Gamified Mobile Application and Stairstep Counter for Stair Walking

Linn Danielsson

Computer Science and Engineering, master's level
2018

Luleå University of Technology
Department of Computer Science, Electrical and Space Engineering
Abstract

Gamification is an area that has attracted a substantial amount of interest in recent years and is something that can be applied almost everywhere. Health is also an area that is frequently discussed regarding how a person’s health and daily exercise can be improved. Applying gamification to motivate people to exercise more is something that has been done before although they are usually marketed as health application. What if an application’s main purpose was not to count the amount calories burnt but instead the exercise is an additional part? In this thesis you will find the development of a prototype that is a gamified Android application with the purpose to motivate people to walk more in stairs. In order to develop an application there was a need for a stairstep counter to count the number of steps a person has taken in a stair. Hence a stairstep counter was developed using the open source machine learning framework, TensorFlow, with classification and neural networks. The counter uses the y- and z-axis of the mobile accelerometer to classify if a movement of a person was a step or not. It was trained with a custom dataset that was created using data from walking in a stair and data collected from other movements. Resulting stairstep counter used in the application is a counter that counts stairsteps rather accurately. The final result of this thesis was a mobile application that uses animation and competition elements to enhance a person’s stair walk experience.
Preface

First I want to thank all the educators in all the courses I have had throughout the years at Luleå University of Technology, I have learned so much. I also want to thank Peter Parnes for being my supervisor, Lena Jonsson at Arctic Group for providing me with this opportunity and the people at Arctic Group for their help, I am ever so grateful.
## Contents

1 Introduction ........................................ 1
    1.1 Background .................................. 1
    1.2 Related Work .................................. 2
    1.3 Purpose and Problem ........................... 2
    1.4 Delimitation .................................. 3
    1.5 Structure .................................... 3

2 Theory .............................................. 4
    2.1 Gamification ................................... 4
        2.1.1 Definition .................................. 4
        2.1.2 Game Elements ............................... 5
        2.1.3 The Behavioural Aspect ...................... 6
    2.2 Accelerometer .................................. 8
        2.2.1 Collecting Data .............................. 9
    2.3 Machine Learning ............................... 9
        2.3.1 Classification and Neural Networks ........... 9
        2.3.2 Preprocessing, Features and Labels ........... 11
        2.3.3 TensorFlow .................................. 12

3 Method .............................................. 13
    3.1 Work Process ................................... 13
    3.2 Use Case ...................................... 14
    3.3 Technical Choices ............................... 15
        3.3.1 Hardware .................................... 15
        3.3.2 Software ................................... 15
    3.4 Stairstep Counter ............................... 16
        3.4.1 Dataset ..................................... 16
        3.4.2 The Model ................................... 18
        3.4.3 Before Training ............................... 19
        3.4.4 Training the Model ........................... 20
    3.5 The Game Elements ............................... 20
        3.5.1 Animation ................................... 21
        3.5.2 Competition ................................ 22
    3.6 The System ..................................... 23
        3.6.1 Application ................................ 23
        3.6.2 Database ................................... 26
        3.6.3 Server ..................................... 27
    3.7 Technical Result ................................. 29
List of Figures

2.1 Picture of the accelerometer’s orientation in a smart phone. . . . . . . 8
2.2 Collected data from an accelerometer of a person walking. . . . . . . 10
3.1 Use case for the application. . . . . . . . . . . . . . . . . . . . . . . . . . 14
3.2 Developed Android application for collection of accelerometer data. . 17
3.3 A graphical representation of collected step data in one session. . . 18
3.4 Result of training the model with the final dataset. . . . . . . . . . . . 20
3.5 Screenshots from application when the ball from the animation ele-
    ment is moving one step up, from left to right. . . . . . . . . . . . . . . 21
3.6 Illustration of the development of the ball’s movement. . . . . . . . . 22
3.7 Simplified architecture of the system. . . . . . . . . . . . . . . . . . . . 23
3.8 Flowchart of the login procedure when user first start application. . 24
3.9 UML diagram of the implementation of step counter activity. . . . . 25
3.10 UML diagram of the implementation of server request with example
    activity. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 25
3.11 EER diagram of the MySQL database. . . . . . . . . . . . . . . . . . . 26
3.12 The first view a user will see depending if they are already logged in
    or not. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 31
3.13 The main view of the two major game elements. . . . . . . . . . . . . 32
3.14 How invitations for a user are displayed and accepted/declined. . . 32
3.15 How more detailed information about competition is displayed and
    view of possible competitions to join. . . . . . . . . . . . . . . . . . . 33
3.16 A user can view, create competitions and invite users to private com-
    petitions. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 33
4.1 The response from the question regarding the animation element. . . 35
4.2 The response from the question regarding the competition element. . 36
4.3 The response from the question regarding the high score element. . . 36
4.4 The response from the question regarding the stairstep counter. . . . 37
List of Tables

2.1 Scipada’s et al. result of classified game design elements. . . . . . . . 6
2.2 Summarization of different personality types and their motivations. . 8
2.3 Some TensorFlow terminology with description. . . . . . . . . . . . . . 12

3.1 The different work parts of the thesis. . . . . . . . . . . . . . . . . . . 14
3.2 The different kind of statuses a competition can have. . . . . . . . . . 27
3.3 Table of the different API routes, its HTTP method and description. 29
3.4 Test results of trained model using the final dataset. . . . . . . . . . . 30
Chapter 1

Introduction

1.1 Background

The interest of gamification has increased immensely in recent years. The industry has recognized the potential in using game elements and how it can increase people’s motivation. One example of gamification you could find in the everyday life is the point system or stamps on cards some food chains uses. They can offer a deal similar to ”get ten stamps and get one for free” to motivate a consumer to purchase their products. Another example is different websites, like Stack Overflow [40], that uses upvotes and medals which a user can receive. An informal definition of gamification could be the use of elements that are usually found in games which then are applied to everyday life situations.

