype, no fundamental change has been carried out to the code of the already existing scenarios. Only some of the buttons have been changed, which will be discussed in following sections. Here, only new implementation details are described.

Noise Recording Database Although no real recordings are being made with the prototype, the Show Map scenario should show noise recordings with their decibel levels on a map (see figure 5.13). For this, when pressing the record button in a recording scenario, a random sound level is generated and geotagged with the current location of the test user. This fake recording is stored in a noise database, which is a persistent SQLite storage structure in Java that holds the fake recordings with their locations. Even when restarting the application, the noise recordings keep being stored. This is done with the `SQLiteOpenHelper` class, which is extended to handle noise recording details.
Of course, this is just a temporary solution for the problem. In the final application, the database is stored on a server, so information can be shared through all instances of the application on different devices.

Google Places Google Places[6] has been chosen to return a list of closeby locations for a Sound Checkin. Although Foursquare API [10] is the most popular choice for this feature, Google Places seemed a lot easier to implement. One can just do an HTTP request to obtain all places in a specified radius of a specified coordinate. The result is returned as a JSON object, which proves to be a very handy format to extract data.

5.2.4 Evaluation

The evaluation of the final digital prototype is very similar to the evaluation of the first digital prototype. The prototype has been evaluated with different test users than the ones of the first digital prototype. They were again asked to think aloud to be able to take notes about their experiences with the application. During the evaluation, they were sometimes questioned about the clarity of everything on the screen.

After the evaluation, the applicants were asked to fill in a questionnaire so problems came to surface that had not surfaced during the evaluation. This questionnaire was exhaustive, but kept shorter than the one of the previous prototype.

Population

This time, nine people are selected to test the prototype. Five of them are students, none of them with a technical background. The other four are already working, of

[6] https://developers.google.com/places

which two have a technical background. The ages reach between 18 and 29. Six of them own a smart phone, of which three have Android installed on their devices, two have iOS and one person has a Windows phone. Two test users had also tested the paper prototype before.

Results

User Test No new big issues were identified during the user test. Every task could be successfully executed in an acceptable amount of time, which is better than in the previous prototype where users often asked what they had to do. The explanation pop-ups were deemed clear enough to let the users discover the features of the application on their own. Still, some suggestions were given by the test users directly or indirectly.

Opponents One of them was to let them see the opponent during a Sound Battle. It might increase the engagement during the game if they can see the opponent they are playing against. Also, it should be more clear that the test user is playing against someone else. As mentioned in the literature study, avatars can also be used for gamified applications to engage users. Such an opponent box can have the same function.

Green Sound Battle location marker A green location marker indicates that the record location is close enough to start recording. However, if the marker is green when you start a Sound Battle, it might give rise to confusion. One user thought that he had to go to that location first, before the other locations. This is not necessary, of course, but probably the most convenient. Nonetheless, this issue is rather small and should solve itself after using the application for a little while longer.

Show Map One test user indicated that it was rather confusing that the coloured dots in the Show Map scenario have the same colour code as the colours during Sound Battle. This could be easily solved by including a legend in the Show Map scenario.

Questionnaire First of all, all test users know Foursquare [10], which is an indication of whether they are familiar with the concept of badges and mayorships. However, only three of them are active users of that application.

Although all test users say that they do not play games very often, all test users indicate they like applications where they can earn badges. One might draw the conclusion that badges alone are not a fundamental fun or success factor for an application.

All test users indicate that they like applications that interact with the environment. Since NoiseMApp is such an application, the test might be a little biased

towards positive results.

44% of the test users found that the Noise Hunt was the most fun to play. This indicates that most users like challenging themselves in a game.
33% of the test users found that the Sound Battle was the most fun to play, which is an example of competition.
Nobody thought the Random Record was fun. These findings seem to contradict the findings extracted from the first digital prototype, where easy fun was liked the most.

SUS questionnaire Nine people have tested the full prototype, which means that the SUS questionnaire can be considered quite reliable. When evaluating nine people, 70% of the results can be considered correct [36]. The SUS score thus gives a rather good indication of the usability of the application.
In general, the mean score given to the application is 86 and the median is 87.5. This means the score has increased since the first, only partial, digital prototype, which is very good.

Conclusion

A first, partial prototype was made to speed up the design and evaluation process, which seemed very successful. No fundamental flow problems were discovered, which means that the prototype could be developed further without any major adjustments. A first indication was given about the usefulness of the application, which is rather positive. Even for people with no noise complaints, the application makes sense. This also means that the application is deemed interesting, which is very hopeful for letting people use the application to generate a complete noise pollution map of Leuven. The average SUS score is 81, which is the same as the paper prototype.

Then, a full digital prototype was made considering the results of the first evaluation. No big changes had to be made, but a lot of implementation still had to be done for the rest of the scenarios. Again, the prototype is positively evaluated. More insight was developed in what users thought was fun, which seemed to be the Noise Hunt. The SUS score averaged 86, which is more than the first digital prototype.

Also, a first indication was given that points and badges alone are not enough to make an application fun.

Chapter 6

Final application

This chapter discusses the full working application. Considering the results of the digital prototypes and other implementation problems, some changes have been carried out. Those changes are explained in section 6.1. The points and badges systems are discussed in sections 6.2 and 6.3. Section 6.4 shows limitations of the application that have not been worked out. Implementation details of the full application can be found in section 6.5. The full application has also been thoroughly evaluated, which is described in chapter 7.

6.1 Changes

In this section, the changes are presented that have been carried out since the full digital prototype. For screenshots of the final application flow that are different from the full digital prototype, the interested reader is referred to appendix C.3. The attentive reader might also notice that the application launcher icon has changed.

6.1.1 General changes

Badges Popup

Whenever a badge is earned, a popup is given with the earned badge. This was already the case when playing Sound Battles, but in the other recording scenarios the badge was shown on the points screen. To make the application more coherent, a popup has been introduced for every badge earned.

Decibel Level

Considering the suggestion of a test user during the evaluation of a digital prototype to return immediate feedback of the noise level after a noise recording, now the decibel level of the recording can be seen on top of the points screen after a recording. The decibel level is also colour coded, so an indication is given of the loudness of the recording. The same colour code is used as in the Show Map scenario, discussed in section 5.2.2, which is green for recordings below 55 dB, yellow for recordings

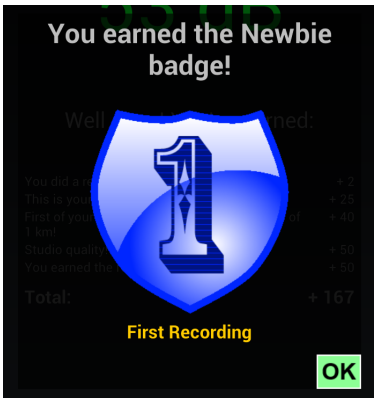


Figure 6.1: An example of a badge explanation in the final NoiseMApp application.

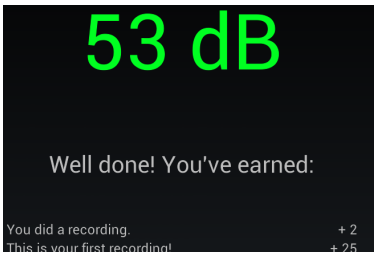


Figure 6.2: An example of the decibel level feedback in the final NoiseMApp application.

between 55 and 70 dB, orange for recordings between 70 and 80 dB and red for recordings that exceed 80 dB. An example of this is given in figure 6.2.

Info button

Figure 6.3: The info button in the final NoiseMApp application.

To explain the users the purpose of the thesis, an info button has been put in the top bar on the main screen (see figure 6.3). There, people can also get the developers email address if any problems would occur.

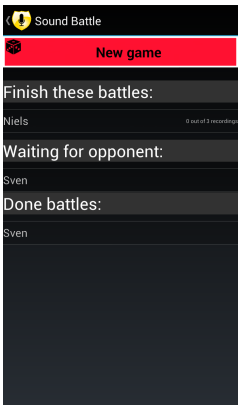


Figure 6.4: The Sound Battle creation screen in the final NoiseMApp application.

6.1.2 Sound Battle Changes

Sound Battle New Game

The first screen that a user sees when opening the Sound Battle scenario has changed totally. Now, only a new (random) game can be created (see figure 6.4). In the digital prototypes, the user had to choose between a random game and a game against a social network friend. Since it is not necessary for the application to work, the social network connection has been removed. In addition, Google Plus[1] easily provides account information of the person owning the device. However, it was not possible to access friends of that person in an equally easy manner. This is the reason why a Sound Battle can only be created against a random opponent for now.

Open, Pending and Finished Sound Battles

Already created, but unfinished Sound Battles can be reopened. A problem that was not discovered using the prototypes, was the possibility that users exit the application while playing a Sound Battle. It is possible while users have to walk to a Sound Battle location, they might open an other application, or even exit the NoiseMApp application. This might result in data loss, or an unfinished game without points.

To make this transparent to the user, the states of Sound Battles can be viewed in the first screen of the Sound Battle scenario (see figure 6.4). There are three states: open, pending and finished. Open Sound Battles still have one or more locations where the user has to record the sound. The amount of recordings to be made is also shown. When tapping an open Sound Battle, the application loads the stored data, so the user can play further with the game he left. In pending Sound Battles, all locations are recorded by the user, but the opponent still has to complete the Sound Battle. These Sound Battles can not be tapped. A Sound Battle is finished

[1] http://plus.google.com

whenever the two players have completed recording the three Sound Battle locations. A finished Sound Battle can be tapped to see the score screen.

Sound Battle Opponent

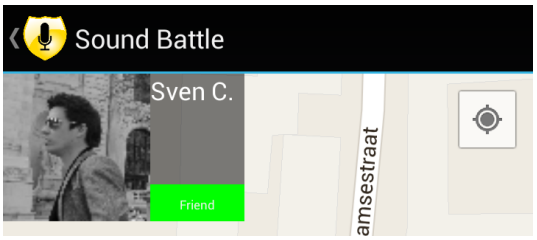
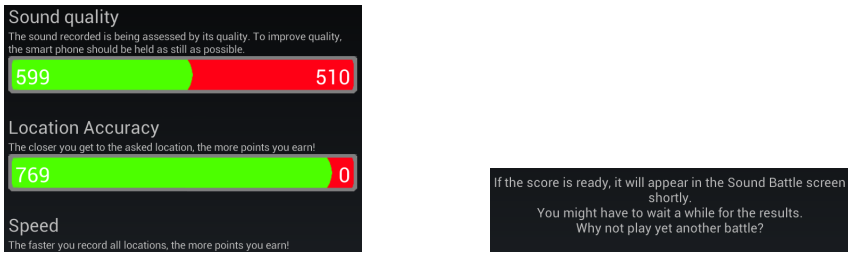


Figure 6.5: The Sound Battle opponent box in the final NoiseMApp application.

A significant part of test users of the prototypes stated that it is important to see who they are playing against. For this, a box is placed in the recording screen in the top left corner, showing a picture, a name and a button to add this person as a friend (see figure 6.5). The name is obfuscated, which means that only the first name is written followed by the first letter of the last name. Adding friends might be encouraging to play Sound Battles and compare results in the profile scenario.

Sound Battle Points screen



(a) The Sound Battle points screen in the final NoiseMApp application.

(b) The Sound Battle wait screen in the final NoiseMApp application.

Figure 6.6: New screens concerning the points in Sound Battle

As the evaluation of the prototypes showed many times, the three terms used to describe the scores were not that clear. To solve this, a description of every term has been written underneath the term to give more clarity, as shown in figure 6.6a. One problem came to surface that has not been discovered during the evaluations of the digital prototypes. If a player finishes the Sound Battle, but the opponent does not, the points screen cannot be shown as the score of the opponent can not be calculated yet. Therefore, a new screen (see figure 6.6b) is shown when this happens.

Push notifications

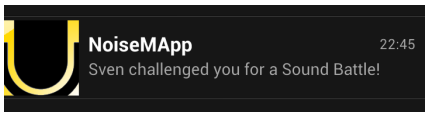


Figure 6.7: An example of a Sound Battle notification in the final NoiseMApp application.

Until now, people were not notified if they were being challenged by another user. The challenge would show up in the open battles, and the user would have to go check for them theirselves. To address this problem, which would hurt the engagement of using the application, push notifications have been introduced. Push notifications are messages sent to a device by applications, which show up in the top bar of the smart phone. The users of NoiseMApp receive a notification whenever they are being challenged, or whenever their opponent has recorded all three Sound Battle locations of a Sound Battle. An example of such a notification can be seen in figure 6.7.

6.2 Points System

Contrary to the prototypes, in a real application real points can be earned. A summary of all points that can be earned is given in following subsections. The point system is based on experience with other applications that give scores for checkins, for example Foursquare [10], which keeps the points to earn rather low. NoiseMApps points to earn are chosen to be higher, to amplify the difference between test users. Points are kept in the database. The total amount of points earned can be viewed in the profile.

Recordings made in the Sound Battle and the Noise Hunt do not contribute to the total points a user gains. Instead, bonus points are earned when winning a Sound Battle or completing a Noise Hunt. For this, the points of Random Record and Sound Checkin are discussed first.

6.2.1 Random Record

For every recording done, 2 points are earned, which is a quite low amount. Still, the user always gets something in return for the recording. Of course, bonus points can be earned, too.

Amount of Recordings

Every recording is stored in a database. This way, it is easy to see how many recordings every user has made. According to this amount of recordings, points are given as can be seen in table 6.1. The first recording returns a little bonus to get the

Amount of recordings	Amount of points
1	25
5	10
10	20
20	50
50	75
100	100
250	100
500	100

Table 6.1: Points received in function of the amount of recordings made by the user.

Radius (metres)	Amount of points
200	5
500	10
1000	20
2000	35
3000	70

Table 6.2: Points received in function of the radius of other recordings made by the user.

user going with the application. Then, gradually, the points to earn become higher when the amount of recordings is exponentially scaled up. A limit of 100 points is set so even active users still have the need to compete with other active users.

First Time in Radius

Points can be earned when users record for the first time in a specified radius. These points depend on the size of the radius, as can be read in table 6.2.

First of your Friends in Radius

If a user has friends, their recordings are also considered. In a similar way as in section 6.2.1, points are given according to table 6.4. The points to earn are doubled in comparison to the points to earn in table 6.2. This is done to encourage social relations, because when no friends are made, the user misses out on these bonuses.

Radius (metres)	Amount of points
200	10
500	20
1000	40
2000	70
3000	140

Table 6.3: Points received in function of the radius of other recordings made by the friends of the user.

Quality	Amount of points
Medium	5
Good	20
Super	50

Table 6.4: Points received in function of the quality of the recording made by the user.

Sound Quality

Sound quality is determined by measuring the movement of the device with the accelerometer. Implementation details are given in section 6.5. Points are given according to this quality. Since it is paramount that recordings are made with the device holding still for better quality, more points are given to super quality recordings.

6.2.2 Sound Checkin

All points that are described in the previous section are also applicable on the recordings made in a Sound Checkin. Additionally, the amount of Sound Checkins done contribute to the total points.

6.2.3 Sound Battle

As mentioned earlier, the recordings in a Sound Battle do not directly contribute to the total points of a user. Instead, once a Sound Battle has been completed, bonus points are given according to the amount of Sound Battles that have been won by the user, as shown in table 6.6. Because Sound Battles are more labour-intensive, the reward is bigger than Random Record recordings.

Amount of Sound Checkins	Amount of points
1	25
5	10
10	20
20	50
50	75
100	100
250	100
500	100

Table 6.5: Points received in function of the amount of Sound Checkins made by the user.

Amount of Sound Battles won	Amount of points
1	50
5	25
10	50
20	50
50	50
100	100
200	100

Table 6.6: Points received in function of the amount of Sound Battles won by the user.

Noise Hunt completed	Amount of points
Walk In The Park	50
Blitzkrieg	100
Party Time	150
Riverside	200

Table 6.7: Points received in function of the amount of Noise Hunts completed by the user.

6.2.4 Noise Hunt

Similar to the points system in a Sound Battle, recordings made during a Noise Hunt are not considered. When completing a Noise Hunt, bonus points are given according to table 6.7.

6.3 Badges System

Next to points, badges can also be earned when playing NoiseMApp. These badges are presented on the profile page of the user. This way, users can easily compare achievements and start to compete against each other.

6.3.1 Random Record

Figure 6.8: The badges to be earned for respectively 1, 50, 100 and 500 recordings made.

During Random Record, badges can be earned considering the amount of recordings made by the user. In figure 6.8, the badges are displayed.

6.3.2 Sound Checkin

Figure 6.9: The badges to be earned for respectively 1, 50, 100 and 500 Sound Checkins made.

When recording during Sound Checkin, badges described in section 6.3.1 can also be earned together with badges shown in figure 6.9

6.3.3 Sound Battle

Figure 6.10: The badges to be earned for respectively 1, 5, 20, 50, 100 and 500 Sound Battle wins.

In a similar way, badges are earned when winning Sound Battles. These badges are displayed in table 6.10.

6.3.4 Noise Hunt

Figure 6.11: The badges to be earned for completing respectively Walk in the Park, Blitzkrieg, Party Time and Riverside.

Last but not least, badges are earned when completing Noise Hunts as well as shown in figure 6.11.

6.4 Limitations

6.4.1 Mayorships

Mayorships are not implemented in the full application. Although earning mayorships was an idea from the beginning of the application, time was too short to implement it fully. Still, extra points are earned when being the first of your friends recording in a specified radius. More about this can be read in section 6.2
As a consequence, the profile screen has been changed, presenting the amount of Sound Battles won instead of the mayorships.

6.4.2 Noise Hunt

Only four Noise Hunts have been implemented: Walk in the Park, Blitzkrieg, Party Time and Riverside. Walk in the Park has already been explained in section 5.2.2.

Blitzkrieg The goal of this Noise Hunt is to make as many recordings as one can in 2 minutes. This makes it possible to quickly map close-by locations at the same time.

Party Time Party Time can only be completed during nights between 10 p.m. and 6 a.m. Also, the users have to go to a party location, which is Oude Markt in Leuven. There they have to make 3 noise recordings.

Riverside Users are asked to go near waters to record the sound on locations there. They have to be near the water and 3 recordings have to be made, separated by 200 metres.

6.5 Implementation

6.5.1 Environment

Android

The digital prototypes and the final application are all implemented in Android[2]. iOS[3] or Windows Phone[4] or even HTML5[5] are not chosen due to two main reasons.

First, since it is not possible to access the microphone of a smart phone through HTML5, this platform was not compliant with the needs of the NoiseMApp application [17]. The second reason is availability. Android is a very popular operating system for smart phones which takes about 80% [16] of the smart phone market share. This makes it easier to find test users to evaluate the application, which causes the test population to be larger. It is also easy to deploy applications on Google Play in comparison to other platforms, which was an extra plus.

During development, it has been chosen to only use the most recent API versions. Consequently, the application will only work on devices (smart phones and tablets) with an Android version number equal to or higher than 3.0.

Development Tools

Eclipse To develop Android application, the Android Software Development Kit[6] (SDK) has to be installed. The SDK has been set up on a Windows 7 (64-bit) computer and the Eclipse[7] development tool has been used to write the code. The Eclipse version used is Helios. Plugins are available for the Android SDK in order to simplify the development. No installation problems have occurred during the whole development cycle.

Github The source code has been backed up and versioned on Github[8]. Github is a software service, based on the Git revision control system. The project has been stored on a free account, which means the project is licensed as Open Source. The interested reader is referred to the source code revisions of NoiseMApp on Github[9]. The final NoiseMApp project is also on Github, see appendix F.1.1.

[2] A mobile operating system, developed by Google. (http://www.android.com)
[3] A mobile operating system, developed by Apple. (http://www.apple.com)
[4] A mobile operating system, developed by Microsoft. (http://www.windowsphone.com)
[5] The fifth version of HTML that supports additional features for multimedia.
[6] http://developer.android.com/sdk
[7] http://eclipse.org
[8] http://www.github.com
[9] https://www.github.com/Phille88/NoiseApp

6.5.2 Android Application

In this section, the anatomy of the Android application is described. First, a general explanation is given of the structure of an Android project in section 6.5.2. Then, in section 6.5.2, the architecture of NoiseMApp is further elaborated.

Android Project Files

Android applications are made up of generally four types of files: self-written source code, generated source code files, resources and the Android manifest file.
The self-written source code holds all the code that is needed to let the application be dynamic. Since Android is mostly Java, this folder only contains Java classes.

When writing (and saving) the source code, other Java files are automatically generated by the Android SDK. These files are needed to link the Java source code files to resources. The class `R`, for example, holds all references to images, sounds, etc. This way, when writing source code, the call `R.drawable.name_of_image` can be used to hold a reference to an image that is placed in the resource folder. When changing layouts dynamically, like changing text, these references are necessary.
Configuration details for building are also generated in the `BuildConfig` class.

The `AndroidManifest.xml` file is generated too. It holds all the permissions needed by the application. All activities are also referred to in the Android manifest. For example, the different `Activities` (which will be discussed below) are written in the Manifest with their parent activities. Also the launcher activity, which is the activity the user sees when starting the application, is defined in the manifest.

Finally, there is the resource folder, which holds all files that are needed for the layout of the application. Files include images, sounds, layout files, text string values and dimension values. Layout files, string values and dimension values are all written in Extensible Markup Language (XML). When assigning id's to every object, they will get a unique reference in the `R` class, which is handled by the Android SDK. Further implementation details will be given in following subsections.

Activity In Android, every screen a user sees, is called an `Activity`. An `Activity` is a Java class that is related to an Android Layout XML file, which represents how the static part of that `Activity` screen looks like. In the `Activity` class, the behaviour of the screen is coded. For example, if a button needs to be clickable so an operation can be done, an `OnClickListener` has to be attached to that button through the source code of that `Activity`. Also the flow from one `Activity` back to another is defined here.

FragmentActivity When tabs are needed, like in the scenario of View Profile, a `FragmentActivity` needs to be implemented. All tabs have to be implemented by using `Fragments`. To let the these fragments behave as tabs in the `FragmentActivity`,

a SectionPageAdapter has to be implemented.
FragmentActivities replace the deprecated, but more describing `TabActivity`.

ListActivity Whenever items need to be listed, a `ListActivity` is implemented. For this, every item in such a list has to be implemented as a `ListItem`. To let such an item behave when being clicked, an `ArrayAdapter` class is needed, which couples the items to the list.

AsyncTask `AsyncTask`s offer the developer a structured way to handle multithreading in an Android application, without having to bother about the details. There are three main methods that are of importance when implementing an `AsyncTask`: `onPreExecute`, `doInBackground` and `onPostExecute`. The `onPreExecute` method handles all code that has to be done before the task is executed. When this is done, a new thread is created which executes the code written in the `doInBackground` method. Afterwards, the main thread is synchronized again with the code written in the `onPostExecute` method.
When implementing `AsyncTask`s, one should be aware that `onPreExecute` and `onPostExecute` live in the main thread and `doInBackground` does not. Due to this, activity references in `doInBackground` often cause errors.
The AsyncTask has been used a lot in this project and is very useful to maintain a fluent UI while doing labour-intensive tasks.

AndroidManifest As mentioned earlier, the Android manifest holds all permissions that are needed for all functions in the application to work properly. If functions need permissions that are not included in the manifest, errors occur during execution. Permissions needed by the application are shown to users when installing the application through Google Play. All persmissions needed for NoiseMApp are included in appendix F.1.3. All activities are also included in the manifest with their parent activities. If one does not allow an activity to have a landscape orientation, this should also be stated here.
Finally, a receiver has to be defined for the push notifications to work, called `com.google.android.gcm.GCMBroadcastReceiver`.