Something else frequently discussed is people’s health and how to improve it since it has several benefits [60]. Some people want to be healthier but they might not have the time or motivation to exercise every other day. The solution could be to increase their daily exercise instead. Now, the combination of gamification and exercise is something that has been done before, like ”Zombies, run!” [55]. The purpose of gamification in this context is to generate a behavioural change that preferably lasts for a longer period of time. There is a sea of articles and studies in that research area and there is no shortage of different health applications applying some kind of gamification either. A health application is an application, usually mobile, whose purpose it to help the user with for example training goals or keep track of what they eat during the day. However, there are not that many application for people who want to walk in stairs. Those that exist usually requires some additional equipment or do not count the number of steps taken but only the altitude.

Most mobile applications that involves any kind of step counter and gamification are often marketed as a health application with the main focus on how many calories the user has burnt. The thing is that exercise does not have to be the main purpose of the application, the exercise could be an additional benefit. An example is Pokémon GO [38] that blew up the summer of 2016. The application is marketed as a game but the interesting thing is how it got people to walk. People did not play the game because they wanted to walk more, instead it was a pleasant surprise that they got exercise using the application.
1.2 Related Work

As mentioned in previous section, there are not that many applications available for specifically stairs. Some health applications have a feature that the user can use so they can see estimated altitude they have walked, but it usually is not possible to see number of stair steps the user has walked. There is one application, called StepJockey [2] that is a health and company oriented mobile application. It keeps track of how many calories that has been burnt and offers the possibility to participate in challenges. The application itself however does not count the number of steps taken by a user, instead there is a need for something called “smart sign”, which is NFC-equipped and placed at the beginning and end of the stairs. To create a smart sign you need the number of flights and the height of the stair step. It is not possible to use the application and walk in a stair that does not have this smart sign.

Something that is not related with stairs but whose goal was to motivate people was the work performed by Takahashi et al. [51] who developed a walking game to motivate elderly people to go outside more. In Japan there is something called “tojikomori” that has become a growing issue. “Tojokomori” is a phenomena where elderly people keep themselves in their room or house and do not go outside. As a solution to this problem Takahashi et al. created an application with a groupwalking program to “enhance going out for older adults”, using different stamps they could collect at locations in the city.

Combining games and exercise, like exergaming [19], is something that is rather well established. Exergaming integrates functional and interactive fitness to make exercising more fun and engaging. Another example is Wii U and their exercising games [23]. They have for example games for dancing or playing tennis. Exergaming and Wii U are however not that focused on the daily exercises that you could for example achieve by bicycling to work or taking the stairs instead of the elevator.

There are tons of different applications that can help people with their exercising. Just searching for a walking or running application in the app store will give you a result of hundreds of applications. As Robert C. Pozen states in his article about how exercise increases productivity [43], “Everyone knows that exercise can improve your health”, but as the article also states is that people find it difficult to exercise regularly even with all the benefits exercising comes with.

1.3 Purpose and Problem

The purpose of this thesis was to study how game elements should be developed for most engagement and to examine a way to count a stair step when a person walks in a stair. An Android application was also developed whose purpose was to apply gamification to the non-game-context of walking in a stair. The questions that was to be answered were:

- How can a stair step be counted?
- What game elements are best in this context to induce a behavioral change?
• What requirements are there to a gamified system?
• What techniques are most suitable for the system?

The purpose of the question ”How can a stairstep be counted?” is to examine if and how some kind of stairstep counter can be implemented to count a stairstep when a person walks in a stair.

To clarify the question ”What game elements are best in this context to induce a behavioral change?”, the purpose of game elements is to motivate but it does not matter how good the elements are, if they are not compatible with the context they may not have the desired result.

The question ”What requirements are there for a gamified system?” is somewhat related to the previous question, to develop good game elements there should be a number of requirements to think about so they work with chosen context.

1.4 Delimitation

This thesis will describe how a stairstep counter was developed and implemented in an gamified Android application. It will not be a step by step description how the application was developed but instead it could be used as a guide that provides some solutions how such application could be developed. The resulting application of this thesis is also more of a prototype and not a finished product.

1.5 Structure

This report begins with Chapter 2, Theory, that describes gamification, accelerometer and machine learning for better understanding of the thesis. The chapter will first go through theory regarding the gamification concept: its definition, what a game element is, what already exists and the behavioural aspect of gamification. The chapter will also describe the accelerometer and how data is collected from the sensor. Finally the chapter will go through machine learning, it will describe concepts like classification, neural networks, preprocessing, features, labels and a short explanation about TensorFlow and how it works.

The next chapter, Chapter 3, is the method and will go through how everything was developed. First it will describe the work process, then use case and some technical choices regarding hardware and software. It will also describe how the stairstep counter was developed, how the dataset and the model was created, the process before training and finally how the model was trained. It will also show the final result of the stairstep counter and application.

Chapter 4 will consist of an evaluation of the application. Chapter 5, the conclusion, will consist of future work, some ethics, discussion and finally the conclusion to finish of the thesis.
Chapter 2

Theory

This chapter is the result of the first part of the thesis, the research. First it will go through gamification and its definition for those that might not be familiar with it. After reading several papers [57, 8, 27, 42, 49] a number of game elements were summarized and categorized depending on what kind of element they were. The section will also go through the more behavioural aspect of gamification, what motivates peoples and their personality types. The result of the gamification research part was used to derive game elements that were later used in the mobile application. The chapter will also briefly go through the accelerometer sensor, how the sensor works and how data is collected using sampling rate. The theory regarding machine learning used for the stairstep counter will also be described.

2.1 Gamification

Gamification is a fairly new phenomena and the industry has just begun realizing how efficient it can be to motivate people, both in the workplace and in their personal lives. Research within the field has grown significantly in recent years so even if it is a rather new area there is no shortage of research papers [25, 57, 5, 48, 3, 28, 27, 26, 36, 21, 22, 56].

2.1.1 Definition

Let us begin with the definition of gamification, what is it? You have most likely encountered some kind of gamification in your everyday life, without thinking about it. For example the bonus points you get when you draw your card buying food, or when you have just signed up on a website and you see a little progress bar when filling in your personal information. That is what you would call gamification elements, but more on that later.
Huotari et al. [25] proposed a new definition in 2012 of gamification in a service marketing perspective. They stated using an example with a geographic game and a park, that it is not gamification if a geographic game brings people to the park, but it is gamification if the park uses the geographic game and offers it to the people that visits the park. According to them the term gamification comes from a blog post by Brett Terill in 2008, describing gamification as ”taking game mechanics and applying them to other web properties to increase engagement”.