Architecture

The source code has been structured in different packages. An overview of the NoiseMApp packages can be found in E.1.
These packages are described below and their class diagrams can be found in appendix E. In these subdiagrams, only attributes, methods and relations to other packages that matter for the explanation are shown of the package of which the implementation is discussed. Layout files, generated files and the Android Manifest are not discussed.

be.kuleuven.noisemapp As shown in the class diagram E.2, there is a `MainActivity`, which represents the main screen of the application. The main screen is the first

screen the user sees when opening the application. From there, the user can navigate to other screens, or activities, which are described in other subpackages below.

The `WaitForLoadingTask` is an asynchronous task that shows a progress circle until registering of the user is done and necessary information has been obtained from the Google server and the NoiseMApp server.

The other classes in this package are responsible for the Google Cloud Messaging[10] service. The implementation is derived from samples of the Android SDK. GCM makes it possible to send push notifications to users of the NoiseMApp application. Every user gets registered to the GCM through the `GCMRegistrar`. The `WakeLocker` class makes it possible to wake up the smart phone when receiving a notification.

be.kuleuven.noisemapp.tools The `tools` package is explained first, because a lot of helper classes discussed here will be used in other packages. Downloading images from a webpage is done by an execution of the `ImageDownloaderTask`. When these images have to be scaled, the `BitmapScaler` takes care of this.
The `QuickSort` class is a generic algorithm to sort any type of object in a list, following the Quicksort algorithm.
Then, there are static classes where constants are defined, such as `JSONTags`, `MySQLTags`, `MemoryFileNames` and `Constants`. These classes make it possible to define a name for respectively a JSON tag, a MySQL parameter, a file name of a file in the application memory or other constants needed by the application to be only defined once.
For tasks that handle JSON answers, the `JSONParser` helper class converts these to Java readable code.
Android only supports primitive values to be stored in the application memory. Therefore, `ObjectSerializer` can convert objects to strings and the other way around to work around this issue.
Classes implementing the `iAfterTimer` interface, have to implement a method `afterTimer`. When a `TimerTask` is executed, after a specified amount of time, the method `afterTimer` gets called and the code of that method is executed.

be.kuleuven.noisemapp.auth The `auth` package handles authorization of the user owning the phone. When a user starts the application for the first time, the `GetInfoInForeground AsyncTask` is executed, which is a subclass of the `AbstractGetInfoTask`. These classes can be found in the Android samples folder of the Android SDK. They fetch user information from the Google Plus servers. By doing this, the user will only once be asked to give permission to access the Google Plus account and no login has to be done.
The information retrieved is stored in a `UserDetails` object, which is stored locally. Of course, the information is also stored remotely on the NoiseMApp server. For

[10] A service of Google to send messages through the cloud. http://developer.android.com/google/gcm/index.html

this, the `CreateUserProfileTask` is executed, which is an AsyncTask.
When a user starts the application again afterwards, or whenever users check their profile, the local profile is updated by the `UpdateLocalProfileDetailsTask`. All data from remote servers is received in JSON format, which explains the relationship with the `JSONParser` class of the tools package.

be.kuleuven.noisemapp.friends In the package `friends`, classes which relate to friendship management in the NoiseMApp application are grouped. `AddFriendshipTask` and `DeleteFriendshipTask` are `AsyncTask`s that are responsible for respectively adding a friend or deleting a friend. The `UpdateLocalFriendsListLocalTask` will update the friendslist locally, so loading friends related objects, like the leaderboard, can happen faster.

be.kuleuven.noisemapp.recording Whenever a user wants to record sounds in each scenario, similar activities are started. The `RecordActivity` superclass holds all similar characteristics of the record screen users see when they want to record sound. The showing of the map, available through the Google Maps API, is also implemented in this class. The record function is encapsulated in the `RecordingTask` which handles the recording of the decibel level and the storing of this decibel level in the database. The decibel level is measured by making a recording of 10 seconds, which is temporarily stored on the phone. During this recording, every fourth of a second the maximum amplitude is measured of that interval, which is a function available in the Android SDK. These maximum amplitudes are then converted to decibels according to following formula: $20 \log_{10}(\frac{amp_max}{amp_ref})$ [35]. Finally, these decibels are averaged so one decibel level can be stored.
Also the readings of the accelerometer are measured, so a measure for the sound quality can be assigned.
Internally, the recordings are captured in `NoiseRecording` objects, so they can easily be passed to other operations, like calculating points. Then, the recording is stored by executing the `SaveNoiseRecordingRemoteTask`.

be.kuleuven.noisemapp.points Every time a recording is made, points can be earned. These `Point` objects have a description and an amount. All points related to a recording is stored in a `RecordingPoints` object that is attached to a `NoiseRecording` after the points are calculated. Similarly, a `Badge` is used internally. The `Badges` class stores all constants and badge ids that are necessary to communicate with the server.

be.kuleuven.noisemapp.randomrecord The `randomrecord` package holds all classes that are needed for the Random Record scenario. First, there is the `RandomRecordActivity`, which is a subclass of the `RecordActivity` discussed earlier. `RecordingTask` is also sublassed by `RandomRecordTask`. `CalculateRandomRecordPointsTask` will return points that are shown on the points screen, which is handled by `RandomRecordPointsActivity`.

be.kuleuven.noisemapp.noisehunt The first screen a user sees when opening the Noise Hunt scenario is captured in the `NoiseHuntActicity` class. For every Noise Hunt implemented, there exists a `NoiseHuntRecordActivity` and a `NoiseHuntRecordTask`. The `RecordActivity` and `RecordingTask` subclasses are needed to represent respectively the screens with the map and the recording function as discussed before. Additionally, every recording made in the Noise Hunt scenario is temporarily stored in the application (which explains the coupling with `NoiseLocation`) and is also stored in the remote database table `noisehuntrecordings`, which is discussed below, by the `SaveNoiseHuntRecordingTask`.
The `SaveFinishedNoiseHuntTask` saves the last Noise Hunt completed in the table `noisehuntstate`, which is also discussed below. `GetNoiseHuntStateTask` then reads the table and will make the right Noise Hunt available to play.
Calculating points is done in the `CalculateNoiseHuntPoints` class and these points are represented in `NoiseHuntPointsActivity`.

be.kuleuven.noisemapp.soundbattle The `soundbattle` package is the most complex of the application. The Sound Battle scenario starts with showing the `SoundBattleActivity`, which loads all Sound Battles created by the user by executing `GetAllSoundBattlesTask`. To enlist all retrieved battles, the `iSoundBattleListItem` interface is implemented by `SoundBattleItem` and `SoundBattleHeaderItem`, which represent an earlier created Sound Battle or a header of the list respectively. The `SoundBattleItemAdapter` is implemented for the behaviour of these items.
When users choose to start a new game, the `CreateRandomSoundBattleTask` is executed, followed by the execution of `CreateSoundBattleTask`. The former chooses a random user and passes it to the latter to create a new Sound Battle. Both could be done in one AsyncTask, but these tasks were created in the beginning of development and have not been optimized later. Alternatively, when users choose a Sound Battle that has been created earlier, the `LoadSoundBattleTask` is executed.
Similar to other scenarios, `RecordActivity` and `RecordingTask` are inherited from. `SoundBattleLocation`s represent the locations internally where users have to record during a Sound Battle. They are generated by the `SoundBattleLocationGenerator` in a random fashion. The generated locations are stored in the database by the `SaveSoundBattleLocations` AsyncTask. Each recording is also separately stored in the `soundbattlerecordings` table, discussed in subsection 6.5.3, by the `SaveSoundBattleRecordingTask`.
When a Sound Battle is over, points are calculated by `CalculateSoundBattlePoints` after which the `SoundBattleWaitActivity` shows. When both players of a Sound Battle have completed the three locations, they can check the score on a screen implemented by `SoundBattlePointsActivity`.
Because locations had to be represented on the map in a Sound Battle, the `NoiseLocation` was implemented. Later, this class was also used in other scenarios, like the Noise Hunt. Unfortunately, the NoiseLocation class has not been revised, which causes the strange and unnecessary coupling in the diagram.

be.kuleuven.noisemapp.soundcheckin The Sound Checkin scenario is captured in the `SoundCheckinActivity` class, which is a subclass of the earlier discussed `RecordActivity`. Equivalently, `SoundCheckinRecordTask` is a subclass of `RecordingTask`, with an additional task of storing the recordings in a separate table, discussed in section 6.5.3. Places that are available through Google Places are listed in this activity. For this, a `PlaceItem` and `PlaceItemAdapter` are implemented. Every time a recording is stored, its points are calculated by `CalculateSoundCheckinPoints` and checkins are separately stored by the `SaveSoundCheckinRecordingTask AsyncTask`.

be.kuleuven.noisemapp.profile All classes in this package are responsible for the View Profile scenario. The `ViewProfileTabActivity` (which is actually a subclass of FragmentActivity) is an activity that exists of two fragments: `MeFragment` and `FriendsFragment`. A `FragmentActivity` is an Android class which makes it possible to have tabs in an application through an adapter, which is the `SectionsPagerAdapter`. To enlist friends in a `ListView`, like in the leaderboard, a `FriendItem` and a `FriendItemAdapter` has been implemented.

be.kuleuven.noisemapp.showmap Finally, the `showmap` package handles the Show Map scenario. Although no recording can be done in this scenario, the screen that shows all noise recordings is very similar to a `RecordActivity`, hence the inheritance. All noise recordings are queried by the `GetAllNoiseRecordingsTask`.

6.5.3 Database Support

For the backend of the application an Apache server with a MySQL database is used. MySQL (My Structured Query Language) is the most popular relation database management system used. Through SQL queries, interactivity with the database is possible. For communication with the database, PHP scripts are used to build up queries and send them to the database.

MySQL and PHP

To focus the development on the application, MySQL has been chosen as database back-end. Other database systems are indeed possible and might be more fitting, as would be Google's BigTable because of its scalability.

In figure 6.12, an overview is given of the tables in the NoiseMApp database. In subsequent paragraphs, their responsibilities are described.

userprofiles The table `userprofiles` holds all information necessary of the users who have installed NoiseMApp. Every user has a unique `userID`. Emails are stored in the database to send the questionnaire at the end of the evaluation to their address. The emails are also necessary to retrieve other information from Google, like the users first name, last name, and Google ID. With the Google ID the Google+ profile picture can be retrieved. Because the Google ID is a number of 21 characters, it is split up in two parts. This was necessary at first, since there is no larger integer

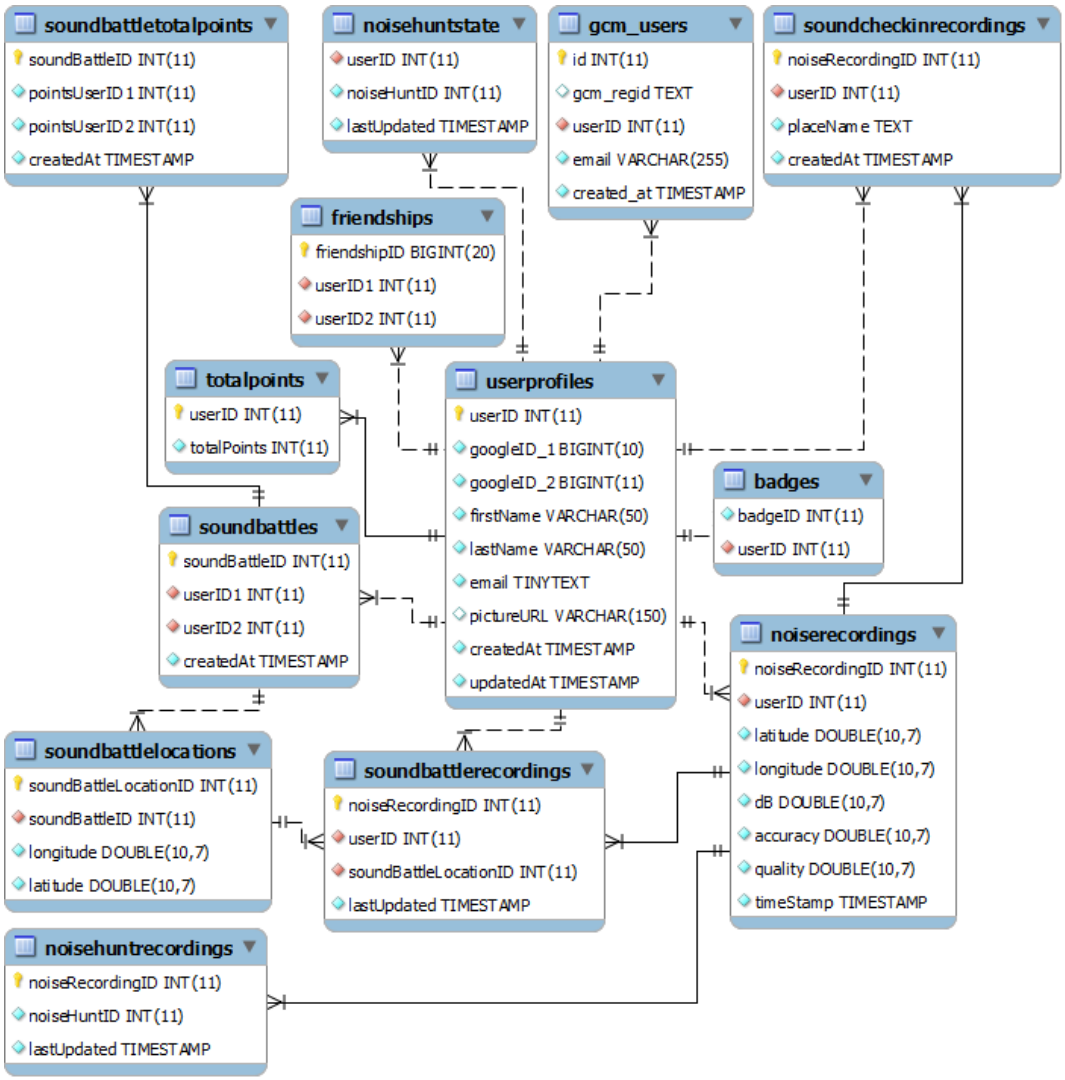


Figure 6.12: Overview of the MySQL database.

possible than BIGINT(19) and the Google ID served as index. Later, it has been chosen to use other indexes, which made this logic obsolete. This of course asks for extra logic in the PHP files to combine the two parts to one Google ID and the other way around.

friendships Between users, friendships can be made. Friendships are one directional, which means that if user a friends user b, user b does not have to be a friend of user a. This is a familiar concept for users of Google Plus, where friends can be added to circles without approval. Friending is only possible during a Sound Battle and users can only see profiles of the users they have friended. To do this, friendships are stored in the `friendships` table.

totalpoints The sum of all points earned by the user during the different scenarios are stored in the `totalpoints` table. Every time a user makes a recording and earns points by doing so, the total of points of that user is increased.

badges All badges earned by a user are stored in the table `badges`. This is necessary to show the badges in the profile.

noisehuntstate Whenever a user completes a Noise Hunt, the Noise Hunt id and user id are stored in the `noisehuntstate` table. This way, it is easy to check which Noise Hunts have been completed already, in order to clear the right Noise Hunts in the Noise Hunt screen.

noiserecordings All noise recordings made are stored in the table `noiserecordings`. These include the noise recordings of Sound Battles and Noise Hunts, too. For every noise recording, the latitude, longitude, decibel level, GPS accuracy in metres and quality is stored. Obviously, noise recordings have unique `noiseRecordingID`s

soundbattles To let users compete with each other in Sound Battles, their userIDs have to be stored in the `soundbattles` table. Every Sound Battle has a unique ID.

soundbattlelocations The Sound Battle locations that are created during the creation of a Sound Battle are also stored in the database with their longitude and latitude in the `soundbattlelocations` table. The `soundBattleID` column is needed to refer to the correct Sound Battle.

soundbattletotalpoints When a user has recorded at all three locations, points are calculated and stored in the `soundbattletotalpoints` table. These scores are used to check how many Sound Battles have been won by a particular user, which is shown in the users profile.

soundbattlerecordings All recordings made during a Sound Battle are stored in the `soundbattlerecordings` table. Every recording is stored with the ID of the nearest Sound Battle location in the field `SoundBattleLocationID`. It is necessary to know which Sound Battle location markers can be turned to a check mark in the Sound Battle scenario. Also, this way, it can be determined what percentage of all made noise recordings have been made with playing Sound Battles.

noisehuntrecordings For the same reason, the values of recordings made during a Noise Hunt are also stored in the `noisehuntrecordings` table.

soundcheckinrecordings All recordings made in the Sound Checkin scenario are also stored in the `soundcheckinrecordings` table. The `placeName` field holds the name of the place the user checked in to, which is necessary to depict on the users profile.

gcm_users In order to make it possible to send push notifications to users, some information has to be stored in the table `gcm_users`. The `gcm_regid` field is generated by Google and is needed to identify a specific device to which notifications can be sent. The `userID` field is needed to link this device to a NoiseMApp user.

PHP is used to populate the MySQL tables and to query the database. PHP version 5.0 is used. Every database query is written in a separate PHP file. These PHP files are executed through a HTTP request of the application by an asynchronous task.
The scripts can be found in appendix F.2.

Conclusion

The full application is implemented in Android with all game techniques and elements proposed in chapter 3. Regarding the evaluation of the last digital prototype, the design of the application has not changed much. Yet, some changes have been carried out in order for the Sound Battle scenario to work properly. A points and badges system has been thought out and created. Further, the implementation of the application has been described.

Chapter 7

Evaluation

The full application has been evaluated. This evaluation stresses the engagement of users more than before. User interface issues are assumed to be solved at this stage because of the evaluations of previous iterations. Still, the usability of the application is evaluated again.

7.1 Setup

NoiseMApp is available on Google Play, which makes it possible for the test users to download the application on their own phone. To develop a better understanding whether game elements and game techniques work, an other version of the application has also been made, which is named NoiseMAppR. With NoiseMAppR, one can only do one recording at a time, just like in the Random Record scenario of NoiseMApp. The only difference is that no points or badges can be earned by using it. No friends can be seen and no games can be played. Yet, it is still possible to see all recordings made with the NoiseMAppR application. NoiseMAppR is in this sense a non-gamified and non-social version of NoiseMApp. It is also available for download on Google Play.

Evaluation method Two groups took part in the evaluation. One group tested the NoiseMApp application, which is referred to as the NoiseMApp group, the other group tested the NoiseMAppR application, which is referred to as the NoiseMAppR group. There were no users that took part in both groups. Both groups were asked to install the application and leave it installed for at least 2 to 3 weeks. They were asked to use the respective application at least 3 times on different days. After the evaluation period, they were given a questionnaire they had to fill in. This questionnaire (see appendix D) consists of five parts:

1. demographic information;
2. game behaviour;
3. environmental characteristics;
4. NoiseMApp experiences;
5. SUS questionnaire.

The NoiseMAppR group had to fill in the same questionnaire with the exception of the fourth part, which was adapted to their experiences with NoiseMAppR. The usability of NoiseMAppR is relevant, for it is interesting to see how the two versions compare to each other.

7.2 Population

In total, 29 people took part in the evaluation, of which 17 people were in the NoiseMApp group and 12 people in the NoiseMAppR group. Through Facebook[1] public messages were posted to find users that have an Android device with version 3.0 or higher. Facebook is a powerful medium to reach many people. The people that wanted to participate were asked if they were regularly or living in Leuven. If they were, they were asked to test the NoiseMApp application. Sound Battle and Noise Hunt can only take place in Leuven, because they were only implemented for Leuven. If they were not regularly in Leuven, they were assigned to the NoiseMAppR group. One person of the NoiseMApp group had lost her phone after 9 days of account creation and thus could not finish the whole evaluation. She still did fill in the questionnaire since she is one of the few users of the Sound Battle, which will be discussed below. One person of the NoiseMAppR group had uninstalled the application, because it crashed during a recording after 12 days of account creation. He has not filled in the questionnaire. In total, 16 people of the NoiseMApp group and 11 people of the NoiseMAppR group filled in the questionnaire, which is 93% of the total test population. The high rate makes it possible to make abstraction of this in further results.

7.2.1 NoiseMApp Group

Although it was asked in advance that test users of the NoiseMApp group had to be resident or regularly in Leuven, only 56% said afterwards to be working or living in Leuven. 31% was in Leuven once a week or less.

88% of the people are between 22 and 25 years old. The other 12% were 29 and 30 at the time of the evaluation. Only 14% are female. The distribution of working and studying test people is fifty-fifty. 63% have a technical background. The population is overall considered capable of using Android phones, since 81% said to be owning an Android phone for more than six months. Finally, only 12% of the participants had tested a paper or digital prototype of the application before.

Game Behaviour 50% of the test users of the NoiseMApp group play games once a week or more on their smart phone. This subgroup is referred to as the gamers, the other are non-gamers. One person never plays games. Although 81% said to be knowing applications (not necessarily on smart phones) that use points, badges or

[1] A social medium (http://www.facebook.com)

achievements and a leaderboard (PBLs) only 51% uses such applications. The latter is called the PBL group in further results.

Environmental Characteristics 75% of the test users of the NoiseMApp group is aware of noise pollution and 69% said to be suffering from this occasionally or more. Although nobody had ever used noise mapping applications before, 44% thinks they are useful and 44% gave a score of 3 on a scale of 1 (strongly disagree) to 5 (strongly agree) to this statement.

7.2.2 NoiseMAppR Group

91% of the people of the NoiseMAppR group are between 22 and 25 years old. One user is 32 years old. In the NoiseMAppR group, 27% is female. 73% of the people are students, while the others are already working. 64% have a technical background. Moreover, 91% says to be owning an Android phone for more than six months. 2 different people had tested a prototype of NoiseMApp before.

Game Behaviour In contrast to the people of the NoiseMApp group, 82% play games once a week or more on their smart phone, of which 33% play once a day or more. One person never plays games. Similar to the NoiseMApp group, 91% said to be knowing applications (not necessarily on smart phones) where they can earn points, badges or achievements, but only 45% uses such applications.

Environmental Characteristics All people of the NoiseMAppR group said to be aware of noise pollution. 91% of them said to be suffering from it occasionally or more. Again, nobody had ever used noise mapping applications before, 73% thought it is useful.

7.3 Results

7.3.1 Noise Level Map of Leuven

In figure 7.1, the noise level map generated with the NoiseMApp application is shown. In total, 98 recordings were made, of which 91 were made in Leuven and Heverlee. Most recordings were made near the very center of Leuven.

7.3.2 Application Usage

NoiseMApp vs. NoiseMAppR

Amount of recordings A first measure to see whether the application reaches its goal, is to check the amount of recordings made by both groups. In total, the NoiseMApp group made 98 recordings, while the NoiseMAppR group only made 48. This means that on average 5.76 and 4 recordings were made per user in the

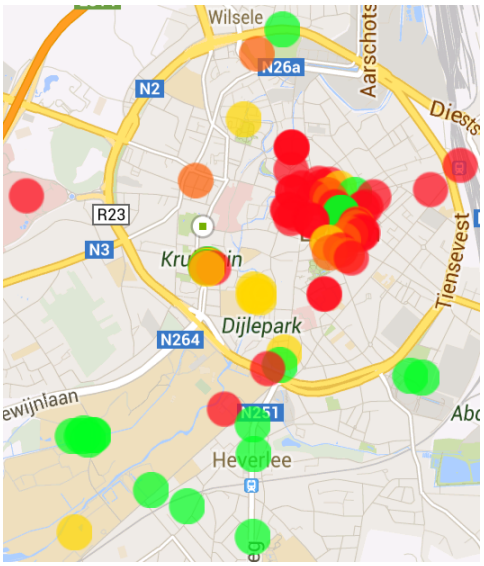


Figure 7.1: Noise level map of Leuven generated by users of NoiseMApp.

NoiseMApp and NoiseMAppR group respectively. It seems that a gamified application like NoiseMApp is more likely to generate a noise level map of Leuven.

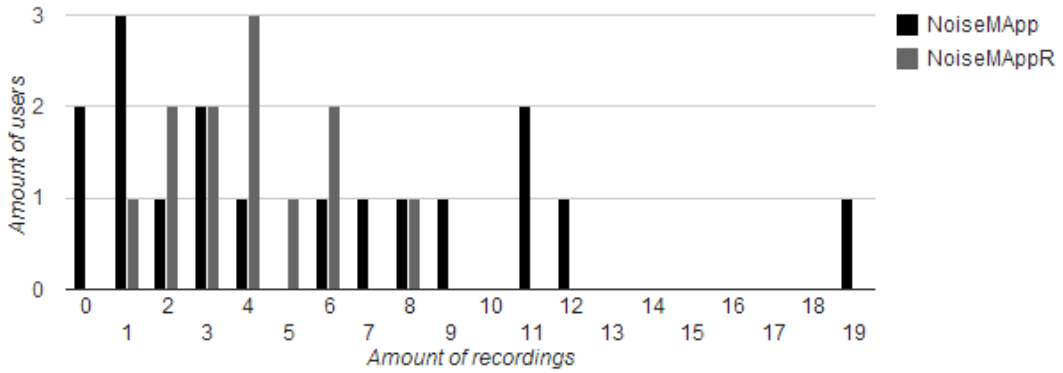


Figure 7.2: Amount of users per amount of recordings of NoiseMApp and NoiseMAppR.

When looking at figure 7.2, one might notice that 2 people (both non-gamers) did not contribute to the recordings of the NoiseMApp application. If one only looks at the average amount of recordings per active user, the average of NoiseMApp even increases up to 7 recordings per active user. For both applications, most users took 4 or more recordings. The maximum amount of recordings made by one user is only 8 with the NoiseMAppR application, whereas for the NoiseMApp application it is more than double. There are also more highly active people in the NoiseMApp group. This seems to indicate that the NoiseMApp application can be very effective for a

couple of users. The two most active users are gamers, although they both do not belong in the PBL group.

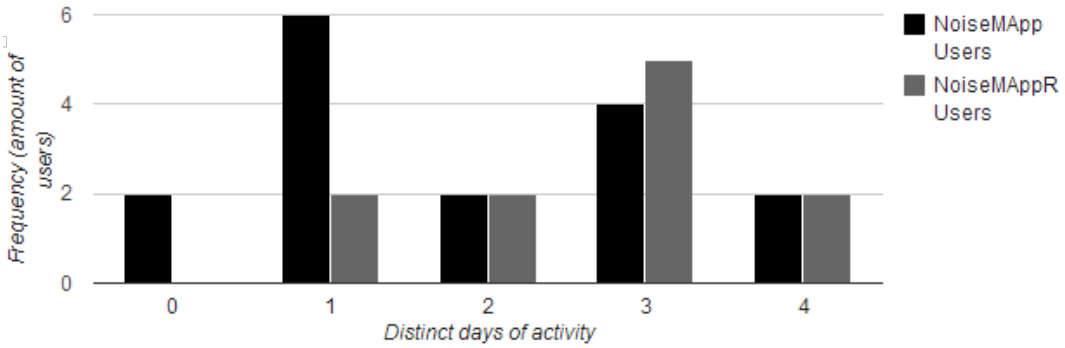


Figure 7.3: Amount of users in function of their activity on distinct days of NoiseMApp and NoiseMAppR.

User Activity The users were asked to at least use the application 3 times on different days. Yet, figure 7.3 shows that only 35% of the NoiseMApp users and 58% of the NoiseMAppR users did. Only 12% used the NoiseMApp application more than suggested, whereas for NoiseMAppR this is 17% of the users.
The gamers that recorded 19 and 12 times, both only had respectively 2 and 1 distinct days of activity. The NoiseMApp application seems to motivate bursts of recordings instead of a longer engaging activity. No significant difference between gamers and non-gamers was found.

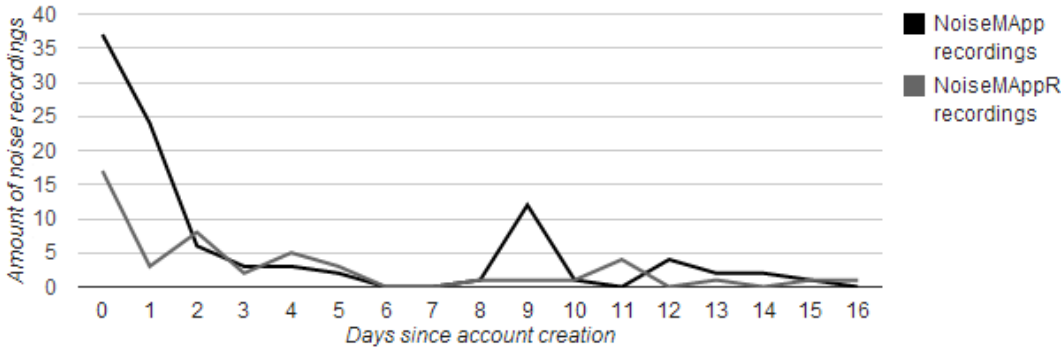


Figure 7.4: Amount of users per amount of recordings of NoiseMApp and NoiseMAppR.

Persistence In figure 7.4 the amount of all recordings made since the creation of the account of the recorder is shown. It is clear that most activity of the users was in the beginning. Yet, there is 1 peak at day 9 for NoiseMApp, but that is caused by a user that just started using the application. The last activity with NoiseMApp

happened at only 15 days after account creation, whereas for NoiseMApp this is 16. Long term engagement is not possible to derive with an evaluation period of only 2 to 3 weeks, but the figure shows that activity tends to decrease.

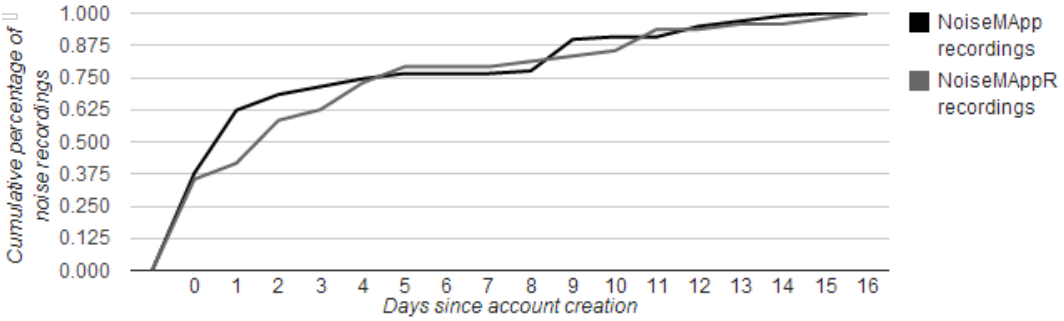


Figure 7.5: Cumulative function of the fraction of recordings for NoiseMApp and NoiseMAppR.

The cumulative recording activity of both NoiseMApp and NoiseMAppR is displayed in figure 7.5. One might notice that the activity in NoiseMApp at first is more steep than the NoiseMAppR application. It seems the test users are more interested in using the NoiseMApp application than NoiseMAppR in the beginning. Still, after 5 days this distinction is negligible.

NoiseMApp Usage

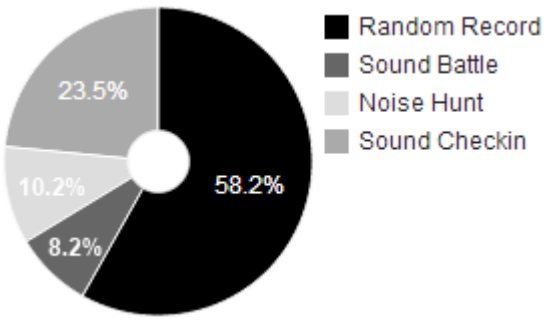
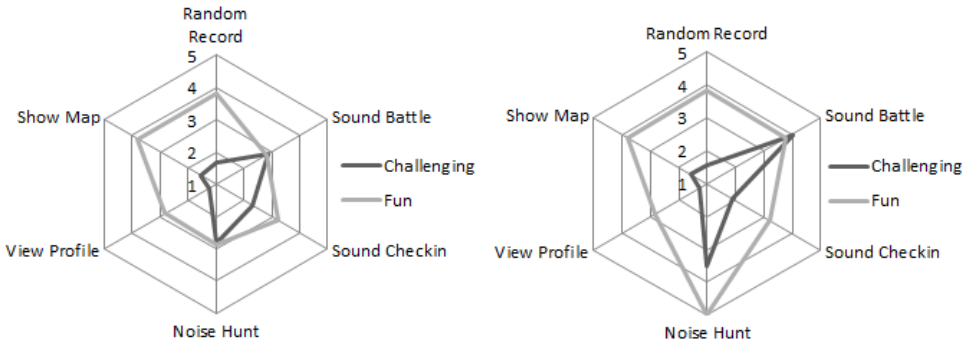


Figure 7.6: Amount of recordings per feature of NoiseMApp.

Game Features Figure 7.6 shows a pie chart of the amount of recordings per feature. It is clear that the Random Record scenario was the most popular with 58.2% of the recordings. Sound Checkin was also popular with a fourth of all recordings made. Contrary to the Sound Battle and Noise Hunt, in which multiple noise recordings are made per execution of the scenario, the Random Record and Sound Checkin only make one recording per execution. Although 30 Sound Battles have been created, none of them have been completed by both players. Only 2 players have finished the Sound Battle and in total 4 players of the total 17 made recordings

during Sound Battle.
The Noise Hunt yields a similar disappointing result. Only 2 people made recordings for the Walk in the Park Hunt. 4 recordings were made by 1 player in the Blitzkrieg Hunt.



(a) Perceived by all users (b) Perceived by 'accurate' users

Figure 7.7: All features of NoiseMApp with their respective fun and challenging score.

One of the reasons for the results in previous paragraph is shown in figure 7.7a. The test users were asked to give a score from 1 to 5 to every feature concerning the fun they had with it and how challenging they thought it was. It seemed none of the features were perceived very challenging, however, gamers did find it a little bit more challenging than non-gamers. The same holds for users of applications with PBLs. Random Record has been most chosen as the most fun, since it was easy to use and could be executed wherever they wanted. Also, the non recording features was deemed fun, as they show the progress the personal score or the whole noise level map. Sound Battle was the less fun of all scenarios.

However, three notes should be made with these results. First, the different features have not been used by everyone. Therefore, figure 7.7a might give bad results, because a feature is not used. Some users gave a score of 1 (strongly disagree) to a statement regarding a feature because they did not use it. In this question, it was not clear to the participants that a score of 3 meant neutral, which would have been a better option. Because a reason was asked why they gave that score, this problem could be discovered. Second, the term 'challenging' was interpreted by some users as a usability measure, rather than a game factor. The results to these questions can be perpendicular to each other. In figure 7.7b, the answers of people that have not used the feature and of people answering the questions with an other interpretation of the word challenge have been left out. The remaining people are referred to as the 'accurate' users.
It is clear that for the users who have at least used the features once, the features seem more fun and even more challenging. Third, only 56% of the users of the

NoiseMApp were working or living in Leuven, which might also be an important reason for the bad results of Sound Battle and Noise Hunt, while it was necessary to be in Leuven when playing these scenarios. Strangely, the only people using Noise Hunt were at most 1 time per week in Leuven.

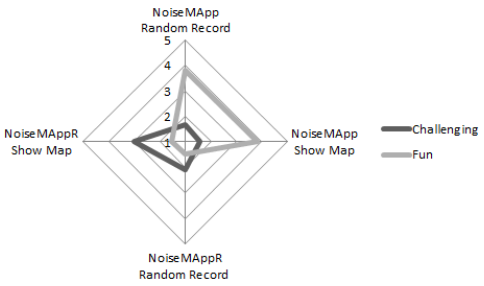
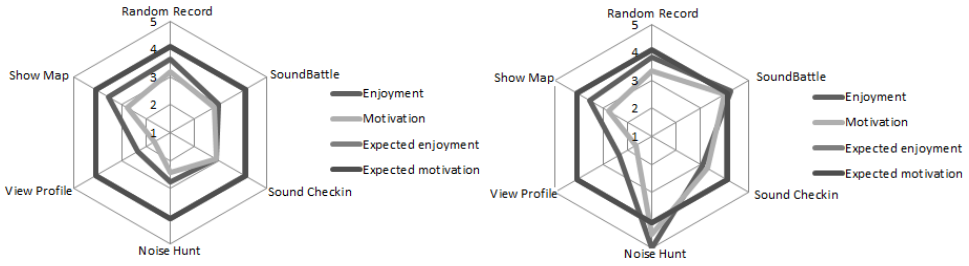


Figure 7.8: All features of NoiseMAppR compared with alike features of NoiseMAppR regarding their respective fun and challenging experiences.

The features of NoiseMAppR are compared to the similar features of NoiseMApp in figure 7.8. The comparison indicates that the fun experience is indeed bigger with the NoiseMApp application. Strangely, the opposite applied to the challenging experience. This can be due to the fact that the word 'challenging' is interpreted differently when no game elements are present.



(a) Perceived by all users

(b) Perceived by 'accurate' users

Figure 7.9: All features of NoiseMApp with their respective enjoyment and motivation score and expected scores of NoiseMAppR users.

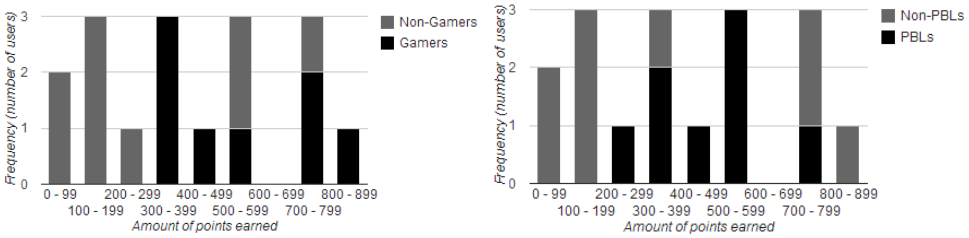
The participants were also asked about the features regarding whether they enjoyed playing it and whether it motivated the player to take more recordings. The results can be seen in figure 7.9a. In general, one could say that the NoiseMApp application was more enjoyable than it was motivating. Also users of the NoiseMAppR application were asked if adding game features (unknown to them which) in order to make recording more game-like would make the NoiseMAppR application more enjoyable or motivating. The score 4 was given for both, which is represented by the outer circle in both diagrams of figure 7.9. This shows that there might be

	Total Amount	Average	Distinct Users
Points	6887	405.1	15
Badges	28	1.6	15
Friendships	4	0.2	3

Table 7.1: Statistics concerning game elements.

a possibility to make the NoiseMApp application more enjoyable and motivating, but that the current NoiseMApp application did not provide it fully. NoiseMApp gamers did enjoy the application more than non-gamers and the same holds for the motivation. Yet, PBL users scored the opposite.

The same note as given above holds. Figure 7.9b shows the answers given by only people who at least recorded once with the respective feature. Again, a similar conclusion can be drawn from these results. It seems that people who have played the feature at least once seemed to enjoy it more.



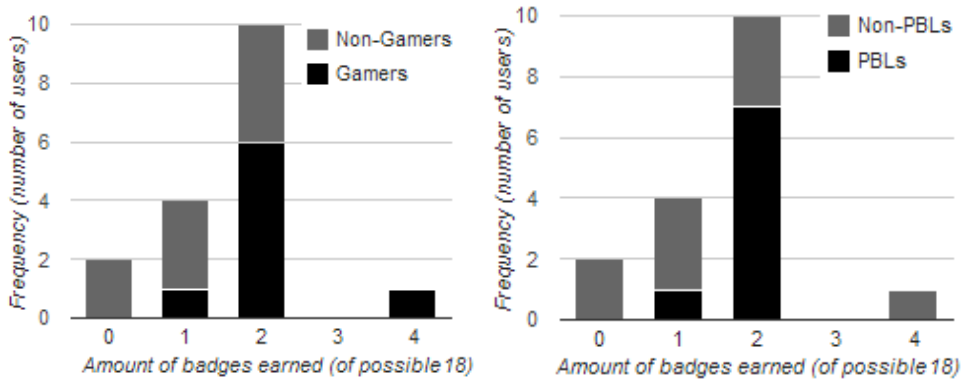
(a) By gamers vs non-gamers (b) By PBL vs non-PBL users

Figure 7.10: Amount of points won in NoiseMapp.

Game Elements In total, 6887 points were earned during the whole evaluation of NoiseMApp. The correlation with the amount of recordings is obvious, since points are given to every recording. Gamers seem to be more willing to get high scores than non-gamers, as is displayed in figure 7.10a. No significant difference is present between users that use applications with points and badges, as shown in figure 7.10b.

A total of 28 badges were earned. 15 of them are evident, since everyone who makes at least one recording, earns a badge. Again, gamers seem to earn more badges in general, yet, for users of PBL applications the opposite applies as can be seen in figures 7.11a and 7.11b.

The social aspect of the application has not been used a lot; only 4 friendships were made. It is peculiar that all these friendships were made by gamers and 2 of them were users of PBL applications.



(a) By gamers vs non-gamers (b) By PBL vs non-PBL users

Figure 7.11: Amount of badges earned in NoiseMApp.

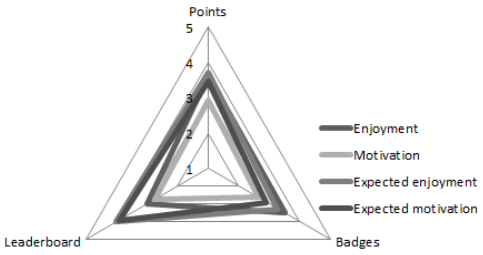


Figure 7.12: All game elements implemented in NoiseMApp were evaluated regarding their enjoyment and motivation.

The players of NoiseMApp were also asked to give a score per game element concerning what they thought was more enjoyable and motivating. NoiseMAppR users were asked whether these game elements (without knowing how they were implemented in NoiseMApp) would make NoiseMAppR more enjoyable or motivating. From figure 7.12, it can be derived that NoiseMApp users seem to enjoy earning points and badges and expectations are met. The leaderboard was not perceived as enjoyable or motivating at all, which is against the expectations of the NoiseMAppR users, which gave it the highest score. Additionally, all three elements lack motivating the users. Still, gamers of the NoiseMApp application found all three elements enjoyable and even motivating, with points on top, whereas their counterparts did not. Again, the opposite applies for users PBL applications.

Although the scores seems to indicate that points, badges and leaderboard do not influence the players of NoiseMApp to go out and record more, 88% of them would not prefer an application without these elements.

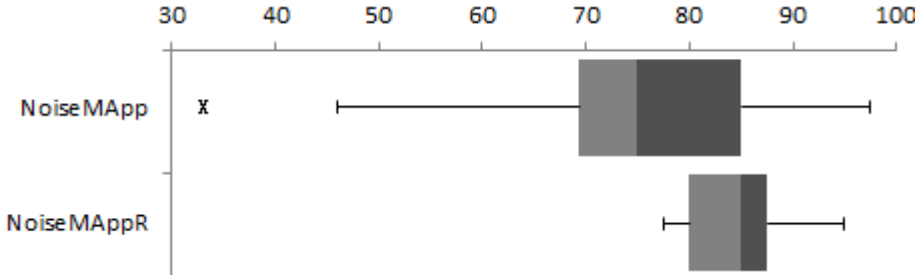


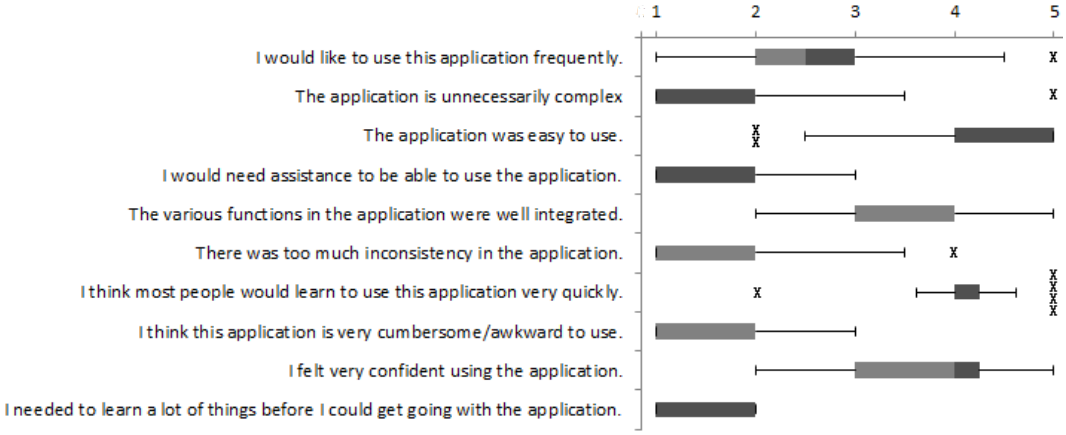
Figure 7.13: SUS-score of the NoiseMApp and NoiseMAppR application.

7.3.3 System Usability Scale

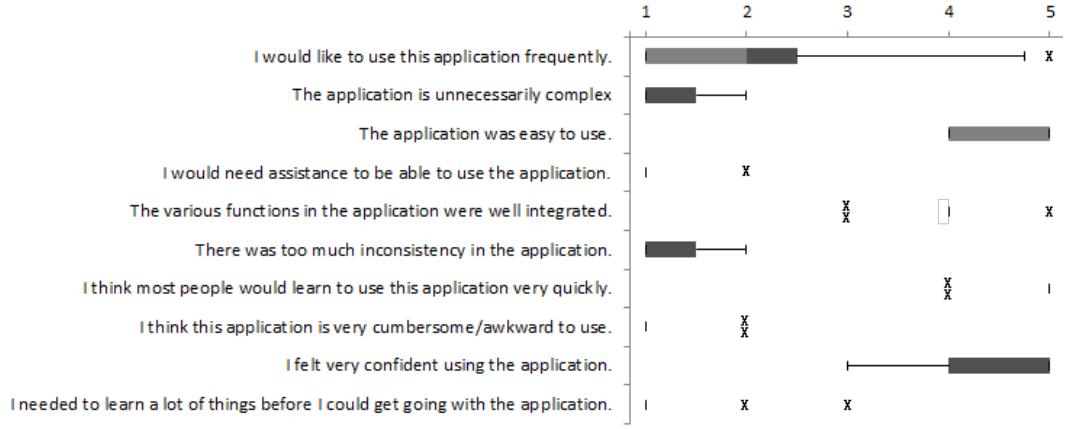
Since 16 people have filled in the questionnaire for NoiseMApp and 12 people for NoiseMAppR, the reliability of the SUS questionnaires rises up to 100% [36]. In figure 7.13 the boxplots are drawn of the total SUS-scores that were given to both applications. Although a maximum score of 97.5 was given to NoiseMApp, whereas for NoiseMAppR this is 95. Generally, NoiseMAppR scores better according to usability. The minimum score of NoiseMAppR is 75, whereas for NoiseMApp this is only 45.9, with one outlier at 32.5. The mean score for the NoiseMApp and NoiseMAppR application is 75.2 and 84.3 respectively. Since the last digital prototype, the fully working NoiseMApp application lost almost 10 points.

In figure 7.14a and 7.14b the boxplots of the different questions of both applications is given. Question 1 seems to confirm the results of previous paragraphs: the NoiseMApp application is more fun or enjoyable to use. It even seems to be more motivating if using frequently is interpreted as motivation. Referring to question 3, although most people thought NoiseMApp was easy to use, 2 people did disagree with this statement. For this statement, NoiseMAppR users all scored 4 or more. Question 5 refers to the different features possible with both applications. The resulting score does also confirm the conclusion drawn from previous paragraphs. It seems that the various features of NoiseMApp are considered too complex to be nicely usable. A reason for this, given by one user was the abundance of features of the NoiseMApp application, which made it unclear. The other questions are quite similarly scored, but the users of NoiseMAppR were more clear in their answers.