In recent year gamification has been defined as ”The use of game design elements in non-game-context” [57, 5, 48]. So if we use the example from before with filling in your profile, then the game design element is the progress bar and filling in your personal information is the non-game-context. The non-game-context should be the main purpose of the application and it is important that it is not undermined by the game design elements, the elements should only support it [3, 28].

2.1.2 Game Elements

Game elements are what they sound like, they are elements that are retrieved from real games. The elements could be for example challenges, levels, avatars, points, achievements, stories or leaderboards [57]. With the recent rise of gamification the number of existing game elements have risen with it. However, there are some game elements that are more frequently used in today’s gamified applications than others. They can be called the most common ones, that is points, rewards, badges, progress bar and leaderboard [8]. Usually a game element is not used on its own, but it is combined with another game element. For example points could be combined with rewards, so that enough points results in that the user receives a reward et cetera. This is probably the most common practice with game elements today, hence making the most used game element rewards [27] since it can be combined with most elements.

Phillips et al. [42] did a research of existing video game rewards and their different types, providing a summary of them. In their article they mention Wang and Sun [42] and their classification of rewards. They presented a form and mechanic of rewards, describing eight different rewards:

- **Score systems**, user receives some kind of score.
- **Experience points**, user gain experience points.
- **Item granting systems**, user gain items.
- **Resources and valuables**, user receives resources of some kind of valuable.
- ** Achievement systems**, user get an achievement.
- **Feedback messages**, user receives some kind of feedback in message form.
- **Animations and pictures**, user is shown animation or a picture.
- **Unlocking mechanism**, user manage to unlock something that was locked before.
They also presented Hallford and Hallford’s [42] work that described four conceptually different reward types:

- **Rewards of access**, the user gain access to something that has not been accessed before, like a new part of the map for example.

- **Rewards of facility**, enable the user’s avatar to do something that it has not been able to do before. For example the avatar levels up and a new move is available.

- **Rewards of sustenance**, the user maintain avatar’s status and possessions that they have obtained throughout the game. An example of this could be extra lives.

- **Rewards of glory**, the user achieves score or achievements. They are for example hailed as the hero in the game for slaying the dragon.

Even if rewards may be the most common one used today, there are several more game design elements that can be grouped into different categories. Spripada et al. [49] suggested a classification of game elements after a SCV, Several Complex Variables, analysis where they listed down common and variable game elements in a software engineering aspect. They divided the different game elements into three different categories: behavioural, feedback and progression. The game elements of each group can be static or dynamic, as seen in Table 2.1. Static elements do not depend on the current game state and the user input while dynamic elements do. A behavioural game element is more focused on the human behaviour, feedback elements are to improve the motivation and progression elements are more to define state and structure.

<table>
<thead>
<tr>
<th>Behavioural</th>
<th>Feedback</th>
<th>Progression</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Static</strong></td>
<td>Achievements, Rewards</td>
<td>Checklists, Milestones</td>
</tr>
<tr>
<td><strong>Dynamic</strong></td>
<td>Leaderboard, Teams</td>
<td>Activity stream</td>
</tr>
</tbody>
</table>

Table 2.1: Scipada’s et al. result of classified game design elements.

### 2.1.3 The Behavioural Aspect

You could say that game elements is the heart of gamification. However, it is not as simply as putting some game elements together and hope it will motivate the user, there has to be some thought behind it. One aspect that has to be considered, and is essential in gamification, is the user’s motivation when performing a task. Why do they do it? There are a lot of different kinds of people in the world and as a result of that each person is motivated differently.
The Self-Determination Theory describes that an action of a human can be intrinsically motivated or extrinsically motivated [54, 3, 57, 28, 27, 26]. Intrinsically motivation is when a person do something just because it is fun or that they enjoy it while extrinsically motivation is when the person performs a task because of some kind of prize that is received from doing it. That prize could be a reward, grade, achievement or money. The theory also mentions that a human has three psychological needs:

1. **Competence**, the need to feel capable of doing something.
2. **Autonomy**, the need to feel the freedom of choice.
3. **Relatedness**, the need to feel connected to other people.

There are numerous ways to fulfill these needs with gamification. To make the user feel competent, a progress bar or goals could be used. For autonomy it should be possible for the user to make choices and not be forced to do something. To achieve relatedness the user should have some way to interact with other people, like seeing other peoples progress or comments. Minović et al. [36] wrote a theory similar to the Self-Determination Theory but you could say it is a bit gamified. They proposed four types of factors that can act as motivation:

1. **Challenge**, the application should challenge the user but it should not be too hard or too difficult.
2. **Control**, the user should feel that they are in control in the application and that they affect the end result.
3. **Curiosity**, there is something unknown for the user that can be discovered.
4. **Fantasy**, there is an imaginary world that the user can be a part of.

In addition to the basic motivations of a user there are also different kind of personalities to consider. This can be apparent when looking at ordinary games and the people that play them. There are some papers that propose similar kind of personalities in the gamification area [21, 48, 27] and a summarization of them, the different personality type and motivations, can be seen in Table 2.2.

Usually you would want to use game elements that reaches all different groups of personalities to maximize the number of users. However, one element that motivates one person might be demotivating for another person, like for example leaderboards. A solution to this could be to personalize the application [21, 22], which also adds an extra layer of autonomy. The level of personalization in an application can vary. It can be something simple like the user has the choice to remove a game element they do not want, or it can be more advanced like using a Recommender System [56] to personalize the gamified application.
### 2.2 Accelerometer

The accelerometer is a sensor that exists in most mobile phones today. You have most likely encountered it if you have ever used a smart phone. It is for example used when you are taking a picture and switch between landscape or portrait view. The accelerometer is also used in a lot of games for tilting or shaking the device. Figure 2.1 shows the orientation of the different axes for the accelerometer [34].

![Figure 2.1: Picture of the accelerometer’s orientation in a smart phone.](image)

<table>
<thead>
<tr>
<th>Personality type</th>
<th>Motivation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Killer, Dominant, Competitive</td>
<td>Winning and ranking. Compares themselves to other people, motivated by leaderboard and high score. Want to be visible.</td>
</tr>
<tr>
<td>Achiever, Objectivist, Goal setter</td>
<td>Completing goals, getting achievements and finish quests. Want to demonstrate their skills by for example leveling up.</td>
</tr>
<tr>
<td>Socialite, Humanist</td>
<td>Socializing, developing network of friends and contacts. Prefer working with someone else than alone.</td>
</tr>
<tr>
<td>Explorer, Inquisitive</td>
<td>Exploration and discovering the unknown. Feels validation through discovering and investigating new things.</td>
</tr>
</tbody>
</table>

Table 2.2: Summarization of different personality types and their motivations.
2.2.1 Collecting Data

Data that an accelerometer gathers is originally continuous, there is no clear start or end of it. It is however possible to reduce it to discrete data and with that achieve a continuous representation. You could say that it is similar to filming a video. Reality is constantly moving so when filming it you are only capturing a small portion of it. The film itself is just a bunch of pictures put together, taken in such a high rate so the human eye perceives it as a continuous film.

To achieve a discrete representation of the data, the sensor is set with a sampling rate. The sampling rate, \( f_s \), is a chosen frequency and describes how many samples the sensor should collect in a second. So using a sampling rate of 50 Hz means that the sensor will collect 50 samples of data in a second. The sampling period, \( T_s \), is the time interval in which the samples are taken. It can be calculated with the sampling rate as following,

\[
T_s = \frac{1}{f_s}.
\]

So if we have a sampling rate of 50 Hz that will correspond to a sampling period of 20 ms [35].

2.3 Machine Learning

When it comes to machine learning and pattern recognition the search to correlate a certain pattern with a specific activity has resulted in the research field Human Activity Recognition, or Activity Recognition. When a human is for example walking or running, there is for the most part a specific periodic pattern to her movements [4, 50, 58, 20]. An illustration of a person walking with a phone in their hand can be seen in Figure 2.2.

But before we can go any further it is necessary to talk a bit about the concept of classification and preprocessing for better comprehension what activity recognition is and how it works, if you are not familiar with it already.

2.3.1 Classification and Neural Networks

Classification

Classification algorithms can be used when solving problems like activity recognition. A program is trained with known input data, a dataset, under supervision so it later can be used to classify unfamiliar data. You could say that training a program for activity recognition follows this procedure, grossly simplified: Feed the program an input \( X \) value into a model, it compares the result with the "true" output \( Y \), then with a learning algorithm it tries to minimize the error and correct its values accordingly. The program tries to learn the correlation between input data and output value [31].
As a primitive example of classification we can use a simple linear function which is our model, \( y = ax + b \). The constants \( a \) and \( b \) are unknown to the program. For it to learn the values of \( a \) and \( b \) we need to train it with a dataset which is fed to the model. The dataset consists of several input \( x \)-values and output \( y \)-values, calculated with the correct \( a \) and \( b \) constants. Before training, any value can be used as start values of \( a \) and \( b \). To train the model, an iteration through the dataset is performed. The program uses the input value, \( x \), from the dataset, to calculate an \( y \) value with its current model. It then compares the result with the correct \( y \) output value from the dataset for that specific \( x \) input. The program updates its model, trying to minimize the error with a learning algorithm. After a number of iterations the program’s \( a \) and \( b \) values in the model approaches the true value of \( a \) and \( b \). It is then possible to use the model in the program to calculate the \( y \) output for any \( x \) input.

There exist several different types of classification algorithms. The example above is a linear classifier with the model \( y = ax + b \). A few other examples of classification algorithms are decision trees, support vector machines, nearest neighbor and neural networks [47].

**Neural Networks**

Neural Networks is a type of classification and as the name suggests it tries to mimic the behaviour of a brain. The brain consists of neurons that uses electrical or chemical signals for communication, that unconsciously or consciously performs different kinds of classifications. Humans and animals use this for several purposes, to for example differ if something might be dangerous or not which is crucial for survival. A human’s brain performs classifications constantly. We use it to for example recognize that a person is walking or running. The type of input data we use for such a classification could be visual, we see that it is a person moving and notice how
they move their body. Since we probably have seen a person running beforehand we can with that previous experience conclude that the person is indeed running. The movement of a person could also be deduced with for example hearing, depending on how fast and loud the sound of the person’s footsteps are. However, we first need to connect that what we are hearing is indeed footsteps of a person and not something else. We cannot be sure with absolute certainty that what we hear is indeed a person running. We can only perform a prediction of what we are most likely hearing using previous experience.

Due to the brain’s complexity it is necessary to do some simplifications to represent the human way of classifications. It can be done using graphs, weights and binary input. The vertices in the graphs represents neurons and the edges are the connection between them. To represent the signals between neurons binary 0 and 1 are used, there 1 represent a signal and 0 no signal, thus resulting in a neural network [32].

An ordinary neural network do not use previous predictions for the next prediction, it is not persistent. To solve this problem there is something called Recurrent Neural Networks, or RNN for short. If we have the sentence ”This is a sentence”, a standard neural network would only look at each word at the time, not considering how the sentence is put together. A recurrent neural network however would use what has been previously read to for example predict what the next word might be in the sentence. RNN is not flawless though, it has some problems remembering things in the long term, such as longer sentences. Fortunately to the rescue is Long Short Term Memory Networks, or LSTM Networks, ”a special kind of RNN”. LSTM Networks was introduced as early as 1997 and have no problems with the memory in long term, which makes it a fitting classification algorithm for activity recognition [39].

2.3.2 Preprocessing, Features and Labels

Preprocessing

For most part when performing activity recognition the data comes from some kind of sensor, in form of raw data. This usually needs to be processed a bit first before it can be used as input to a classification algorithm, hence the name preprocessing. This preprocessing usually consist of reducing noise and segment the data.

Since sensors have a tendency to be very sensitive to noise, especially the accelerometer, some kind of filter is often used to filter it out. The standard for accelerometer is a low-pass filter that filters out the gravitational acceleration. Segmentation of the data is pretty much what it sounds like, dividing the data into smaller segments/windows of a certain size. It is with these segments features are calculated through feature extraction.