Considering [36], NoiseMAppR would be the application that would be generally preferred, if usability was the only measure of the application. Although one might think that stripping functionality of an application would have a negative effect on the usability, the opposite is true for NoiseMApp. Yet, since NoiseMAppR has less functionality, it makes sense that it is easier to use. Still, the SUS score of NoiseMApp is well above the average of 68, which ranks the application in the top 25% of the applications regarding usability [32].



(a) Likert scale scores of the NoiseMApp SUS questionnaire.



(b) Likert scale scores of the NoiseMAppR SUS questionnaire.

7.3.4 General Application Perception

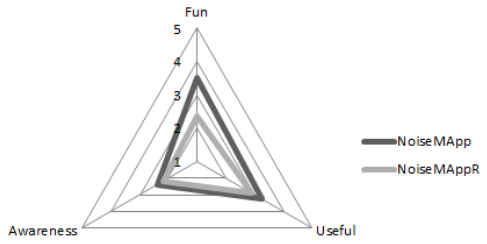


Figure 7.15: Overall perception of NoiseMApp and NoiseMAppR concerning awareness, fun and usefulness.

The test users were also asked to give a general perception of the application. They had to answer if the application made them more aware about noise pollution, was fun in general and if the application seemed useful to them. These three topics

give an indication of the answers of the the research questions, posed in section 1.2. The results of these questions are drawn on a radar plot in figure 7.15.

In general, the NoiseMApp application scored better than the NoiseMAppR application. Fun is the most convincing factor, whereas the difference regarding awareness and usefulness scored below expectations. Still, regarding the awareness, the users were already aware of noise pollution before using the application as discussed in section 7.2. As most people did not agree to to the statement that noise pollution maps would be useful, this opinion has not changed. The PBL users of NoiseMApp in general gave a higher score (round and about +1) on all three topics than their opposites. The difference between gamers and non-gamers was negligible.

Conclusion

In this chapter, the NoiseMApp application has been thoroughly evaluated. A non-gamified version, NoiseMAppR, has also been created and evaluated, so results could be compared. NoiseMApp has yielded more than double of the amount of recordings made by NoiseMAppR users. In general, the NoiseMApp application was experienced as more fun, but did not seem to motivate more. Yet, no significant difference in user engagement could be detected between both application, which might be due to the short evaluation period of 3 weeks. The NoiseMApp applications SUS score has decreased to 75.2, which still is a good usability result.

Chapter 8

Conclusion

8.1 NoiseMApp

8.1.1 Summary

NoiseMApp is an Android application created to gamify the generation of noise level maps in Leuven. Gamification is defined as the use of game elements and techniques in a non-game context [14]. By adding game elements and game features to a noise mapping process, an attempt was made to engage users to record more. The game elements used were points, badges and the leaderboard. Features included:

- Random Record: an easy to execute recording;
- Sound Checkin: similar to Random Record, but coupled to a checkin at a location;
- Sound Battle: competing against an opponent to record at three locations;
- Noise Hunt: users can complete challenges on their own.

The application was designed through an iterative development process, called rapid prototyping [24]. First, a paper prototype was created, after which two digital prototypes were developed. All prototypes were evaluated with the think-aloud method [37] and every time a SUS score [3] was obtained by applying a SUS questionnaire. Finally, a fully working application was implemented and evaluated by 17 people in Leuven during an evaluation period of 3 weeks. To compare results, a non-gamified version of the NoiseMApp application was created, called NoiseMAppR, which was evaluated by 12 people during the same evaluation period. All game elements and features of NoiseMApp were evaluated concerning activity, enjoyment, fun and motivation. Also a SUS questionnaire was performed to capture the perceived usability of the application.

8.1.2 Results

Users of NoiseMApp generated more noise recordings than the users of the NoiseMAppR application, with respectively 98 and 48 recordings. If only active users are considered, the average is respectively 7 and 4 recordings per user. This means that with gamified applications like NoiseMApp, a noise level map could be delivered with

almost twice as much data in comparison to a non-gamified version with the same amount of users. However, this holds for just a short evaluation period of 3 weeks. Most activity was measured at the beginning which was bigger for NoiseMApp, but already after 5 days after account creation it decreased to almost zero, resulting in a negligible difference of activity with NoiseMAppR. Gamers (people that play games on their smart phones once or more per week) are the ones most contributing to NoiseMApp. It seems they want to earn more points and badges than non-gamers and thus feel more motivated. They were also most social by adding friends on NoiseMApp. Gamers seem to be a good choice as target group regarding gamified applications. However, users that already use applications with points, badges and leaderboards (PBL) yield the opposite result. A possible explanation is that the motivational and fun aspects of people who often use PBL applications decrease in time, as for non-PBL users the new experience of earning points and badges in a non-game context is more motivating. The Random Record and Sound Checkin features were most popular, with 58.2% and 23.5% respectively of all recordings. Random Record was also experienced as the most fun feature, yet no challenge was encountered. This feature relates most to easy fun, which is one of the four keys of fun, according to Nicole Lazzaro [22]. Features like Sound Battle and Noise Hunt were more difficult or challenging and were less popular. Implementation constraints concerning these two scenarios may have influenced the results. For these features, users had to be in Leuven to complete the game. Only 56% said to be working or living in Leuven, which has an impact on other users during a Sound Battle. Because of this, people did not finish their games, which led to more demotivational experiences. NoiseMAppR users also thought that adding game elements like points, badges and a leaderboard and game techniques to make NoiseMAppR more game-like should increase the motivation and the enjoyment of the application significantly. Yet, NoiseMApp users did not feel like these expectations were met. The SUS score gives an indication of the usability of the application. NoiseMApp scored on average 75.2, which is less than the previous digital prototype. It seems the interaction with real users did not feel as comfortable as with the prototype. NoiseMAppR scored on average 84.3 and consequently, according to [36], NoiseMAppR would most likely be preferred by the whole population. Yet, NoiseMApp users indicated they did not prefer to having no points or badges to earn.

As main conclusion it might have been too ambitious to implement this many features into one application. People like to keep it simple, as has been shown by the popularity of Random Record, which generated most noise level data. A reason could be that the benefit of users did not compare to the cost of playing challenging games. Adding only game elements to a non-game context might already yield a significantly better result. Gamification in general does seem to have a positive effect in the generation of noise level maps. However, simplicity and single-featured applications (i.e. only Random Record) might achieve better results.

8.2 Future Work

The evaluation of the application has showed that NoiseMApp yielded a higher activity rate than NoiseMAppR. Yet, long term engagement could not be thoroughly evaluated, due to the short evaluation period. It is yet feasible to evaluate the user engagement on a longer period.
To broaden the applicability of the application, recording of specific sounds (car horns, barking dogs, construction works, ...) could be made possible. There already are studies [25] of mobile applications that try to capture as many sounds as they can to classify them in order to recognize such sounds in the future.
Features of the NoiseMApp could be improved. The profile, for instance, only shows the last Sound Checkin location. Friends can see this location, but do not get an update whenever it is changed. Adding a news feed to the application might raise engagement when people directly see what other people have been up to. Also the Sound Battle could be improved. Now, a lot of people do not finish their Sound Battle. This is frustrating for people who like to play. A forfeit score could be given to users that are inactive during a couple of days. The implementations of Sound Battle and Noise Hunt could also be extended in order to be usable outside of Leuven. Users could be asked for their home locations, so Sound Battle and Noise Hunts will always reside in their neighbourhood. The Show Map feature could also be enhanced. Now, only dots are placed on a map with the according decibel level. More information about the time of the recording could be included, perhaps with filters for specific time intervals or a specific day of the week. The recording itself can also be improved. Different smart phones (especially on the Android platform) have different types of microphones. These microphones have different parameters that influence the noise level recorded by that microphone. Although this has not been discussed in this thesis, it is critical to study this in order to generate accurate noise level maps. Further, social media integration might enhance the activity of users when they see friends using the application.
Since points are one of the main elements that motivate users, it might make sense to couple the points earned to an online BitCoin[1] community on which users can use their BitCoins to buy products.
Finally, since easy fun is most liked by the users, one might think about crowdsensing applications. These are applications on smart phones where (almost) no human interactivity is involved. The device automatically sends measurements to a server on regular intervals. It might be interesting to add points to this application whenever the application is active. Still, for noise measurements it is necessary to hold the device outside (not in a pocket) to have a more accurate measurement.

[1] BitCoin is a platform on which electronic money can be earned. Developers can integrate this cash system through Open Source software. (http://www.bitcoin.org)

Appendices

Appendix A

Scientific Paper

In this appendix, the scientific paper of his master thesis is included. The paper can also be found in colour on Github: https://github.com/Phille88/NoiseMApp/tree/master/Paper.

NoiseMApp: Genereren van Geluidsniveaukaarten in Leuven met behulp van Gamificatie

Philippe De Croock
HCI Departement
KU Leuven
philippe.decroock@student.kuleuven.be

Samenvatting

Geluidsniveakaarten genereren zijn een dure en arbeidsintensieve bezigheid. Vandaag bestaan er al applicaties op smart phones, waarmee iedereen kan worden ingezet om geluidsniveaus te gaan opmeten met de microfoon van hun smart phone. Dit is echter een onbaatzuchtige voor de gebruikers, waardoor het engagement snel afneemt. In deze paper wordt een manier voorgesteld om zulke applicaties leuker te maken door gebruik te maken van gamificatie. Rapid Prototyping, een iteratief ontwikkelproces, werd gebruikt om 1 papieren en 2 digitale prototypes te ontwikkelen, die telkens werden geëvalueerd op gebruiksvriendelijkheid en bruikbaarheid. De finale applicatie, genaamd NoiseMApp, implementeert spelelementen zoals punten, badges en ranglijsten en speltechnieken waaronder Random Record, Sound Battle, Noise Hunt en Sound Checkin. NoiseMApp is uitvoerig geëvalueerd gedurende een evaluatieperiode van 3 weken en er werden meer opnames gemaakt in vergelijking met een niet-gegamifieerde versie.

1. PROBLEEMSTELLING

Sinds het begin van het academiejaar 2012-2013 worden in Leuven maatregelen genomen tegen geluidsoverlast van studenten [9]. *Fakbars*[1] krijgen een sluitingsuur, studentenkaarten zijn verplicht bij het binnenkomen en verhoogde politiecontroles proberen ervoor te zorgen dat het geluidsniveau 's avonds daalt. Deze maatregelen vereisen extra tijd en geld van de politiediensten, alsook van het fakbarpersoneel. Studenten zijn vaak 's avonds luidruchtig op straat na feestjes en ook het decibelniveau van muziek op feestjes zorgen voor geluidsoverlast. Bewoners rond de Tiensestraat en de Oude Markt kunnen niet meer van hun rust genieten na 22u op uitgaansavonden [7].

Om het probleem van geluidsoverlast in kaart te brengen, kan men geluidsniveaukaarten ontwikkelen. Dit is een stratenplan van de stad waarop kleuren zijn aangebracht die een indicatie geven over het geluidsniveau op die locatie. Groen stelt een laag geluidsniveau voor, terwijl rood een hoog geluidsniveau aangeeft. Dergelijke kaarten bestaan reeds in Antwerpen, Gent en Brugge [2] om conform te zijn met de milieuwetgeving, opgenomen in de VLAREM[2] [6]. Geluidsniveaukaarten ontwikkelen is echter arbeidsintensief en vereist een groot budget vanwege de nodige apparatuur en arbeidsuren. Dit maakt het zo goed als onmogelijk om dit werk regelmatig uit te voeren met up-to-date data.

In Leuven bestaat er nog geen geluidsniveaukaart. Jong CD&V Leuven trok daarom reeds de stad in met decibelmeterapplicaties op hun smart phone om op bepaalde plaatsen in Leuven het geluidsniveau te meten. [11].

Veel mensen zijn tegenwoordig in het bezit van een smart phone. Deze smartphones zijn uitgerust met microfoons, gps, accelerometers, wifi en allerhande sensoren. In dit paper wordt een applicatie voorgesteld waarbij mensen meer worden geëngageerd om geluid te gaan opnemen in Leuven via hun smart phone. Dit gebeurt door het gebruik van gamificatie.

2. DEFINITIE

De meest gangbare definitie van gamificatie is het gebruik van spel-mechanismen en spelontwerptechnieken in contexten die niets met een spel te maken hebben. Deze definitie wordt zowel gebruikt door S. Deterding [3] als door Kevin Werbach [15]. De definitie is drieledig.

Met spel-mechanismen worden onder meer spelelementen bedoeld die vaak voorkomen in spelletjes.

[1] Café's in Leuven voor studenten, uitgebaat door studenten.

[2] Vlaams reglement betreffende de milieuvergunning

Wanneer men denkt aan voorbeelden zoals de sociale spelletjes FarmVille[3] en Foursquare[4], komen meteen badges, leaderboards en punten (ofwel PBL's), avatars, virtuele goederen, beloningen, ... ter sprake. Het *gebruik* van deze elementen is echter niet voldoende om van een applicatie een spel te maken.

Het tweede deel van de definitie legt hier de nadruk op. Er moet worden nagedacht zoals een spelontwikkelaar om van iets een spel te maken. Hier komen spel-ontwerptechnieken bij kijken. Zij zorgen voor de cohesie van spelelementen zodat de spelelementen nut hebben. Dit heeft als resultaat dat spelers de applicatie inderdaad als een spel ervaren.

Het derde deel is een onmisbaar deel van de definitie. Zonder het derde deel van de definitie betreft de definitie immers alle spelletjes. De gegamifieerde applicatie heeft echter betrekking op een context die op zich niets met een spel te maken heeft. Een voorbeeld van een non-spel context is een persoon die gaat joggen, werken, naar de winkel gaat, enzovoort. Vaak wordt gamificatie vergeleken met serious games om dit verschil duidelijk te maken. Serious games zijn volledig ontwikkelde spellen die met een doel gemaakt zijn, zoals bijvoorbeeld het duidelijkmaken van hoe belangrijk een levensverzekering is[5].

Een applicatie die spelelementen en spelontwerptechnieken gebruikt om een taak voor die persoon leuker en gemakkelijker vol te houden te maken, is een gegamifieerde applicatie. DueProps [4], bijvoorbeeld, is een applicatie die collega's aanzet om complimenten te geven over taken door 'props' aan hen uit te delen.

3. VERGELIJKBAAR WERK

Er bestaan reeds tal van applicaties waar geluid kan worden opgenomen met locatiegegevens door een smartphone.

Een voorbeeld van zo'n applicatie is NoiseWatch [5]. Hier kan je erop uit trekken en geluid opnemen. De gegevens van het geluid worden verwerkt en de gegevens worden bijgehouden. Deze kunnen dan later worden bekeken op de website `http://www.eyeonearth.org`.

Een applicatie die exact hetzelfde doet is NoiseTube Mobile [14]. NoiseTube maakt het ook mogelijk voor andere applicaties om gebruik te maken van hun API. Gegevens kunnen dan via de noisetube.net server worden opgeslagen.

NoiseDroid [8] is een applicatie die ook rekening houdt met de kwaliteit van de opname. Wanneer de gebruiker bijvoorbeeld beweegt tijdens de opname, zal de kwaliteit verminderen. De API van NoiseDroid is echter niet meer onderhouden en maakt het lastiger om te gebruiken.

Verder werd er reeds een paper geschreven over het idee van een gegamifieerde geluidsoverlastapplicatie researchers aan de Jaume I universiteit [10]. Zij stelden reeds prototypes voor zoals NoiseBattle en NoiseQuest. In NoiseBattle wordt het te mappen gebied in vakjes ingedeeld. Elke speler moet dan over een zo groot mogelijk gebied heersen door voldoende en op regelmatige tijdstippen geluid te gaan opnemen. Vakjes kunnen echter worden overmeesterd door tegenstanders, wanneer zij recentere opnames maken. NoiseQuest is een missiespel dat elke gebruiker op zich kan spelen en volgt een bepaalde verhaallijn.

4. DOELSTELLING

De applicatie die in deze paper wordt voorgesteld gebruikt gamificatie om zo veel mogelijk mensen aan te zetten om geluid te gaan opnemen. Op die manier worden er veel geluidsgegevens gegenereerd zowel op verschillende locaties als op verschillende tijdstippen. Om dit te verwezenlijken, worden spelelementen en -technieken toegevoegd.

4.1. Spelelementen

Spelelementen worden toegevoegd om de gebruiker feedback te geven over hoe goed hij bezig is. Dit werkt zowel engagerend als informatief. Er is gekozen om met de welgekende punten, badges en leaderboards (PBLs) te werken, vanwege hun alomtegenwoordigheid in allerhande spelen.

4.1.1. Punten. Bij het opnemen van geluiden op locatie kunnen punten worden verdiend. Deze punten hangen af van een aantal factoren: geluidskwaliteit, eerste keer op locatie, eerste keer van vrienden op die locatie, enz. De punten op zich hebben geen nut. Het is eerder bedoeld als feedback naar de gebruiker toe. De spelers kunnen zich dan onderling meten aan de hand van een maatstaf. In de Sound Battle bijvoorbeeld (zie sectie 4.2.2), worden de punten van twee spelers vergeleken om de winnaar te bepalen.

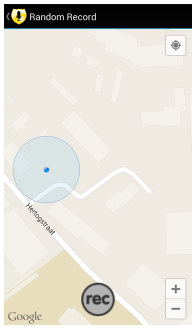
[3] `http://www.farmville.com`
[4] `http://www.foursquare.com`
[5] `http://www.axa-equitablepassiton.com`

4.1.2. Badges. Badges worden verdiend wanneer vooraf opgestelde doelstellingen bereikt worden. Deze doelstellingen zijn echter niet gekend voor de speler, waardoor een verrassingseffect verkregen wordt. Een voorbeeld is dat wanneer spelers voor de eerste keer een geluid opnemen, zij de *Newbie* badge ontvangen. Deze Newbie badge krijgt men automatisch en komt voortaan te voorschijn op het profiel van de speler.
De badges, die ook met bonuspunten komen, hebben voornamelijk het doel om de speler beloningen te geven. Ook kan de gebruiker met de badges prijken op zijn profielpagina die door andere spelers kan worden bekeken. Een mogelijkheid om de verdiende badge te delen op Facebook is hier een gemakkelijke uitbreiding van. Dit werd echter niet geïmplementeerd.

4.1.3. Leaderboards. Om spelers nog meer met elkaar te laten competitie voeren, kan ook een leaderboard worden geraadpleegd. Dit leaderboard maakt het mogelijk dat de speler zichzelf weet te plaatsen tussen zijn vrienden.

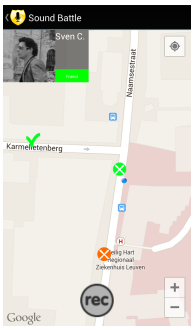
4.2. Speltechnieken

De applicatie maakt het mogelijk om vier verschillende spelletjes te spelen: Random Record, Sound Battle, Noise Hunt en Sound Checkin. Wanneer de applicatie start krijgt de speler de keuze een van deze modi te kiezen. De scenarios van deze modi worden hieronder verder toegelicht.



Figuur 1: Een Random Record spel in NoiseMApp.

4.2.1. Random Record. Random Record maakt het mogelijk dat de spelers op elk moment erop uit kunnen trekken om geluid te gaan opnemen. Het scenario wordt geselecteerd door op de desbetreffende knop te drukken. De locatie van speler wordt dan via GPS getracht vast te leggen. De speler krijgt een scherm te zien met een kaart van Google Maps en een opneemknop 1. Pas wanneer er een voldoende sterk GPS signaal gevonden is, zal de gebruiker geluid kunnen opnemen. Hierdoor wordt de gebruiker verplicht om buitenshuis op te nemen. Zo'n opname duurt 10 seconden. Na de opname krijgen de spelers een overzicht van de punten en de eventuele badges die ze met die opname hebben verdiend. Uiteraard kan dit zo veel keer als men wil.
Het doel van Random Record is om gebruikers de vrije wil te laten om op te nemen waar ze maar willen.



Figuur 2: Een Sound Battle spel in NoiseMApp.

4.2.2. Sound Battle. Men kan er ook voor kiezen om te spelen tegen iemand anders. Dit scenario is te vergelijken met het NoiseBattle prototype, vermeld in sectie 3, maar mag er niet mee worden verward. Bij het drukken op de Sound Battle knop, kunnen spelers kiezen om een nieuw spel te maken waarbij een tegenstander random wordt gekozen. Als er reeds spelletjes bezig zijn, maar nog niet afgewerkt, kunnen deze spellen ook worden gekozen zodat ze verder kunnen worden afgemaakt. In een Sound Battle zijn er drie random gegenereerde locaties op hoogstens 200m afstand op de map aangeduid. Het is de bedoeling dat beide spelers op de drie aangeduide locaties geluid opnemen. Deze locatie ligt altijd op een weg die voor het publiek toegankelijk is. De afstand tot de locatie wordt weergegeven met een kleurcode die gaat van rood (ver weg) tot groen (dicht genoeg om op te nemen). Hoe sneller, hoe dichter bij de opgegeven locatie en hoe beter de geluidskwaliteit, hoe beter spelers scoren in de Sound Battle. Het opnemen door beide spelers gebeurt volledig onafhankelijk. Nadat het geluid op de drie locaties is opgenomen door beide partijen, verschijnt er een overzicht van wie er heeft gewonnen. Aan de hand van het aantal gewonnen Sound Battles, vallen er badges met bonuspunten te verdienen.
Doordat in een Sound Battle drie geluidsopnames dicht bij elkaar moeten worden gemaakt, kan er op korte tijd een grote hoeveelheid aan geluidsniveaudata worden verkregen. De Sound Battle kan enkel in Leuven worden gespeeld. Tijdens het spel is het mogelijk om de tegenstander toe te voegen als vriend.

4.2.3. Noise Hunt. Spelers kunnen er ook voor kiezen om tegen zichzelf te spelen. Noise Hunts zijn verschillende opdrachten die de speler één voor één moet uitvoeren. Vier Noise Hunts werden geïmplementeerd: Walk in the Park, Blitkrieg, Party Time en Riverside. In Walk in the Park moeten 3 geluidsopnames gemaakt worden in het stadspark van Leuven op 50 meter afstand van elkaar. Als die Noise Hunt correct wordt uitgevoerd, kan men de Blitzkrieg spelen. Daar moeten spelers in een tijd van 2 minuten zoveel mogelijk opnames maken. Party Time is gelijkaardig aan de eerste, maar nu moeten de opnames in het centrum aan de Oude Markt worden gemaakt tussen 10 uur 's avonds en 6 uur 's morgens. Met de laatste Noise Hunt moeten er opnames worden gemaakt vlakbij het water. Ook hier zijn badges te verdienen bij elke Noise Hunt.
De Noise Hunt maakt het mogelijk om spelers te richten naar bepaalde locaties waar meer geluidsinformatie nodig is. Dit scenario komt qua motivatie overeen met het eerder vermeldde NoiseQuest in sectie 3. Noise Hunts vinden plaats in Leuven.

4.2.4. Sound Checkin. Vaker zijn er studenten op feestjes met oordopjes. Hoewel er al maatregelen worden genomen tegen geluidsniveaus op de dansvloer, blijkt dit niet voldoende. Sound Checkin maakt het mogelijk dat men op plaatsen zoals café's, discotheken, musea, enz. kan inchecken, zoals bij Foursquare. Men moet er echter wel het geluid voor opnemen op die locatie. Er is echter geen GPS verbinding voor nodig. De plaatsen om in te checken worden geleverd door Google Places API.
Sound Checkin dient ervoor om informatie te geven aan vrienden over het geluidsniveau op die locatie. Op deze manier heeft NoiseMApp ook een sociaal karakter. Zo kan men weten of men beter oordopjes meeneemt naar een feestje of niet.