Features

Features are essential if a classification algorithm is going to learn anything. It describes the thing that is to be classified. A good example of features are those that describes flowers, that classifies which type of flower it is. Features of a flower
could be the number of petals and leaves it has, its petal color and the length of its petals, stem or leaves et cetera. The better the features the better a classifier learns. As previously stated, features for activity recognition are extracted from the segments which the data is divided into. These features could be time-domain or frequency-domain features. Example of time-domain features are mean, max or min values of a segment, it could also be the standard deviance or variance. Frequency-domain features could be energy or entropy of a segment [50].

**Labels**

Even if features are essential in classification they would be meaningless if there is nothing for the algorithm to associate them with. It is not enough to have only the petal color of a flower, it is also necessary to link it to which type of flower it belongs to. In relation with activity recognition it means that data from an activity in a dataset should have a label, this could be walking, running, standing or sitting and so on. So, if a segment is input to a neural network model then the output would be a label with a probability rate.

### 2.3.3 TensorFlow

TensorFlow [52] is an open source framework for machine learning developed by the Google Brain Team. The way TensorFlow works without going into too many details, is similar to Lego. You have your pieces which you can put together in any way you want. There are variables, placeholders, sessions and tensors, a description of what each of these do can be seen in Table 2.3 [59].

<table>
<thead>
<tr>
<th>Terminology</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Variables</td>
<td>Used for example to defined biases and weights for the graph. Values can change under run time.</td>
</tr>
<tr>
<td>Placeholders</td>
<td>Used to process input data and to be able to retrieve output from a model.</td>
</tr>
<tr>
<td>Session</td>
<td>This is needed to run any kind of operation. Defines how the program should run.</td>
</tr>
<tr>
<td>Tensor</td>
<td>A typed multi-dimensional array. Used to for example represent images with dimensions.</td>
</tr>
</tbody>
</table>

Table 2.3: Some TensorFlow terminology with description.

To create a model in TensorFlow cells and layers are used. Usually when training a model the dataset is very large so it has to be divided into smaller parts called batches. Epoch is the number of iterations the entire dataset goes through the model [45].
Chapter 3

Method

This chapter describes the methodology of how everything was developed, the stairstep counter, the game elements and the Android application. The work was divided over a total of 20 weeks: 4 weeks went to research, 5 weeks for the stairstep counter, 5 weeks developing the application, 3 weeks writing the report and the last 3 weeks preparing for the presentation of the thesis.

The research was performed by searching after scientific articles in the areas gamification, mobile sensors and activity recognition. A collection of 30+ articles were then read and summarized to identify their most beneficial parts. Of those articles 25 were chosen and presented in the theory section, Chapter 2, where a definition of gamification was given including some game elements and examples of the most common elements. A behavioural aspect of gamification was also discussed together with the Self-Determination Theory and some personality types to describe what motivates people. In addition of the gamification theory some fundamental theory about machine learning was given, how a pattern when a human is moving can be seen with an accelerometer and how classification can be used to recognize that pattern.

The development of the stairstep counter consisted of a significant amount of testing, using different features and labels that in the end resulted in a final counter. Before the application was developed, game elements were derived using information gained from the research part. An application was then created with the stairstep counter using the accelerometer sensor, server and a database.

3.1 Work Process

The work in this thesis was divided into three different parts. The parts and their description can be seen in Table 3.1. Except from the different parts, the work was also run in two weeks shifts, similar to sprints. In the beginning of each sprint a plan of what had to be done during the sprint was planned and in the end of the sprints a summary was created of what had been done and what should be done in the next print and so on.
3.2 Use Case

The use case of the application can be seen in Figure 3.1. The user walks in a stair, holding a phone with the mobile application active. The user will then walk in a stair and have their steps counted. In the application the user should be able to walk stairs with or without animation. They should also be able to join, create and compete in competitions, accept or decline invites and send invites to other users to competitions they have created. They should also be able to create an account and log in.

Table 3.1: The different work parts of the thesis.

<table>
<thead>
<tr>
<th>Part</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Searched after and read articles about gamification, mobile sensors, machine learning and activity recognition. Using different studies possible game elements for the application was thought out.</td>
</tr>
<tr>
<td>2</td>
<td>Performed tests with different datasets, features and window sizes using a LSTM-model for the stairstep counter to see if it was possible to count stairsteps that way.</td>
</tr>
<tr>
<td>3</td>
<td>Implementation. Used derived stairstep counter and game elements to develop an Android application.</td>
</tr>
</tbody>
</table>

Figure 3.1: Use case for the application.
3.3 Technical Choices

3.3.1 Hardware

The first hardware decision that had to be made was what kind of device should be used to count the number of steps a person has taken. After some consideration a smart phone seemed like the preferred choice, since most people own or have access to one. The phone used in this work for development and data collection was a Samsung Galaxy S7.

The next decision that had to be made was which sensor, or sensors, to use. The accelerometer seemed like the most favorable one for the task in mind because it exists in pretty much every smart phone in this day and time. Other sensors, like the gyroscope or barometer, were considered but with battery consumption in mind it was excluded.

3.3.2 Software

When smart phones are involved there is always the decision which platform to use, Android or iOS. Since the only smart phone available was a Samsung phone the obvious choice would be Android. The cross-platform React Native was considered, but it was not picked due to the machine learning framework, TensorFlow, that was not fully supported in React Native. The language picked for the Android application was Java due to personal preference.

TensorFlow seemed as a good choice since it is said to make machine learning a little bit easier for those that are new to it. It also ”takes care” of feature extraction which saves a lot of time since there is no need to get into the mathematics to calculate the features on your own, which is why TensorFlow was chosen. The only drawback is, as previously stated, that it only supports iOS and Android and no cross-platforms yet. This limited possible platforms for development to Android.

With TensorFlow came another decision, what programming language to use when training the model. TensorFlow has the property to freeze and optimize models so they can be used in an iOS or Android application, independently which language was originally used to train them. Python seemed as a good choice since it has the NumPy library which makes the preprocessing easier and most activity recognition examples using TensorFlow on the Internet is written in Python.