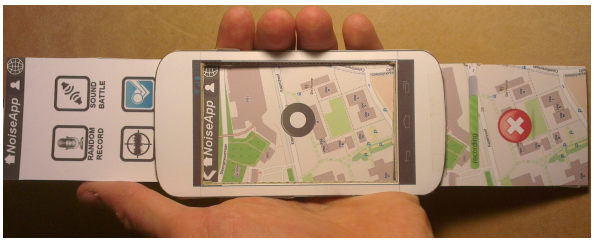
4.2.5. View Profile. Met View Profile kunnen spelers hun eigen profiel bekijken. Hierop is te lezen hoeveel punten ze hebben verdiend, hoeveel Sound Battles er gewonnen zijn, waar er laatst is ingechecked en alle badges die zijn verdiend, worden opgesomd. Ook is het mogelijk om de ranglijst te bekijken van vrienden om te zien hoe goed men bezig is, wat de competitie kan opdrijven. Via deze weg is het ook mogelijk hun profiel te bekijken.

4.2.6. Show Map. Het doel van het Show Map scenario is dat mensen geïnformeerd blijven over wat ze juist doen. Op een kaart zijn alle opgenomen geluidsniveaus te zien met een kleurcode gelijkaardig aan die van officiële geluidsniveaukaarten.

5. ONTWERPPROCES

Het ontwerpproces volgde de methodiek van Rapid Prototyping. Dat is een techniek die gebruikt wordt om iteratief applicaties te ontwerpen. Eerst werd een papieren prototype gemaakt. Daarna werd er een partiële en een volledig digitaal prototype ontwikkeld.

5.1. Papieren Prototype



Figuur 3: Werking van het papieren prototype

Eerst werd er een papieren prototype gemaakt. Dit bestond uit een kartonnen hoes die dienst deed als smart phone. Door de hoes kunnen dan de schermen worden geschoven (zie figuur 3). Testgebruikers moesten dan doen alsof de applicatie echt werkte. Wanneer gebruikers op een papieren knop drukken, wordt het volgende scherm door de hoes geschoven, wat interactie met de gebruiker mogelijk maakt. Acht personen werden getest. Aan de hand van de Think-Aloud methode [13], waarbij de testpersonen luidop moeten zeggen wat ze denken, is er een eerste evaluatie gebeurd omtrent de functionaliteit en de gebruiksvriendelijkheid. Voor dat laatste werd de System Usability Scale (SUS) gebruikt [1]. Dat is een enquête waarbij 10 uitspraken punten moeten worden gegeven van 1 (volledig oneens) tot 5 (volledig eens), ook wel de Likert schaal genoemd. Uit de resultaten kan dan een score op 100 worden berekend. Het papieren prototype scoorde al meteen 81, wat een stuk hoger ligt dan de gemiddelde score van 68 [12].

5.2. Digitale Prototypes

Aan de hand van de goede resultaten van het papieren prototype, werd er een eerste digitale prototype gemaakt. Hoewel het prototype nu op een smart phone werkte, was er nog steeds geen echte functionaliteit aanwezig. Aanvankelijk werden enkel het Random Record en het Sound Battle scenario geïmplementeerd. Deze scenarios leken het belangrijkste voor de applicatie. Ze werden meteen getest door 5 personen op dezelfde wijze als het papieren prototype. Achteraf kregen ze een enquête met vragen over de scenarios en

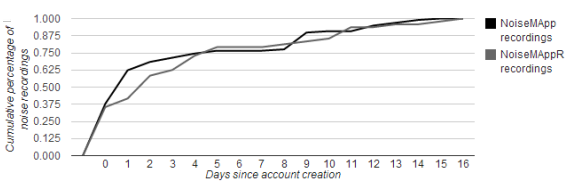
een SUS enquête. Het prototype scoorde dezelfde SUS score en er waren kleine opmerkingen over het design. Het laatste digitale prototype werd dan ontwikkeld waarmee alle scenarios konden worden getest. Nog steeds was er geen echte functionaliteit. Op dezelfde manier werd de applicatie geëvalueerd met 9 personen. Deze keer was de SUS score zelfs gestegen tot 86.

6. NOISEMAPP

Figuur 4: The badges to earn in Noise Hunts.

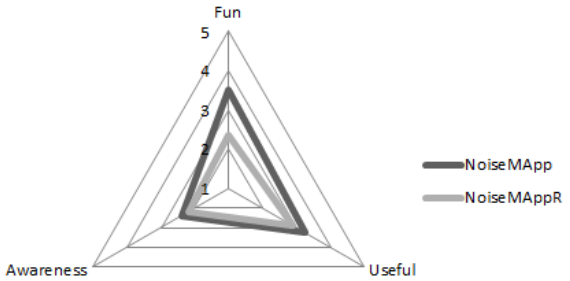
Na de positieve evaluaties van de prototypes is de implementatie van de functionaliteit van de NoiseMApp applicatie begonnen. Android, het populairste mobiele besturingssysteem, werd gekozen als platform om de applicatie te implementeren. Dit werd reeds beslist bij het implementeren van het eerste digitale prototype. Enkele aanpassingen waren nodig om de applicatie werkelijk te doen werken. Zo werden echte punten en badges nu toegekend aan opnames. De badges die bij te verdienen zijn met de Noise Hunt worden getoond in figuur 4. Om de applicatie volledig te doen werken, werd een back-end ontwikkeld. Een MySQL database werd aangelegd waar de applicatie opnamedata en andere gebruikersgegevens in bewaarde. De communicatie met de database gebeurde via PHP scripts die werden uitgevoerd, data werd uitgewisseld in JSON formaat.

7. EVALUATIE



Figuur 5: Cumulatieve opnameactiviteit in functie van actieve dagen.

Om NoiseMApp te evalueren, werd een niet-gegamifieerde applicatie ontwikkeld waarmee kon worden vergeleken, genaamd NoisMAppR. Met NoiseMAppR kon het Random Record scenario worden uitgevoerd, maar geen punten konden worden verdiend.



Figuur 6: Vergelijking van NoiseMApp met NoiseMAppR op basis van sensibilisering, plezier en bruikbaarheid.

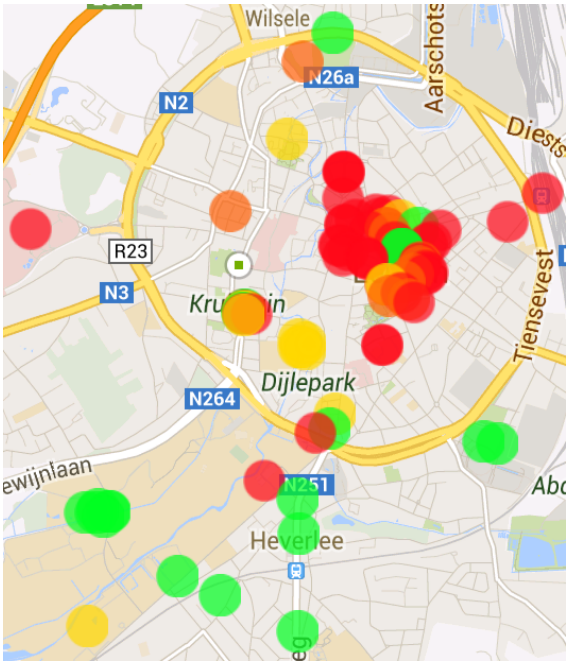
Ook kon de map getoond worden met alle opgenomen geluidsniveaus. Gedurende een periode van 3 weken werd NoiseMApp gespeeld door 17 mensen en NoiseMAppR door 12. Zij genereerden respectievelijk 98 en 48 geluidsopnames. Dit betekent dat gemiddeld 5.7 en 4 opnames per persoon werden gemaakt. Echter, 2 personen van de NoiseMApp groep waren volledig inactief en indien verwaarloosd, stijgt het gemiddelde tot 7 opnames per gebruiker. In figuur 5 is de activiteit van NoiseMApp en NoiseMAppR in functie van het aantal dagen na account creatie uitgezet. In het begin is de activiteit van NoiseMApp sterker dan die van NoiseMAppR, maar al na reeds 5 dagen wordt het verschil verwaarloosbaar. Bij testpersonen die wekelijks een spel spelen op hun smart phone is de activiteit duidelijk groter dan bij niet-gamers.
In totaal zijn er 28 badges verdiend, waarvan 15 evident, vermits ze gegeven werden bij de allereerste recording. De meesten hebben 2 badges verdiend, maar 1 gebruiker heeft 4 badges verdiend en deze was een gamer.
Van alle features in NoiseMAppR, is Random Record het vaakst gebruikt (58.2%), Sound Checkin voor 23.5% van de recordings en het minst werd Sound Battle gebruikt met 8.2%. Wederom waren het vooral gamers die de Sound Battle wel gebruikten.
Alle deelnemers werden gevraagd om hun applicatie een score te geven van 1 (volledig niet) tot 5 (volledig wel) op volgende vlakken: sensibilisering, plezier en bruikbaarheid. Het resultaat staat weergegeven op het diagram in figuur 6. Hoewel NoiseMApp gebruikers meer plezier ondervonden, is NoiseMApp er niet in geslaagd om een hogere score te behalen op de andere domeinen.

8. TOEKOMSTIG WERK

Met de evaluatie van NoiseMApp kon geen langdurig engagement van de gebruikers worden aangetoond. Een langere evaluate zou nog een belangrijk licht kunnen werpen op de zaak. De applicatie kan ook met an-

dere doeleinden worden gecombineerd. Zo kan er getracht worden om gebruikers specifieke geluiden (zoals autotoeters, bouwwerken, enz.) te laten opnemen, die kunnen worden geanalyseerd en geclassificeerd.
Ook kan er gedacht worden aan crowdsensing, waarbij geen interactie met de gebruiker nodig is, om geluidsniveaukaarten te maken. Dit kan dan eventueel gegamifieerd worden door punten te koppelen aan het aantal uren dat de applicatie actief is op de smart phone. Hier stelt echter wel het probleem dat de geluidskwaliteit van de opname veel minder is, doordat de gebruiker niet aangespoord wordt om stil te blijven staan.

9. BESLUIT



Figuur 7: The noise level map generated by NoiseMApp users.

Gamificatie heeft nog heel wat onderzoek nodig. Er bestaat nog geen goede methodologie om applicaties te gamifieren. Met NoiseMApp is getracht om het probleem van geluidsniveaukaarten goedkoop te genereren. Het resultaat hiervan is te zien in figuur 7. Hoewel er meer opnames zijn gemaakt met NoiseMApp dan met een niet-gegamifieerde variant, is de gebruikersactiviteit niet noemenswaardig toegenomen. Gamers lijken wel de ideale doelgroep te zijn voor zulke applicaties. Vermits Random Record het populairst was, kan men stellen dat de gamificatie vooral effect heeft op simpelere handelingen. Sound Battle en Noise Hunt waren leken eerder iets te complex. Ook is er geen directe baat voor de gebruiker om geluidsopnames te maken, behalve het plezier element, dat toch beperkt blijft. Het lijkt daarom aangeraden om voor het genereren van geluidsniveaukaarten de applicatie simpel te houden.

Referenties

[1] J. Brooke. SUS-A quick and dirty usability scale. *Usability evaluation in industry*, pages 1–8, 1996.

[2] N. e. E. Departement van Leefmilieu. Goedgekeurde geluidskaarten. `http://www.lne.be/themas/hinder-en-risicos/geluidshinder/beleid/eu-richtlijn/goedgekeurde-geluidskaarten`.

[3] S. Deterding, D. Dixon, R. Khaled, and L. Nacke. From Game Design Elements to Gamefulness : Defining Gamification. In *Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments*, volume Tampere, F of *MindTrek '11*, pages 9–15, New York, NY, USA, 2011. Acm.

[4] DueProps. Peer Recognition Game for Worplaces. `https://dueprops.com`.

[5] European Environment Agency. NoiseWatch. `http://watch.eyeonearth.org`.

[6] Het Europees Parlement en de Raad van de Europese Unie. RICHTLIJN 2002/49/EG VAN HET EUROPEES PARLEMENT EN DE RAAD van 25 juni 2002 inzake de evaluatie en de beheersing van omgevingslawaai. *Publicatieblad van de Europese Gemeenschappen*, L 189:12–25, 2002.

[7] KU Leuven. Geluidsoverlast in Tiensestraat: stewards brengen rust, Dec. 2000.

[8] Lfgi. NoiseDroid. `https://play.google.com/store/apps/details?id=de.noisedroid\&hl=nl`.

[9] Lokale Politie Leuven. Projectmatige aanpak uitgaansoverlast. `http://www.lokalepolitie.be/5388/uitgaan/125-project-aanpak-uitgaansoverlast/320-projectmatige-aanpak-geluidsoverlast.html`.

[10] I. G. Martí, L. E. Rodríguez, M. Benedito, S. Trilles, A. Beltrán, L. Díaz, and J. Huerta. Mobile application for noise pollution monitoring through gamification techniques. In *Proceedings of the 11th international conference on Entertainment Computing*, ICEC'12, pages 562–571, Berlin, Heidelberg, 2012. Springer-Verlag.

[11] P. V. Rompuy. JONGCD&V Leuven pakt geluidshinder aan. `http://www.politico-shock.be/peter/2012/jongcdv-leuven-wil-een-geluidskaart.html`.

[12] J. Sauro. MeasuringUsability.

[13] M. W. van Someren, Y. F. Barnard, and J. A. C. Sandberg. *The Think Aloud Method: A Practical Guide to Modelling Cognitive Processes*. Academic Press Limited, London, 1994.

[14] VUB BrusSense Group. NoiseTube. http://www.noisetube.net.

[15] K. Werbach. Coursera: Gamification Course. https://www.coursera.org/course/gamification.

Appendix B

Poster

In this appendix, the poster of this master thesis is included. A version in A3 format can be found on Github: https://github.com/Phille88/NoiseMApp/tree/master/Poster.

KU LEUVEN
Master Thesis 2012-2013
Prof. Dr. Erik Duval
Gonzalo Parra
José Luis Santos
Philippe De Croock Master of Science in Engineering: Computer Science, *Distributed Systems*

NoiseMApp

Generating Noise Pollution Maps through Gamification in Leuven

Noise in Leuven

Hot topic in Leuven

- *Fakbars'* curfew at 5 o'clock
- Extra stewards & police

Noise pollution has many **disadvantages:**

- Stress
- Aggression
- Sleep disorder

Noise pollution is still an **underestimated problem** and creating **awareness** is a good start to tackle this issue.

Noise Pollution Maps

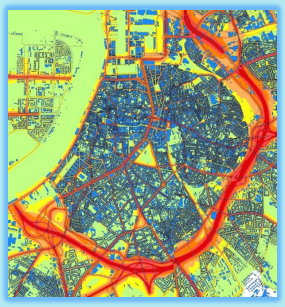
Static maps with **color codes** representing **noise level**

Already made for:

- Antwerp
- Bruges
- Ghent

"Why not **Leuven?**" (Jong CD&V Leuven)

Why not generate these maps **cheaply** and **dynamically?**



Gamification

"the use of game elements and techniques in a non-game context"

Game elements

Points
for every recording made
Badges
received occasionally when reaching different goals
Leaderboard
compare yourself to your friends
Profile
check your status and brag about it

Game techniques

Random Record
go outside and record wherever you want
Sound Battle
compete with an opponent to make the best recordings
Noise Hunt
challenge yourself in completing specific tasks
Sound Checkin
check in to bars and discotheques with the sound level

Non-game context

*Players have to **record noise** during 10 seconds in order to **generate data** that makes it possible to automatically create a **noise level map** of Leuven. With **NoiseMApp**, a **dull task** like the recording of noise, becomes **fun!***

http://philippedecroock.wordpress.com

Appendix C

Screen flows

All screenshots of the digital prototypes and the final application can be found here. They can also be viewed on Github: https://github.com/Phille88/NoiseMApp/tree/master/Images/Screenshots.

C.1 First Digital Prototype

All screens of a the scenario flows of the first digital prototype are shown.

C.1.1 Main screen

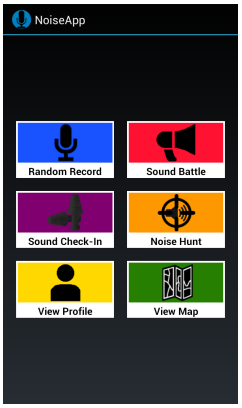


Figure C.1: The main screen of the first NoiseMApp prototype.

C.1.2 Random Record

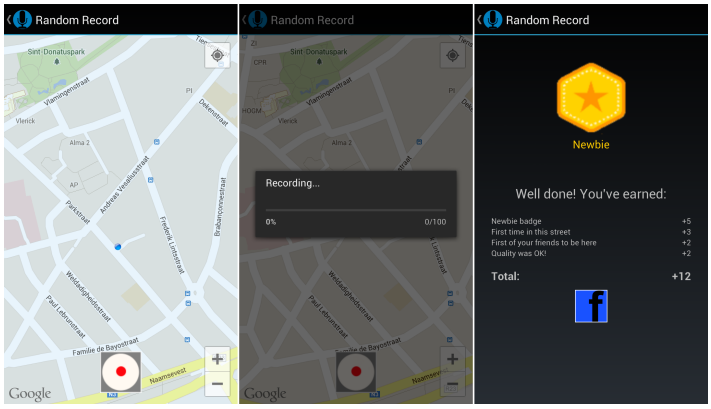


Figure C.2: The Random Record scenario of the first NoiseMApp prototype.

C.1.3 Sound Battle

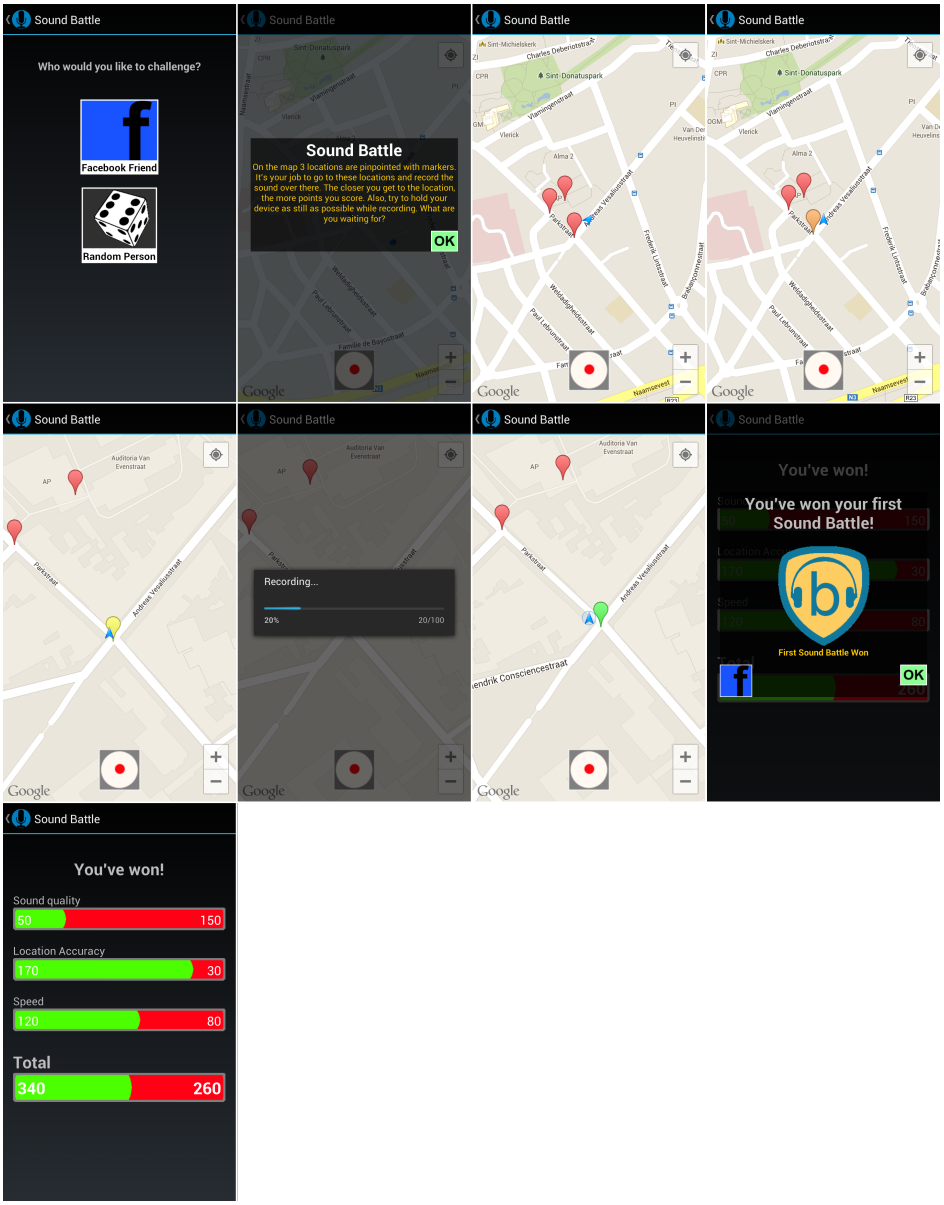


Figure C.3: The Sound Battle scenario of the first NoiseMApp prototype.

C.2 Full Digital Prototype

All screens of a the scenario flows of the full digital prototype are shown.

C.2.1 Main screen

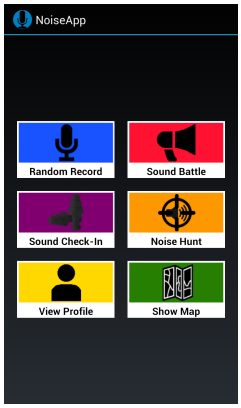


Figure C.4: The main screen of the full digital NoiseMApp prototype.

C.2.2 Random Record

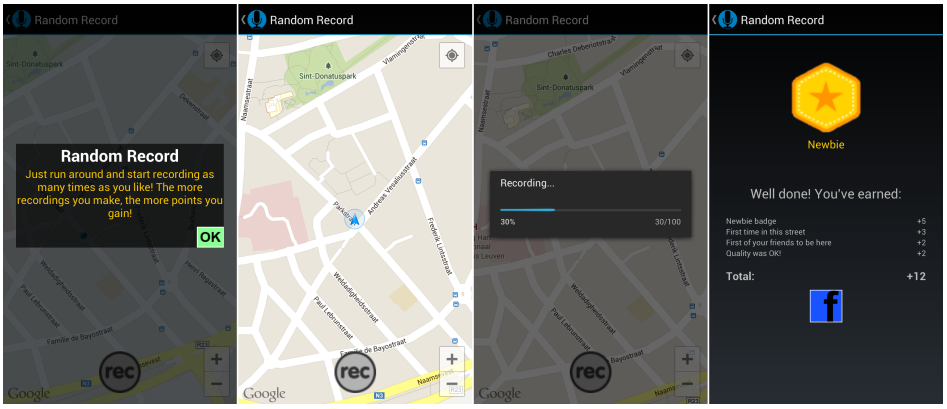


Figure C.5: The Random Record scenario of the full digital NoiseMApp prototype.

C.2.3 Sound Battle

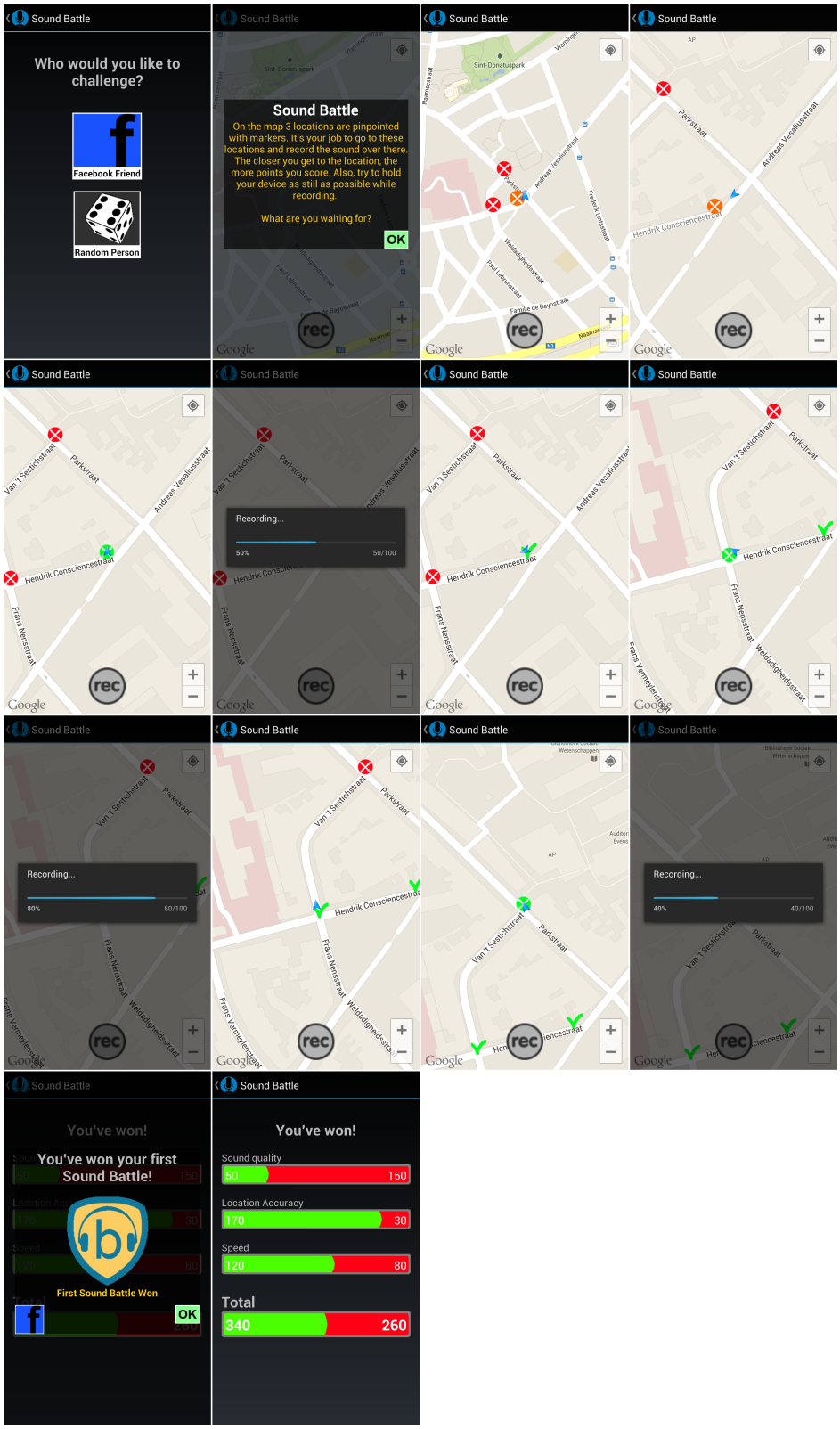


Figure C.6: The Sound Battle scenario of the full digital NoiseMApp prototype.

C.2.4 Sound Checkin

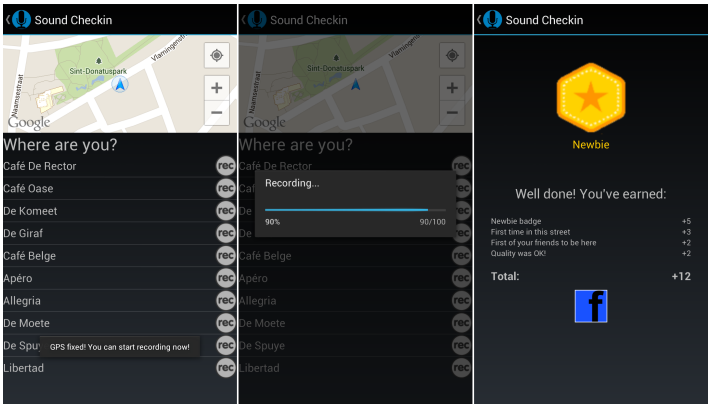


Figure C.7: The Sound Checkin scenario of the full digital NoiseMApp prototype.

C.2.5 Noise Hunt

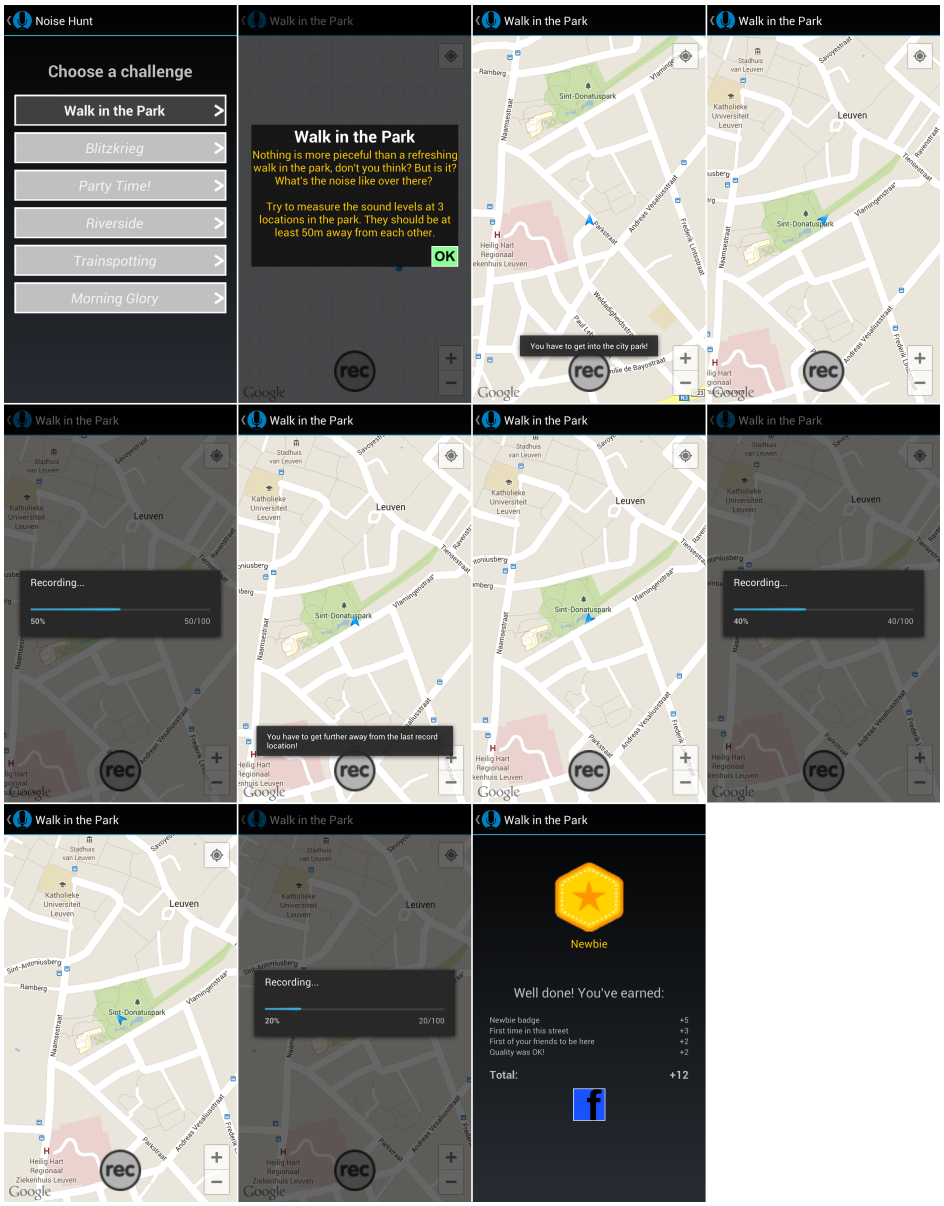


Figure C.8: The Noise Hunt scenario of the full digital NoiseMApp prototype.

C.2.6 View Profile

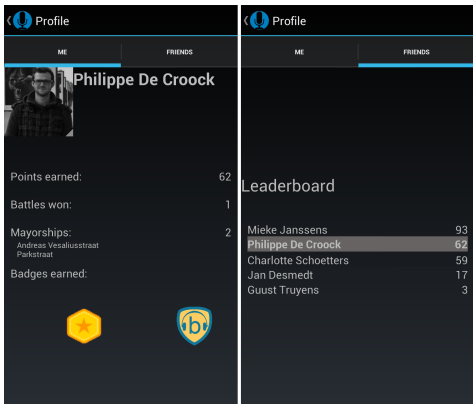


Figure C.9: The View Profile scenario of the full digital NoiseMApp prototype.

C.2.7 Show Map

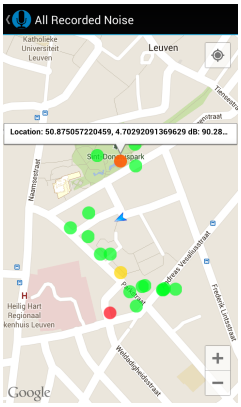


Figure C.10: The Show Map scenario of the full digital NoiseMApp prototype.

C.3 Final Application

The screens that have been changed since the final digital prototype are included.

C.3.1 Sound Battle

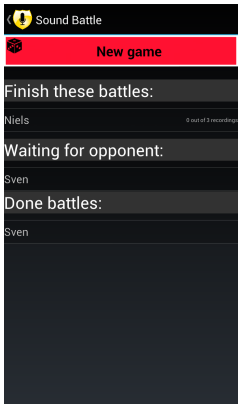


Figure C.11: The Sound Battle create screen of the final NoiseMApp application.

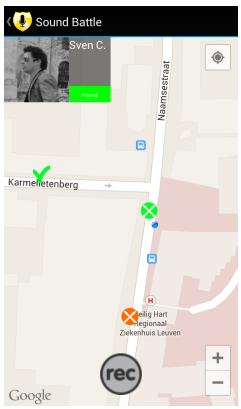


Figure C.12: The Sound Battle screen during game-play with opponent box of the final NoiseMApp application.

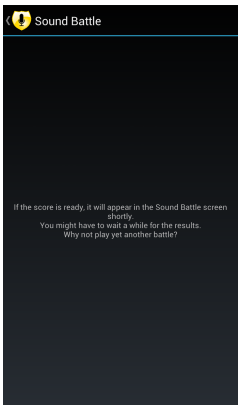


Figure C.13: The Sound Battle wait screen of the final NoiseMApp application.

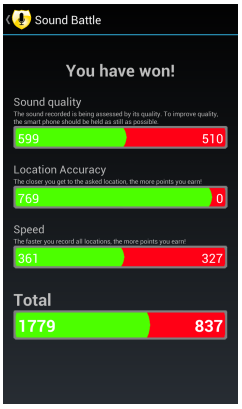


Figure C.14: The Sound Battle points screen of the final NoiseMApp application.

C.3.2 View Profile

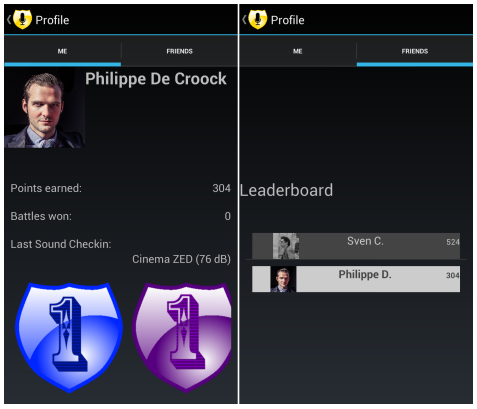


Figure C.15: The View Profile scenario of the final NoiseMApp application.

C.4 NoiseMAppR

NoiseMAppR is a non-gamified version of NoiseMApp. Screens that are the same as in the NoiseMApp application are omitted.

C.4.1 Main screen

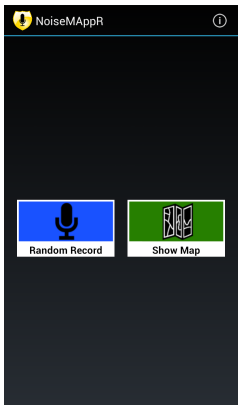


Figure C.16: The main screen of NoiseMAppR.

C.4.2 Random Record

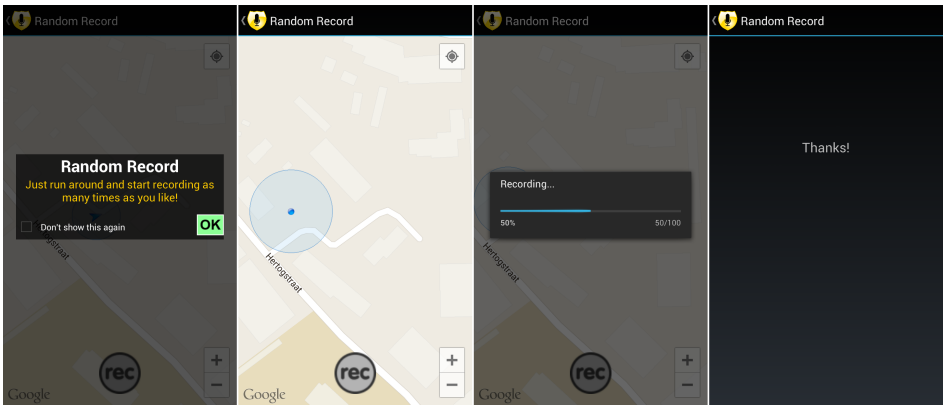


Figure C.17: The Random Record scenario of NoiseMAppR.

Appendix D

Questionnaires

All questionnaires and their summary of results can be found on Github:

- Paper prototype: https://github.com/Phille88/NoiseMApp/tree/master/Evaluation/Prototypes/PaperPrototype
- First digital prototype: https://github.com/Phille88/NoiseMApp/tree/master/Evaluation/Prototypes/FirstDigitalPrototype
- Full digital prototype: https://github.com/Phille88/NoiseMApp/tree/master/Evaluation/Prototypes/FullDigitalPrototype
- Final application: https://github.com/Phille88/NoiseMApp/tree/master/Evaluation/FinalApplication

Appendix E

Class Diagrams

All class diagrams of the Android source code are bundled here per package. They can also be found on Github: https://github.com/Phille88/NoiseMApp/tree/master/Images/ClassDiagrams

E.1 Overview

An overview of the packages is given.

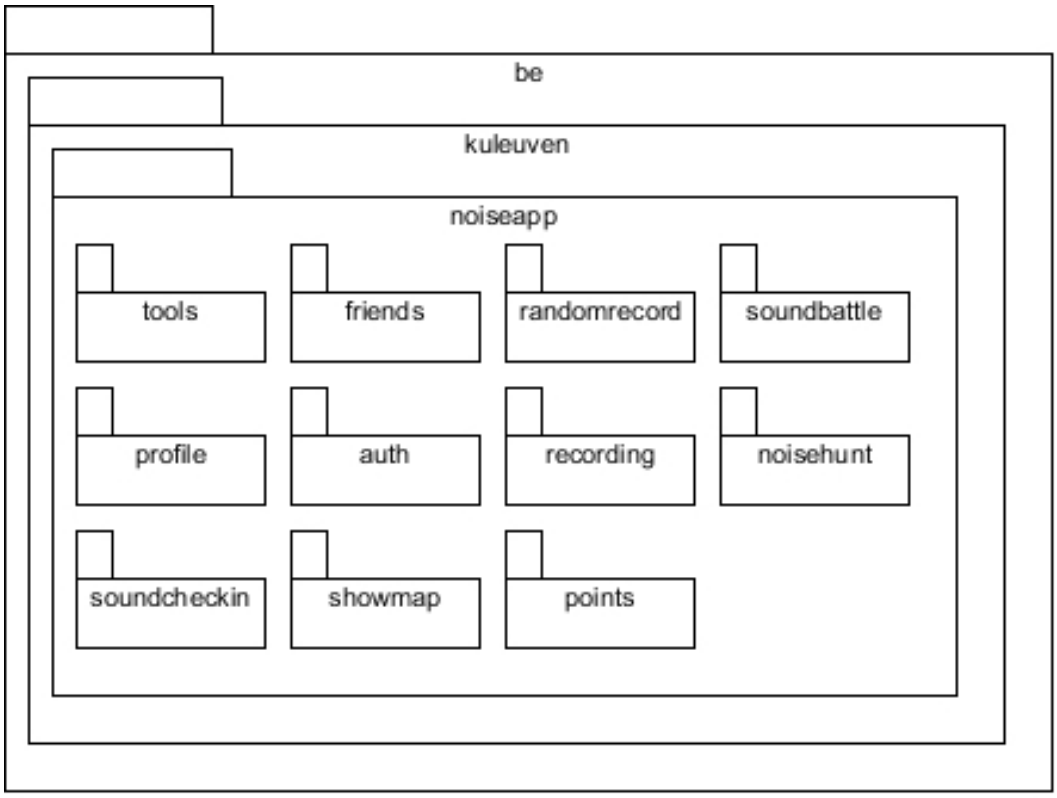


Figure E.1: Overview of the whole NoiseMApp package structure.

E.2 be.kuleuven.noisemapp

An overview of the main source folder is given.

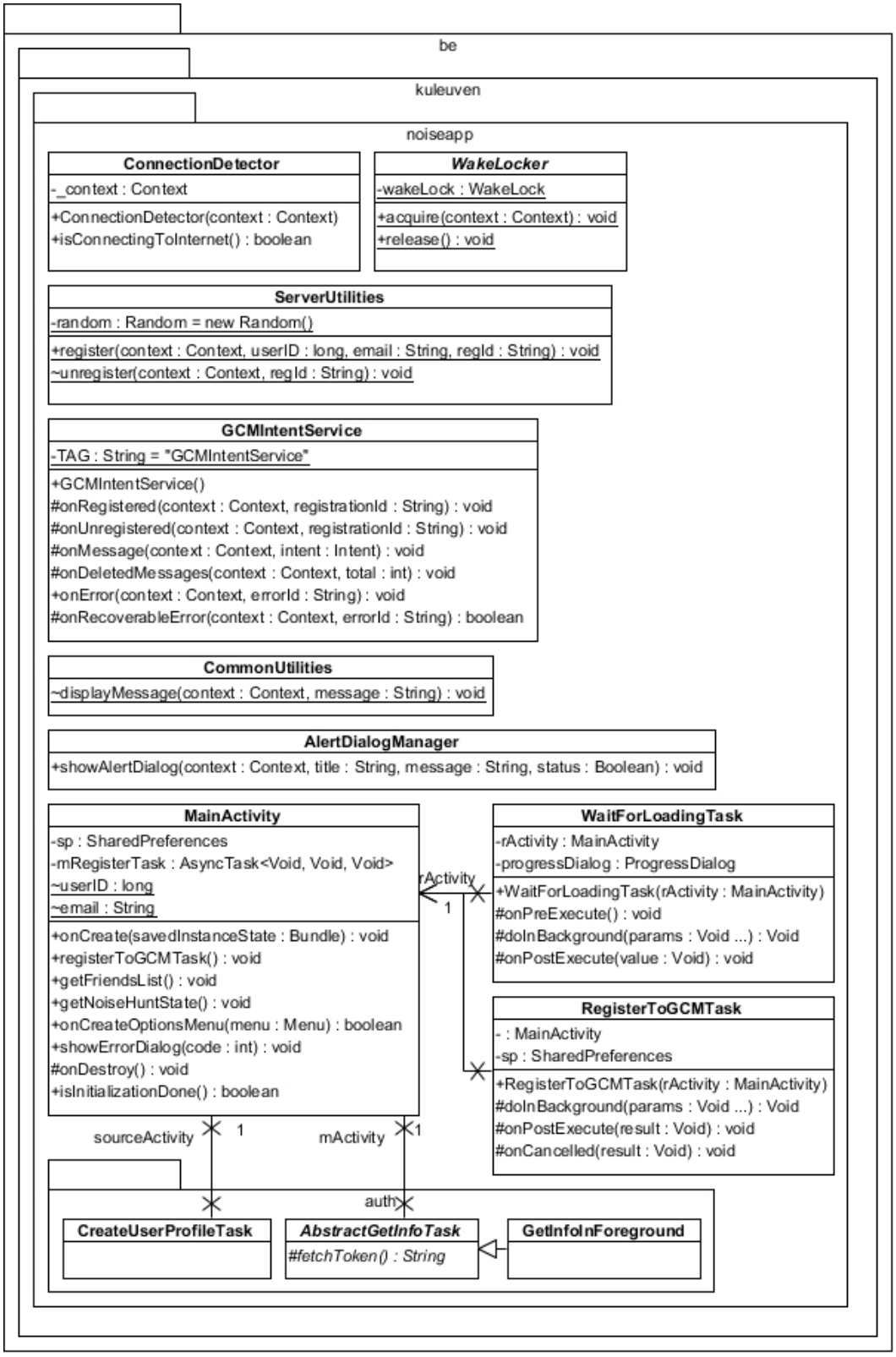


Figure E.2: Overview of the be.kuleuven.noisemapp package.

E.3 be.kuleuven.noisemapp.tools

An overview of the tool package is given.

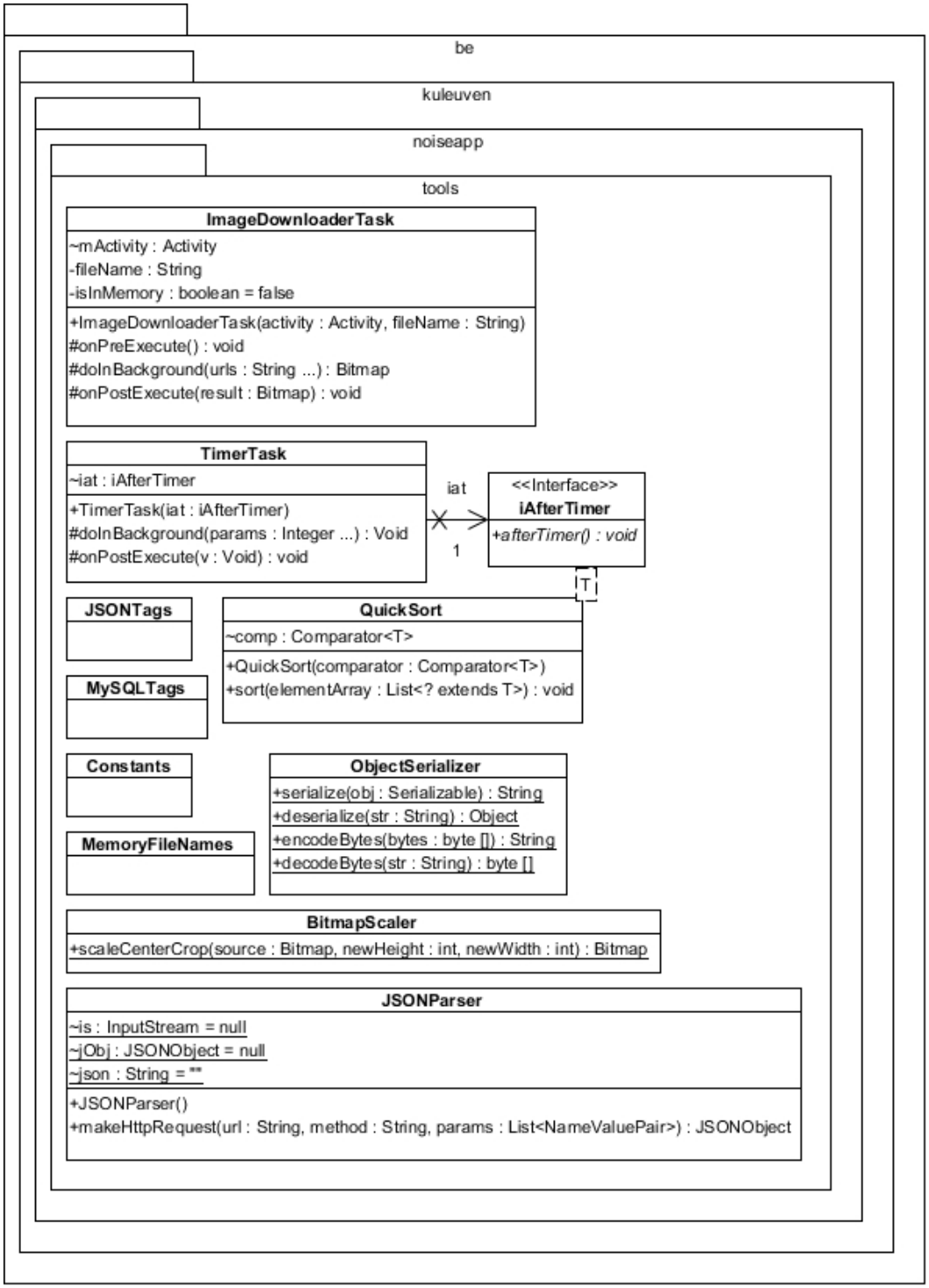


Figure E.3: Overview of the be.kuleuven.noisemapp.tools package.