For the Android application a server and a database were needed for creating user accounts and storage of application data. To keep it simple a standard HTTP, Hypertext Transfer Protocol, server was used and due to an earlier idea involving training the model on the server, the server and its Restful API, Application Program Interface, was written in Python. For the database MySQL was picked. The reason was previous experience with it and that it would suffice for the task at hand. Both the server and the database was set up on an Amazon EC2 instance with Ubuntu 16.0.4.
3.4 Stairstep Counter

Before an application could be developed there was the need to find a solution on how to count the number of steps a person have walked in a stair. The initial idea was to in some way get the height and length a person had walked and with that estimate, using the average height of a stairstep, the number of steps a person had taken. However, it is not possible to get a very accurate altitude change indoors with a smart phone. There is the barometer sensor but it is highly sensitive to pressure changes and cannot handle the change in climate indoors due to the air conditioner and such [37]. So instead machine learning, or more specifically activity recognition, seemed as a solution to the problem. Google offers an API for activity recognition but activities supported are "in vehicle", "on foot", "running", "walking", "on bicycle" and "still". It does not support up- or downstairs activities, hence leading to the decision to implement custom activity recognition.

The "usual" activity recognition uses labels like downstairs, walking, standing et cetera but in this case it is of more interest to recognizing a single step a person has taken in a stair and all other activities can be overlooked. You could say it is more similar to motion recognition than activity recognition but lets call it activity recognition anyway. Since no existing dataset for these labels was found a decision to create a custom dataset was made.

3.4.1 Dataset

To create a dataset a simple data collecting mobile application was first created for the Samsung Galaxy S7. A screenshot of the application can be seen in Figure 3.2. The application displays the values of the x-axis, y-axis and z-axis and has two input fields to set the filename and what activity is performed, which is used to label collected data. It also has a simple counter at the bottom for easier monitoring of how long data has been collected.

The sensor used in the data collection application was the raw data accelerometer [9], so the gravity should be filtered out. That was done using a low-pass filter to retrieve the force of gravity of each axis which then was subtracted from the data. There does exist a linear accelerometer sensor where gravity is already excluded, but it is not supported by all phones and it also needs calibration before use. To decide the sampling rate, the article "Smartphone Sensor Fusion based Activity Recognition System for Elderly Healthcare" by Fareed [20] and an existing activity recognition dataset by UCI Machine Learning Repository [44] were used as guidelines which resulted in using a sample rate of 50 Hz.

Another thing that had to be considered was the placement of the phone when collecting data. In an article by Heng et al. [24] they tried different positions of the phone and got best result with their activity recognition when the phone was in the hand. This would also be the easiest placement for the collection of data, a simple button click to start or stop the collection. Thus, the decided placement of the phone when collecting data was in the hand.
The data collected also had to be stored in some way and format. An ordinary text file seemed as the preferable choice and in deciding the format of the data a Comma-Separated Values (CSV) format, similar to that of the WISDM Lab [30] dataset, was selected. To keep it simple no new lines was added in the file. Python has a data analysis library called pandas [41] which makes reading a CSV file rather easy. The format of collected data follows

\[ \text{user, activity, timestamp, x-axis, y-axis, z-axis}, \]

where user is an integer and activity is step or other. Step is when a person are walking up- or downstairs and other is every other movement. Timestamp and the values of the axes are pretty self-explanatory. A collected sample for the accelerometer can look something like

\[ 0, \text{step}, 518295237320688, 0.221897, -0.247981, 0.103532, \]

Since the needed amount of data for a model to recognize a step was unknown the activities collected was held in separates files, for greater flexibility. This made it possible to merge different collections of activities in different ways to see if and how it affected the result. To create a dataset for training a model several ”activity
files” were merged together to one big set. For the segmentation to work properly the data had to be merged so not two of the same activity followed each other. So to create a dataset with three activities, two step and one other data, they would have to be merged together in the order step, other, step.

Data was collected from seven different people between the ages 23 to 50 with an even distribution of females and men. The way it was collected was a person performing an activity would press the start button in the application when they began the activity and then the stop button when they were done. To collect step data the person would walk up or down a stair, starting at the top or bottom of the stair and stopping once the entire stair had been climbed. A graph representation of a step activity can be seen in Figure 3.3, where a person walked ten steps in a stair which clearly can been seen in the y- and z-axis. An other activity collected could be a person simply walking or holding the phone in their hands for 10 seconds.

![Graphical representation of collected step data in one session.](image)

### 3.4.2 The Model

With the dataset ready it was time to decide what model to use. Since there was a lack of any previous experience with machine learning, using an already written model seemed like the best option. There are some activity recognition examples and tutorials using TensorFlow and Python on the Internet and after some searching a LSTM model, released under the MIT license by Guillaume Chevalier [6], was found. However, this model was for categorizing six different activities: ”walking”, ”upstairs”, ”downstairs”, ”sitting”, ”standing” and ”laying” and could not be frozen for usage in a mobile application so some modifications were needed.
So before any training could be done it was first necessary to change the number of labels/activities the model should classify. This is defined with variables so it was an easy enough change. Next thing that had to be done was to make it possible to freeze the graph. To do this input and output had to be named and the output had to be converted into a probability, which was done with TensorFlow’s neural networks Softmax function.

The model itself consists of two LSTM cells, stacked on top of each other to achieve deepness to the neural network which resulted in it being two recurrent layers deep. The feature extraction is performed by TensorFlow but it is still necessary to feed the model features that describes the activity. Here the features are data that each axis collected from the accelerometer. If there is data from the \(x\)-axis and \(y\)-axis of the accelerometer there are two features. If there is data from \(x\)-, \(y\)- and \(z\)-axis from two different sensor, like the accelerometer and gyroscope, there are six different features and so on. The model by Chevalier used three features, but this was modified to two features, the \(y\)- and \(z\)-axis from accelerometer. The \(x\)-axis was excluded because, as can be seen in Figure 3.3, it does not generate that distinct step pattern as the other axes.

### 3.4.3 Before Training

There were some more steps that had to be done before the model could be trained. First, all data had to be read and merged together into that one big dataset. This was done using the pandas library. Except from filtering out the gravity, made in the data collection application, some additional preprocessing was performed in form of normalization of each axis. Normalization was performed using following equation

\[
\frac{x - \bar{x}}{s}
\]

where \(\bar{x}\) is the mean value and \(s\) is the standard deviation. After that the data was ready for segmentation. The data that should be segmented were the \(y\)- and \(z\)-axis.