E.4 be.kuleuven.noisemapp.auth

An overview of the auth package is given.

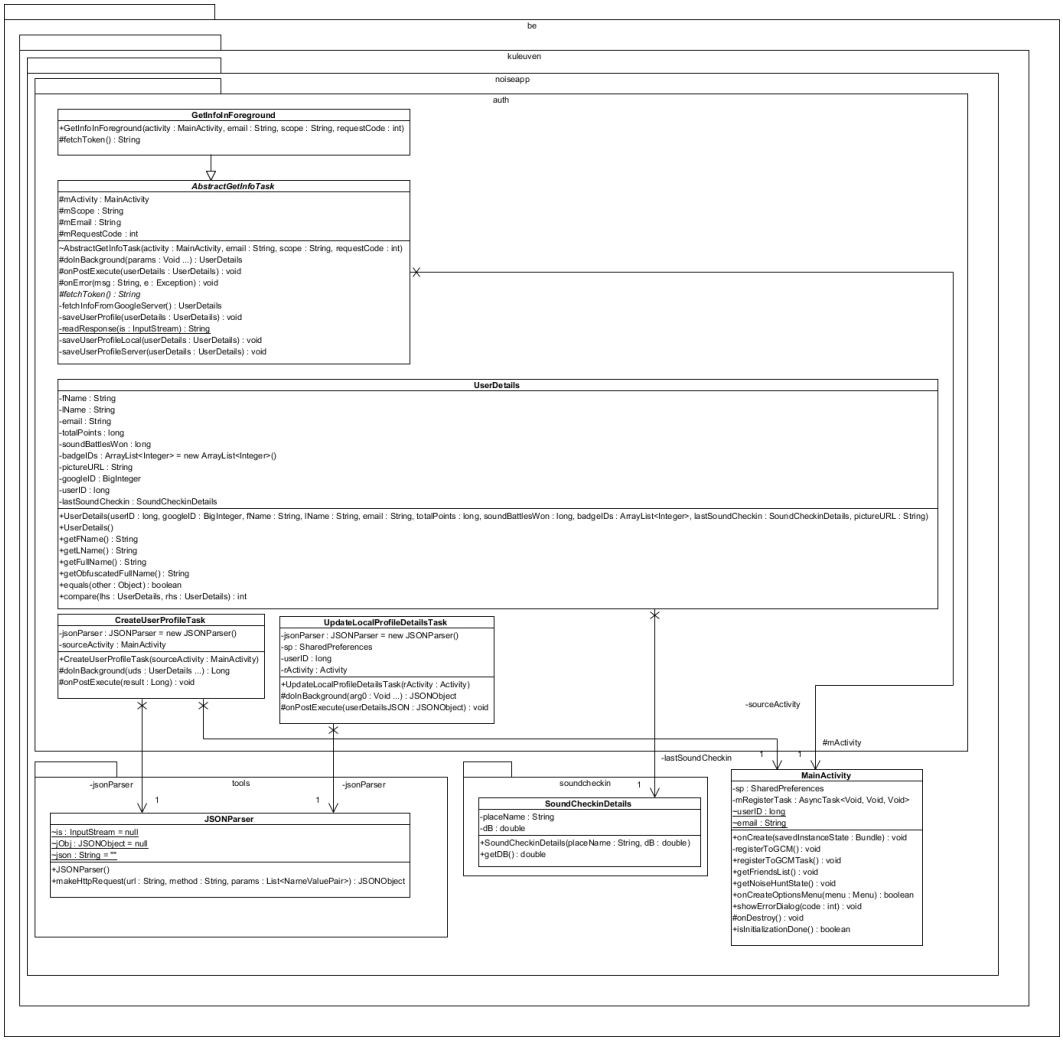


Figure E.4: Overview of the be.kuleuven.noisemapp.auth package.

E.5 be.kuleuven.noisemapp.friends

An overview of the friends package is given.

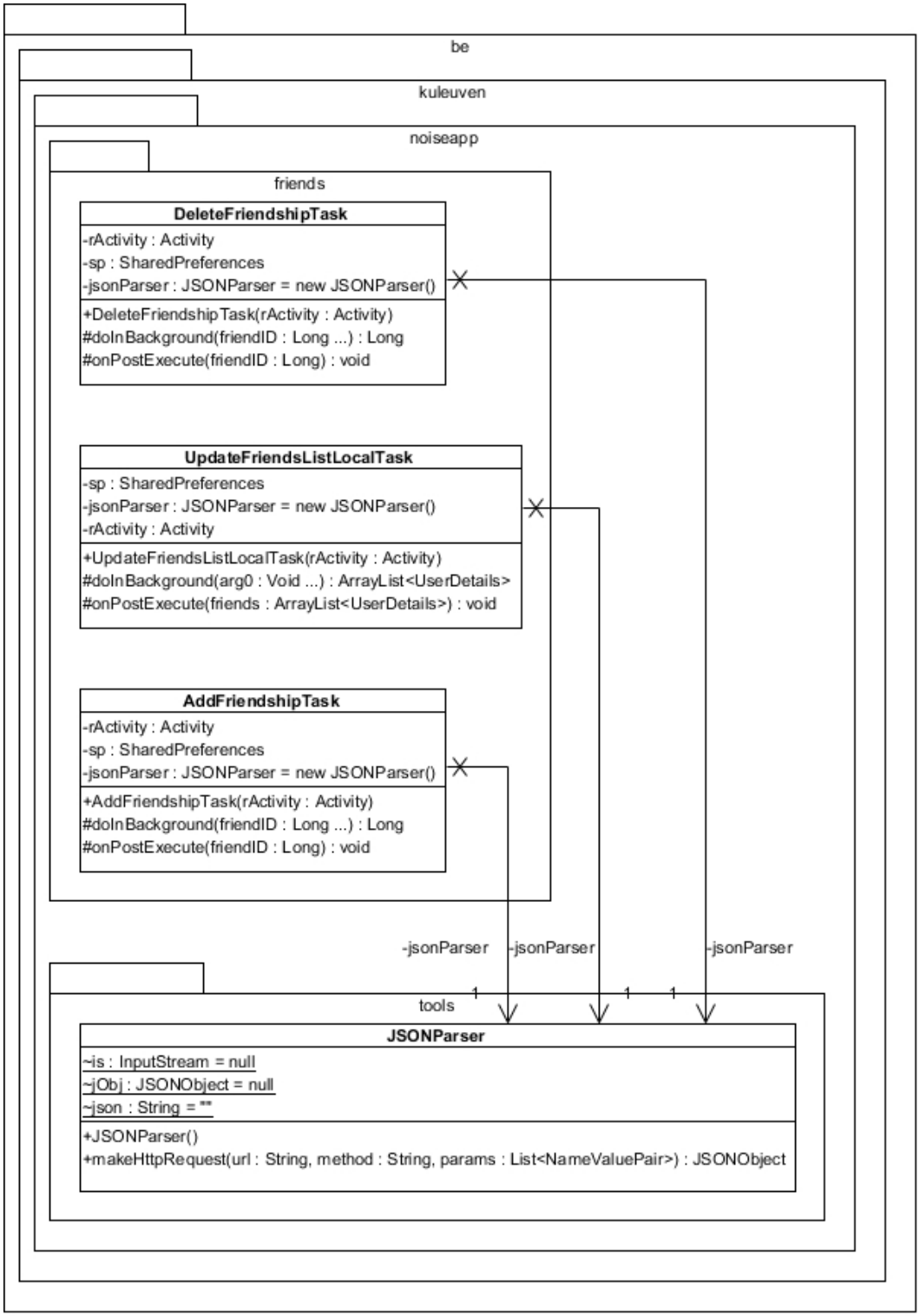


Figure E.5: Overview of the be.kuleuven.noisemapp.friends package.

E.6 be.kuleuven.noisemapp.recording

An overview of the recording package is given.

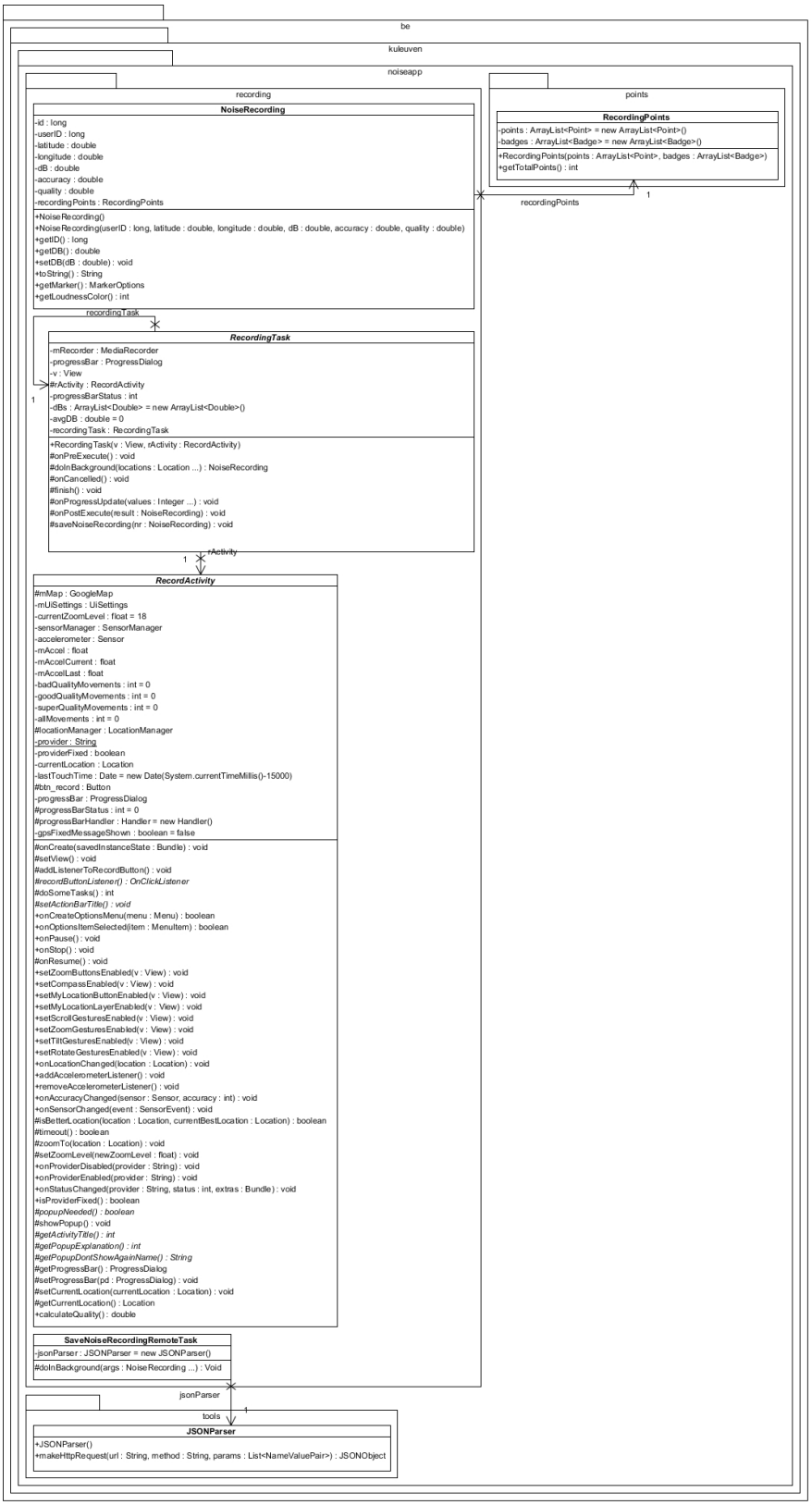


Figure E.6: Overview of the be.kuleuven.noisemapp.recording package.

E.7 be.kuleuven.noisemapp.points

An overview of the points package is given.

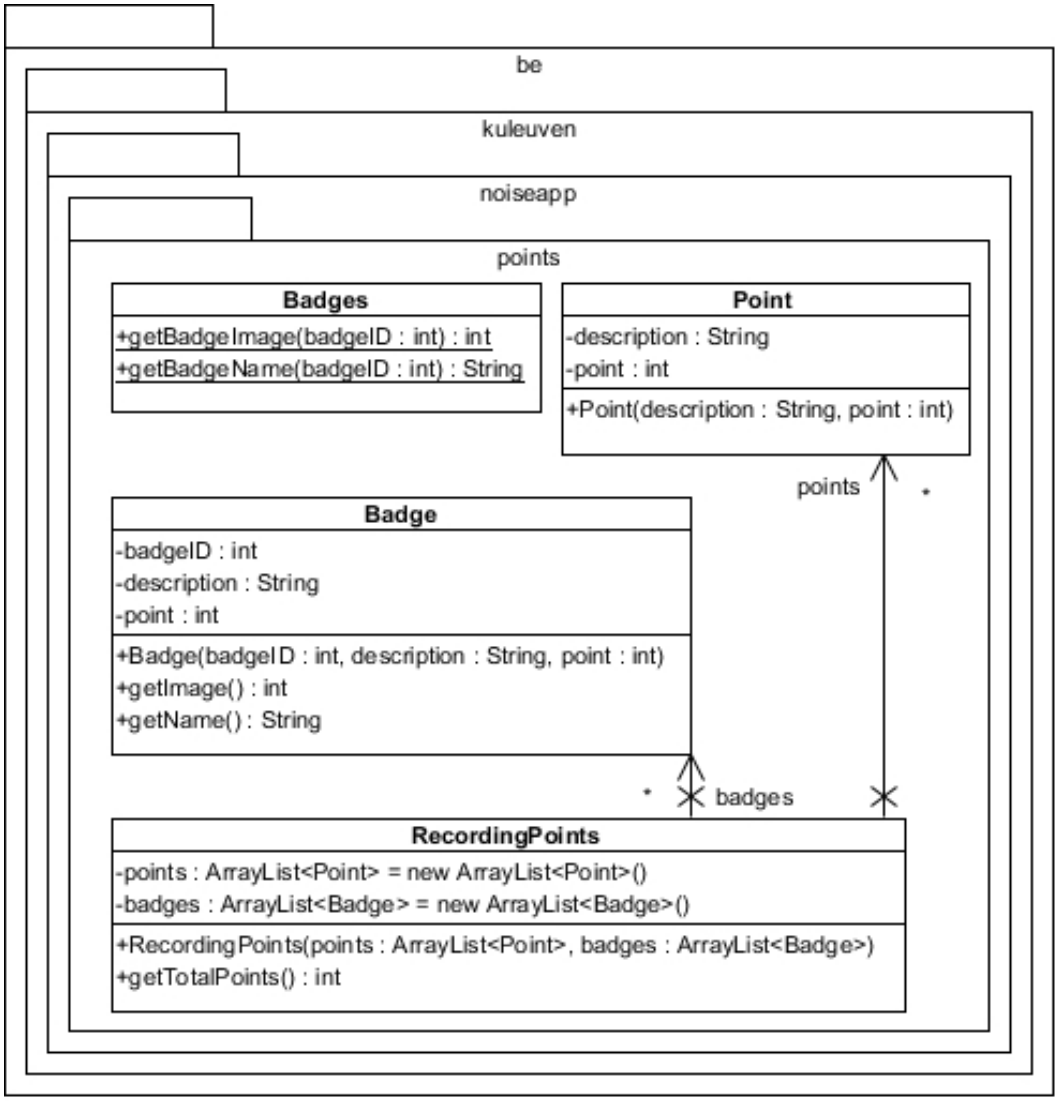


Figure E.7: Overview of the be.kuleuven.noisemapp.points package.

E.8 be.kuleuven.noisemapp.randomrecord

An overview of the randomrecord package is given.

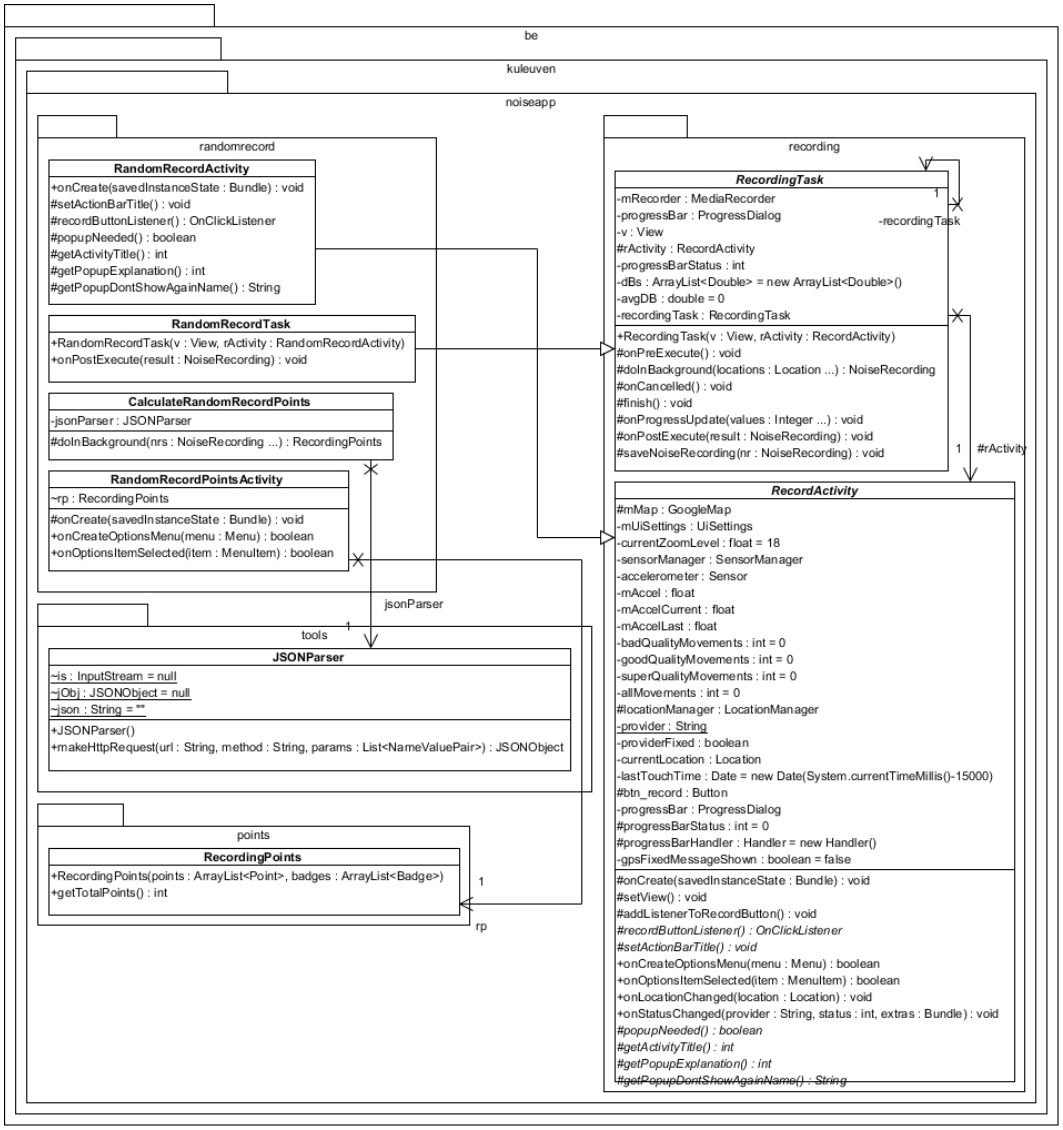


Figure E.8: Overview of the be.kuleuven.noisemapp.randomrecord package.

E.9 be.kuleuven.noisemapp.noisehunt

An overview of the noisehunt package is given.

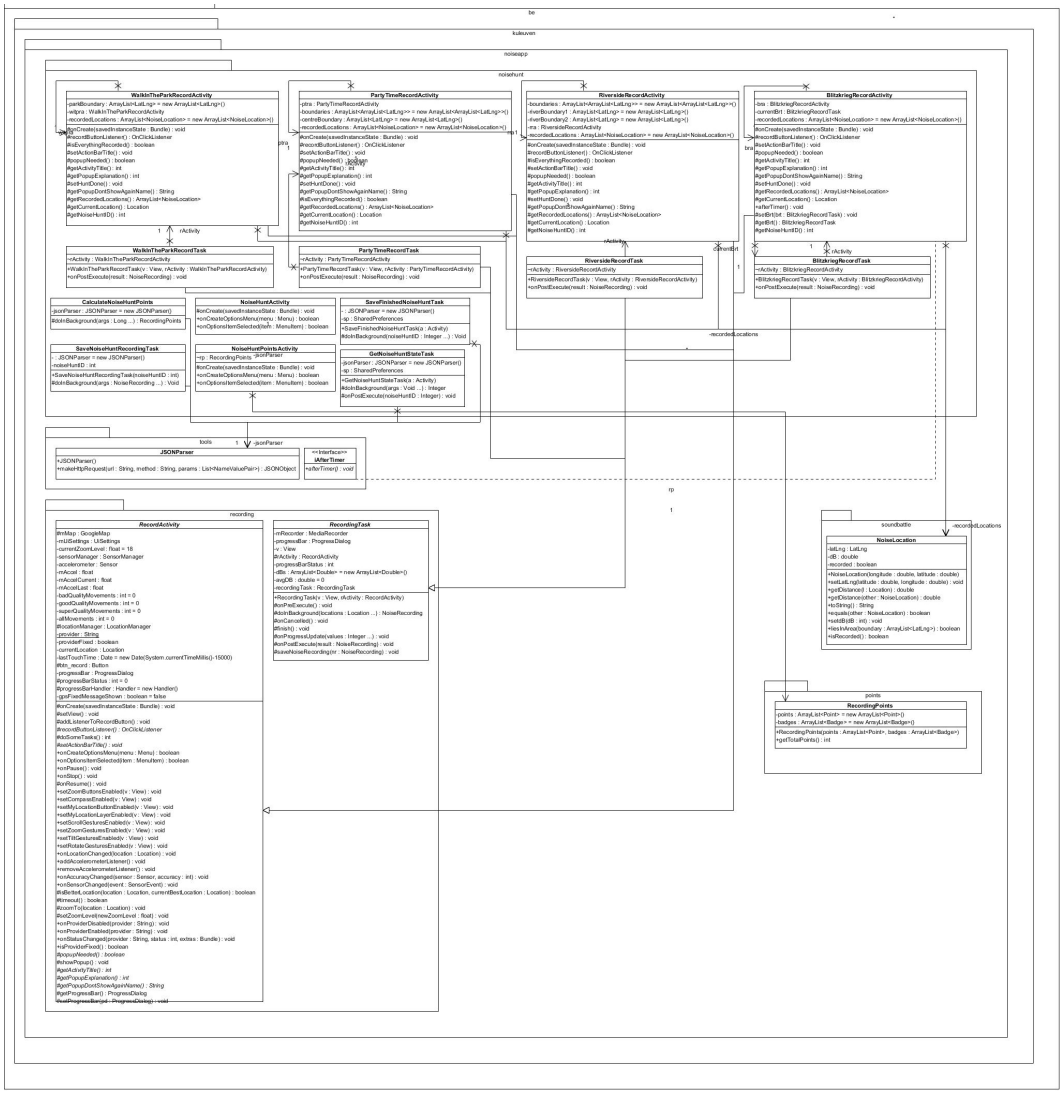


Figure E.9: Overview of the be.kuleuven.noisemapp.noisehunt package.

E.10 be.kuleuven.noisemapp.soundbattle

An overview of the soundbattle package is given.

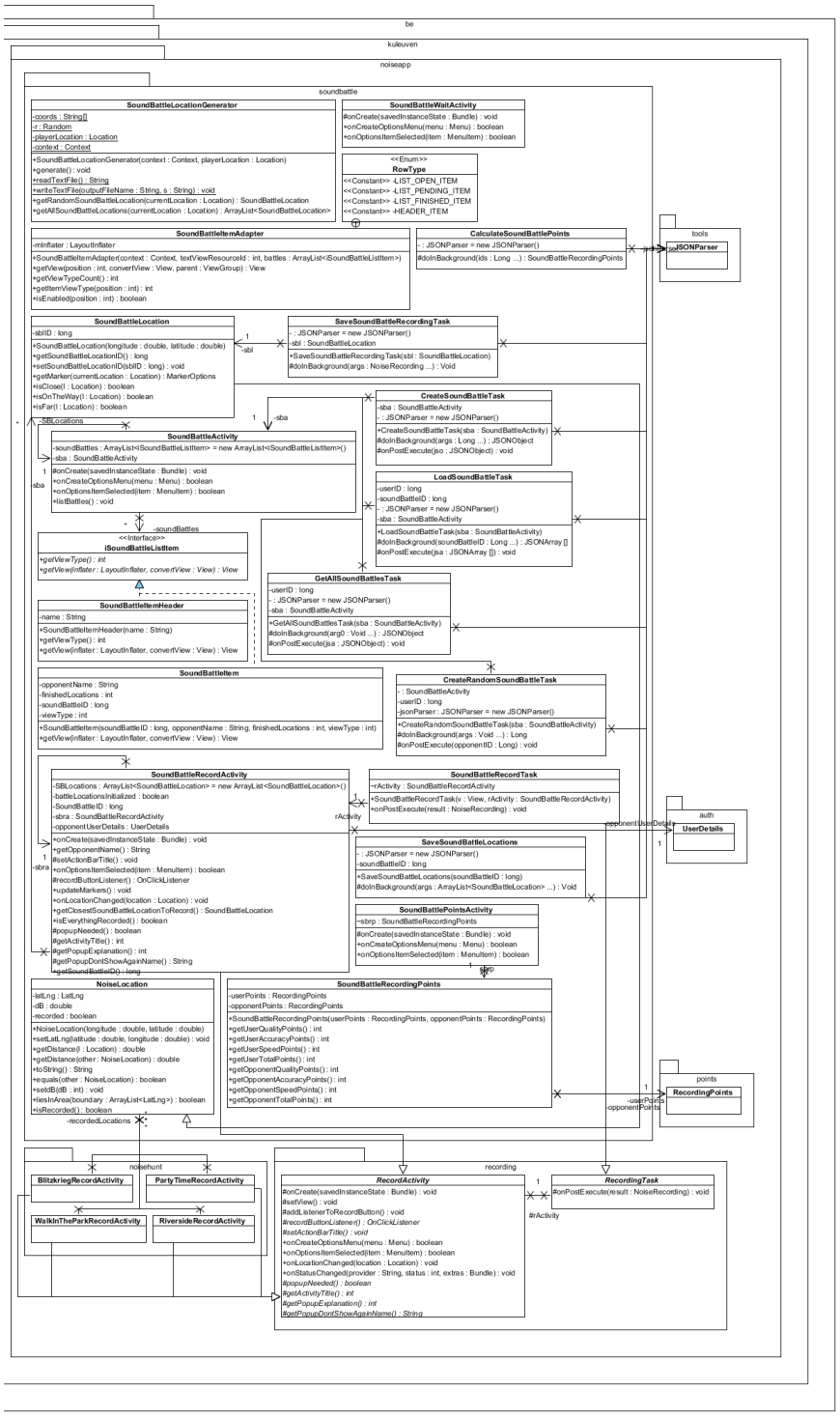


Figure E.10: Overview of the be.kuleuven.noisemapp.soundbattle package.

E.11 be.kuleuven.noisemapp.soundcheckin

An overview of the soundcheckin package is given.

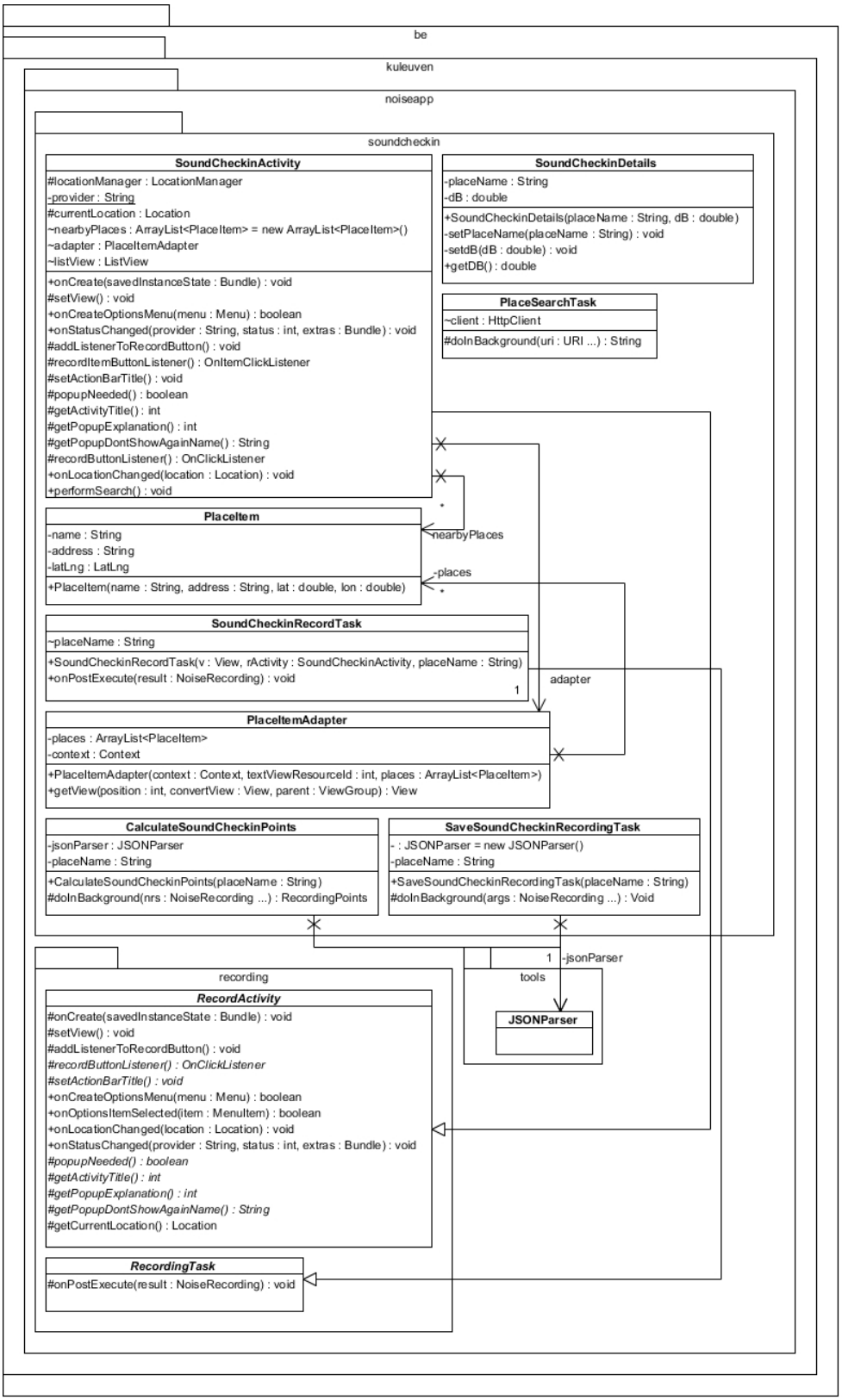


Figure E.11: Overview of the be.kuleuven.noisemapp.soundcheckin package.

E.12 be.kuleuven.noisemapp.profile

An overview of the profile package is given.

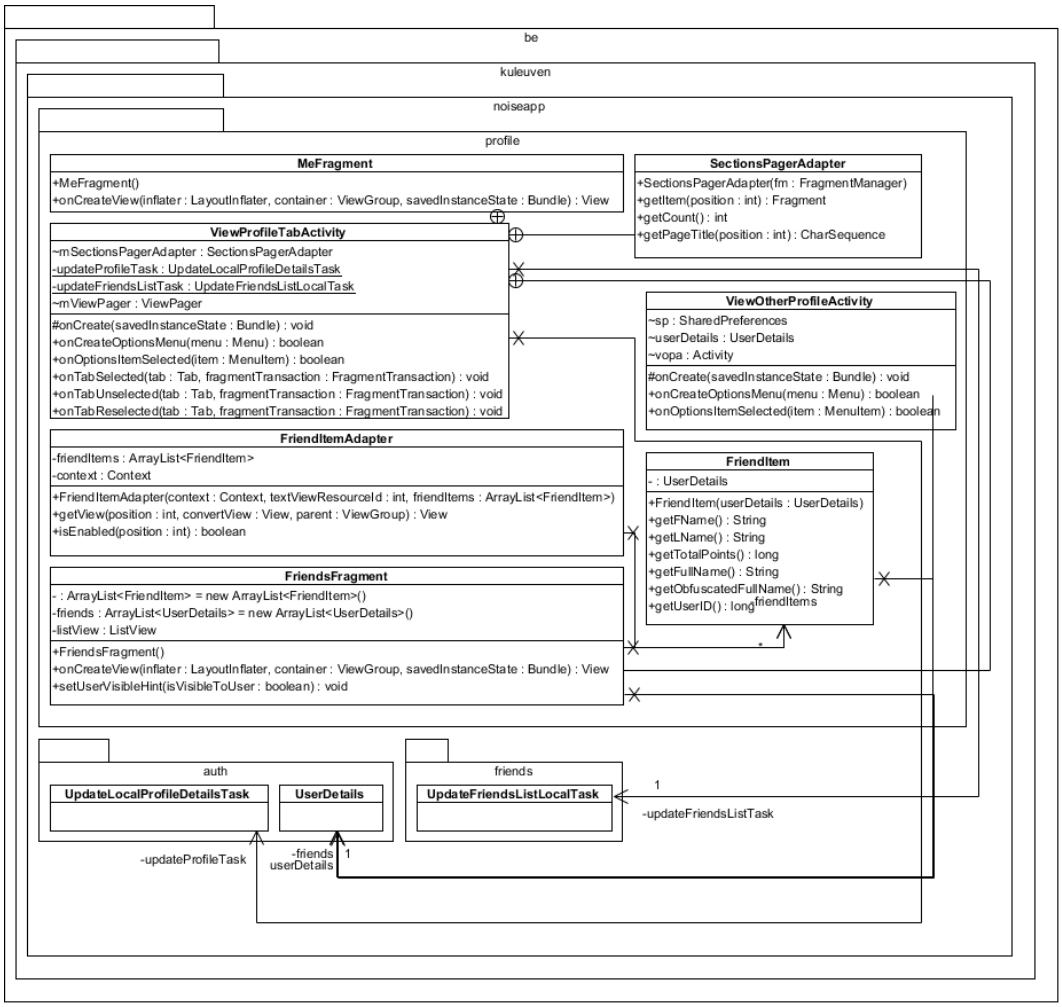


Figure E.12: Overview of the be.kuleuven.noisemapp.profile package.

E.13 be.kuleuven.noisemapp.showmap

An overview of the showmap package is given.

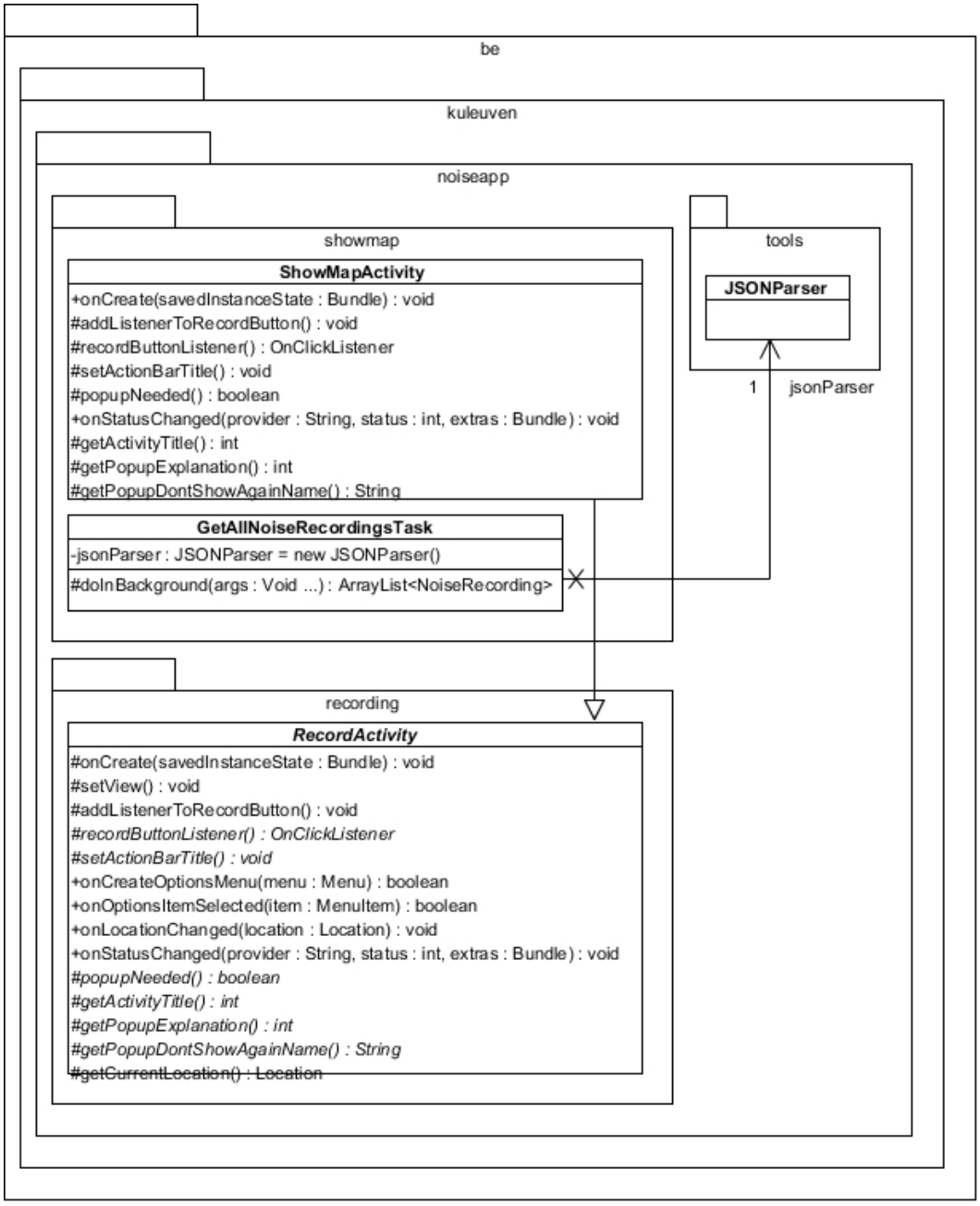


Figure E.13: Overview of the be.kuleuven.noisemapp.showmap package.

Appendix F

Source Code

F.1 Android

F.1.1 Project

The source code of the final application can be found on: https://github.com/Phille88/NoiseMApp/tree/master/SourceCode/Android.

F.1.2 Revisions

All revisions of the source code can be found on: https://github.com/Phille88/NoiseApp

F.1.3 Permissions

All permissions needed by the NoiseMApp application.

1. `android.permission.INTERNET` to make an internet connection;
2. `android.permission.WRITE_EXTERNAL_STORAGE` to write files that are stored in memory;
3. `android.permission.ACCESS_NETWORK_STATE` to
4. `android.permission.ACCESS_COARSE_LOCATION` to access coarse location details of the user;
5. `android.permission.ACCESS_FINE_LOCATION` to access fine location details of the user;
6. `android.permission.RECORD_AUDIO` to record audio;
7. `android.permission.GET_ACCOUNTS` to access Google profile information
8. `com.google.android.providers.gsf.permission.READ_GSERVICES` to access Google profile information;
9. `android.permission.USE_CREDENTIALS` to access Google profile information;
10. `android.permission.WAKE_LOCK` to wake the phones when a message is received;
11. `be.kuleuven.noiseapp.permission.C2D_MESSAGE` to make notifications possible;

12. `com.google.android.c2dm.permission.RECEIVE` to make notifications possible;
13. `android.permission.VIBRATE` to let the phone vibrate when receiving a notification.

F.2 PHP

The source code of the PHP back-end can be found on: https://github.com/Phille88/NoiseMApp/tree/master/SourceCode/PHP.

Bibliography

[1] R. C. Bell. *Board and Table Games from Many Civilizations*. Dover Publications, rev sub edition, 1980.

[2] V. Blaylock. Gamification 3 - Game Thinking, 2012.

[3] J. Brooke. SUS-A quick and dirty usability scale. *Usability evaluation in industry*, pages 1–8, 1996.

[4] R. Caillois. *Man, Play, and Games.* University of Illinois Press, Urbana, Chicago, 2001.

[5] J. CARDINAELS, F. PIETERMAAT, and P. ROMBOUTS. Leuven keurt controversiële geluidsregels goed, May 2013.

[6] A. I. Chorney. Taking the game out of gamification. *DJIM*, 8(Spring 2012):26, 2013.

[7] Y. Chou. Gamification Research: How FarmVille uses Game Mechanics to become Winning & Addicting, 2012.

[8] E. Commission and Others. Green paper on future noise policy. *Brussels: European Commission*, 1996.

[9] G. Costikyan. Game styles, innovation, and new audiences: An historical view. 2005.

[10] D. Crowley and N. Selvadurai. Foursquare, 2009.

[11] De Redactie. Nieuwe geluidsnormen definitief goedgekeurd, Feb. 2012.

[12] Departement van Leefmilieu, Natuur en Energie. Goedgekeurde geluidskaarten. http://www.lne.be/themas/hinder-en-risicos/geluidshinder/beleid/eu-richtlijn/goedgekeurde-geluidskaarten.

[13] S. Deterding and D. Dixon. Gamification : Using Game Design Elements in Non-Gaming Contexts. pages 2425–2428, 2011.

[14] S. Deterding, D. Dixon, R. Khaled, and L. Nacke. From Game Design Elements to Gamefulness : Defining Gamification. In *Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments*, volume Tampere, F of *MindTrek '11*, pages 9–15, New York, NY, USA, 2011. Acm.

[15] dotPDN LLC. Paint.NET.

[16] D. Etherington. Android Nears 80% Market Share In Global Smartphone Shipments, As iOS And BlackBerry Share Slides, Per IDC, 2013.

[17] M. Firtman. HTML5 compatibility on mobile and tablet browsers with testing on real devices, 2013.

[18] Het Europees Parlement en de Raad van de Europese Unie. RICHTLIJN 2002/49/EG VAN HET EUROPEES PARLEMENT EN DE RAAD van 25 juni 2002 inzake de evaluatie en de beheersing van omgevingslawaai. *Publicatieblad van de Europese Gemeenschappen*, L 189:12–25, 2002.

[19] Het Laatste Nieuws. "Leuven heeft tegen 2015 nood aan 3.700 extra koten", Mar. 2011.

[20] K. Huotari. Defining Gamification - A Service Marketing Perspective. pages 17–22, 2012.

[21] KU Leuven. Geluidsoverlast in Tiensestraat: stewards brengen rust, Dec. 2000.

[22] N. Lazzaro. Why We Play Games: Four Keys to More Emotion Without Story. In *Game Developers Conference*, 2004.

[23] Lokale Politie Leuven. Projectmatige aanpak uitgaansoverlast. urlhttp://www.lokalepolitie.be/5388/uitgaan/125-project-aanpak-uitgaansoverlast/320-projectmatige-aanpak-geluidsoverlast.html.

[24] S. Long, R. Kooper, D. Abowd, and G. Atkeson. Rapid Prototyping of Mobile Context-Aware Applications : The Cyberguide Case Study Aware Application Context- Related. 1996.

[25] H. Lu, W. Pan, N. D. Lane, T. Choudhury, and A. T. Campbell. SoundSense : Scalable Sound Sensing for People-Centric Applications on Mobile Phones. pages 165–178, 2009.

[26] E. Luis. Noise Battle : A Gamified application for Environmental Noise Monitoring in Urban Areas.

[27] I. G. Martí, L. E. Rodríguez, M. Benedito, S. Trilles, A. Beltrán, L. Díaz, and J. Huerta. Mobile application for noise pollution monitoring through gamification techniques. In *Proceedings of the 11th international conference on Entertainment Computing*, ICEC'12, pages 562–571, Berlin, Heidelberg, 2012. Springer-Verlag.

[28] J. McGonigal. *Reality Is Broken: Why Games Make Us Better and How They Can Change the World*. Penguin Group , The, 2011.

[29] Officiële site van de stad Leuven. Aantal inwoner, 2012.

[30] B. Reeves and J. L. Read. *Total Engagement: Using Games and Virtual Worlds to Change the Way People Work and Businesses Compete*. Harvard Business School Press, 1 edition, Nov. 2009.

[31] P. V. Rompuy. JONGCD&V Leuven pakt geluidshinder aan. http://www.politico-shock.be/peter/2012/jongcdv-leuven-wil-een-geluidskaart.html.

[32] J. Sauro. MeasuringUsability.

[33] I. Schweizer, R. Bärtl, and A. Schulz. NoiseMap-real-time participatory noise maps. *Proc. 2nd Int'l ...*, 2011.

[34] C. Snyder. *Paper Prototyping: The Fast and Easy Way to Design and Refine User Interfaces*. 2003.

[35] R. Soute. The Decibel Done Right : A Matter of Engineering the Math. (2), 2009.

[36] T. S. Tullis and J. N. Stetson. A Comparison of Questionnaires for Assessing Website Usability. pages 1–12, 2004.

[37] M. W. van Someren, Y. F. Barnard, and J. A. C. Sandberg. *The Think Aloud Method: A Practical Guide to Modelling Cognitive Processes*. Academic Press Limited, London, 1994.

[38] K. Werbach and D. Hunter. *For the Win: How Game Thinking Can Revolutionize Your Business*. 2012.

[39] P. H. T. Zannin, A. M. C. Ferreira, and B. Szerametta. Evaluation of noise pollution in urban parks. *Environmental monitoring and assessment*, 118(1-3):423–33, July 2006.

Master thesis filing card

Student: Philippe De Croock

Title: NoiseMApp: Generating Noise Level Maps through Gamification

Dutch title: NoiseMApp: Genereren van Geluidsniveaukaarten met behulp van Gamificatie

UDC: 621.3

Abstract:

Generating noise level maps can be expensive and labour-intensive. This thesis presents a way to gamify a noise level mapping application in order to engage users to map noise levels with their smart phones in Leuven. Through Rapid Prototyping an application has been developed and evaluated. The results is NoiseMApp, which has been evaluated and compared to a non-gamified version. More noise recordings were contributed by users of NoiseMApp than the non-gamified version.

Thesis submitted for the degree of Master of Science in Engineering: Computer Science, major Distributed Systems

Thesis supervisor: Prof. dr. ir. Erik Duval

Assessors: Frans Van Asche
Prof. dr. Danny Hughes

Mentors: Gonzalo Alberto Parra Chico
Jose Luis Santos Odriozola