Before a segmentation could be performed it was necessary to decide segment size, how many samples a segment should contain. For this an average time it took to take a stairstep was calculated using collected data from the seven different people. First the average of the total number of samples collected when they walked up- or downstairs was derived, which was 280 samples. With a sample rate of 50 Hz this corresponds to \(\approx 5.6\) seconds. The stair used contained 10 steps, so one step took around 0.56 seconds to take. With the sample rate a sample is taken every 0.02 second, so this results in

\[
\frac{0.56}{0.02} = 27.5\text{ samples}
\]

per stairstep. Used segment size was rounded up to 30 samples, or 0.6 seconds, and seemed to be an appropriate length of a step. With decided segment size and yet again using the pandas library the data was segmented. Labels for each segment were also extracted so each input segment had a corresponding output label. The labels were converted to a binary array representation with the pandas dummies function. It was done in alphabetical order so other label corresponds to \([1, 0]\) and step label is represented by \([0, 1]\). The segments and labels were divided into
different NumPy arrays, segments as an input $X$ array and labels as an output $y$ array. The segment at the first position in output $X$ is the activity, or label, at the first position in the $y$ array, for example $[1, 0]$ which is the activity other.

### 3.4.4 Training the Model

With the modified LSTM model and segmentation of data completed the training of model could begin. To be able to monitor the model how it learns during training, a dataset for testing were created and segmented in the exact same way as the training data. All training segments were divided into batches of size 100 segments and then fed into the model for 10 epochs. During training a test segment was given to the model after each epoch to see what accuracy and loss it currently had. In Figure 3.4 a full training process can be seen where final accuracy is around 83% and loss around 0.74.

![Figure 3.4: Result of training the model with the final dataset.](image)

After the training was done the model had to be frozen and optimized for mobile use. First step was to save the session with the model’s current weights and biases. The model could then be frozen in a `.pb`-file format and optimized so the file size was kept to a minimal. The file could then be used as an asset in the android application. The model was after that tested to see its accuracy and how many stairsteps it counted in a stair. These tests were done manually, walking in a stair with 10 steps. Depending on the result, how many steps the counter counted, the dataset was changed accordingly and a new model was trained with the new dataset, repeating the cycle until a good result was achieved.

### 3.5 The Game Elements

Using the theory from the previous chapter game elements were developed. When creating the elements it was kept in mind that they should not overshadow the stair walking. An already natural game element provided by counting the number of stairsteps was the point element. With points a high score also seemed appropriate.
since it is a good way to motivate a user to improve. Since visualization using animation also can be a good motivator an animation element was decided. A competition element was also decided. From the theory it was also recommended that the elements should be able to be turned off. Hence a setting was developed where a user could turn off the competition element and animation element if desired.

3.5.1 Animation

As mentioned in Section 2.1.2 animations can be seen as a type of reward and having only a counter when walking in a stair is not that engaging, therefore an animation element was added. However, the user’s safety had to be kept in mind. The animation should not be too complex since looking on the phone screen while walking up- or downstairs is not very safe since it increases the risk of the user loosing their balance and fall. Another reason to keep the animation simple was lack of previous animation experience, so a more simple animation would be advantageously in several aspects. So, as a result the animation was created with a round circle and some lines to create an illusion of a ball jumping up a staircase, as seen in Figure 3.5.

![Figure 3.5: Screenshots from application when the ball from the animation element is moving one step up, from left to right.](image)

The movement of the ball was created from scratch. The steps of its development can be seen in Figure 3.6. The initial step was simply getting the ball from A to B, in a rather square way. This was done first decreasing the $y$-coordinate and then increasing the $x$-coordinate. The grid has its origin, the (0, 0) position, in the top left corner so to go down/up $y$-coordinate is increased/decreased and to go right/left the $x$-coordinate is increased/decreased. Next step was to first go up, then change both $x$- and $y$-coordinate to get a diagonal and finally only increasing the $x$-coordinate. To make it look more natural the time the $x$- and $y$-coordinate was increased and decreased was prolonged. Then instead of only increasing the $x$-coordinate in the end, the $y$-coordinate was also increased to make it look like the ball was going down which resulted in the final movement of the ball. Even though the movement might still look a little bit square the ball moves in such speed that it is interpreted by the eye as a curved movement.

The first direction that was implemented was the ball jumping from left to right, like in Figure 3.6. To implement the other direction, moving right to left, the movement in $y$-position was exactly the same and the $x$-position was decreased instead
To decide when the ball should switch direction two different boundaries were declared, one left boundary and one right boundary. The left boundary was 20% of the screen and the right boundary 80% of the screen. The direction was toggled whenever the \( x \)-position reached any of these boundaries. So, while the \( x \)-position had not reached the right boundary the direction of the ball would be from left to right and once it reached the boundary it would switch to right to left instead. This direction would continue until the \( x \)-position reached the left boundary, the direction would then switch again and so on.

Once the ball was able to jump left and right some kind of screen translation had to be implemented. In game development a "scrolling background" is a well known trick to create the illusion that the sprite is moving when it in fact is standing still. Only a portion of the background is shown and it is "scrolled" forward in a certain speed when the player press the forward key, once the player reached the end of the background it is reused giving the impression that the background continues. With this trick as inspiration, a similar scrolling illusion was implemented in a vertical direction. Since the ball is moving up the screen the \( y \)-position will decrease until it has reached a negative state and the ball is not visible anymore. In the Android application the ball and background is drawn on something called Canvas, and this can be translated with \( x \)- and \( y \)-positions. In this case it is only necessary to translate the \( y \)-position. The translation works in that way that it translates the origin, (0, 0) to the new value, (0, \( dy \)), so every position on the screen is then \((x, y + dy)\). Using this the Canvas was translated each time the ball jumped up, keeping it in the center of the screen. To create the illusion of a flowing background four staircases of lines were created and the translation of background and ball were reset to the bottom with an offset once the ball had jumped a certain number of steps, reaching the top of the staircases.

3.5.2 Competition

Shameli et al. [46] performed a study how walking and competitions affects people's activity. Their result was that competitions could be used to reach a broad user base and could be a good way to motivate people if implemented right. What had to be considered was that the people competing against each other are on similar levels or there is a risk of loss in motivation. A competition could also be favourable following the behavioural aspect, as described in Section 2.1.3. It can satisfy the user’s feeling of competence, relatedness and, depending how implemented, autonomy as well. It would also work well with the animation element, therefore a competition element was decided.
3.6. THE SYSTEM

When creating the competition element the first idea was to have one-on-one competition, something similar to Wordfeud [1]. To increase the social aspect and the feeling of autonomy, it was decided to have one-on-many competitions that could be private or public. There is still the possibility for one-on-one competition, which gives the user freedom of choice. To add some additional feeling of being in control the users were given the ability to create competitions and "own" them. To give some personalized feeling to the competitions, beyond the standard settings, the opportunity to name them and choose an image was also added.

With the possibility of private and public competitions came the need of invitations for the private competitions. To add additional feelings of freedom the invites were developed so a user could accept or decline them. Only an owner of a private competition is able to invite a user, which was done with a search function so the owner can search through the database after the user they want to invite using their username. To join a public competition it was implemented so there is no need for invitation, but the owner of the competition can set a limit on how many can join the competition.

3.6 The System

Except from an Android application that runs on the clients, with the different game elements came the demand for a database and a server to store and access the information about users and competitions. As written in the technical choices, the HTTP server were written in Python and the database chosen was a MySQL database. An overall view of the architecture of the system can be seen in Figure 3.7.

![Simplified architecture of the system.](image)

3.6.1 Application

The development of the Android application began with a standard main activity [10] which is the initial activity when a user start the application. Here the application controls if the user is logged in already or not. If not, it redirects the user to the login activity so they can log in. There is also the possibility for a user to create an account. To create an account only a username and password is needed,
no additional information is required to keep it as simple as possible. When the user logs in, there will be a check if it is the first time the user logs in on that device. If it is, then the user will be taken to a configuration activity to set which game elements they want enabled in the application. A flowchart of the entire login procedure can be seen in Figure 3.8.

Figure 3.8: Flowchart of the login procedure when user first start application.

To control in the main activity whether the user is logged in or not and if it is a first time login, the Android Shared Preference [15] was used. Shared Preference stores data locally on the device in a key, value format and will be deleted when a user uninstalls the application. Its operation is rather costly though and can affect performance according to the documentation so it should not be used frequently. Therefore the username for the user is retrieved once in the main activity from the Shared Preference and is then carried on through as extra information in Intents [13] when switching activities. If the user is not logged in yet, the user name will be retrieved from the login activity and then carried to the main activity through Intents. Except from the control if the user is logged in, the Shared Preference is also used for settings of the application. Since settings is not something that is regularly accessed in an application, using the Shared Preferences seemed as an acceptable solution despite its costly operations. The settings was implemented using a JSON, JavaScript Object Notation, object which was converted to a string format before it was stored. For a user to start another activity in the main activity, on click listeners for buttons and some Intents to start another activity were used.

In the stair activity the stairstep counter model and the accelerometer are used. The animation is also implemented here and will show if it is enabled in settings. The animation itself runs on a separate thread than the User Interface (UI) thread, using SurfaceView [12] for better performance. The stairstep counter model, or step model, uses the Java Inference Library [53] provided by TensorFlow to load the model so inputs can be run and outputs received. For the stair activity to know that a step has been registered a step listener was implemented. The step listener use the accelerometer to collect data which is then used as input to the step model once enough samples are collected. The model produces a prediction of activity
performed and the step listener decodes it, if it were a step or other activity, and notifies the stair activity if a step has been taken. When the stair activity receives a step notification it will update the animation, if enabled, and increment a standard counter. An overall view of the dependencies can be seen in Figure 3.9. The step listener follows the "listener" or "observer" pattern [7].

![Figure 3.9: UML diagram of the implementation of step counter activity.](image)

The competition element was implemented rather straightforward, using a competition activity as parent to the rest of the activities and with buttons and Intents to switch between them. The majority of the work was how to display the information from the server in a proper way. To display competitions and invites List Views [14] together with custom adapters and different layouts were used. Some fragments [11] were also used for a tab layout view. Since pretty much all information displayed had to be retrieved from the server, a way to effectively call the server without blocking the main UI thread had to be implemented. This was solved using an asynchronous task together with a callback, as seen in Figure 3.10. The figure shows the dependencies of the create competition activity, how it uses the asynchronous task to make a request to the server with the callback.

![Figure 3.10: UML diagram of the implementation of server request with example activity.](image)
3.6.2 Database

First a simple sketch of the database was drawn to establish the relationship between the tables and how it should be built. A table called competitions and one table called users seemed as a good start. The tables participants and invites were also added. An EER diagram of created database can be seen in Figure 3.11. An auto incremented integer as id resulted in being the primary key of the competitions table. In the users table a username was chosen as primary key since they should be unique so using an id as well did not seem necessary. Since a user should be able to be an owner to a competition the primary key of the users table was used as foreign key in the competitions table. Next step was how to keep track of the participants of a competition. This was done using a participants table with competition id and username as foreign keys and primary keys. The reason for this was because a user should be able to participate in several different competitions and a competition should be able to have several participants but the combination of certain a competition and a user should not occur twice. The same way of thinking was applied when creating the final table, invites, that uses three foreign keys which also are primary keys. The password stored in the database for the user’s account is hashed with a salt, encrypted on the server before a new row was inserted into the database.

![EER diagram of MySQL database](image)

Figure 3.11: EER diagram of the MySQL database.

Because competitions have start and end dates some kind of scheduled event to update a competition status depending on the current date had to be created. A status of a competition is represented with an integer, as shown in Table 3.2. This was done using an event in the database that runs every day at around two o’clock.
in the morning. The event retrieves current date and set status to 1 if the start date has passed and set status to 2 if the end date has passed. It also removes all competitions with end dates that is older than one week. All competitions are created with a status of 0 and it is not possible create a competition with a start date of current day, it has to start at least one day later.

<table>
<thead>
<tr>
<th>Status</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Pending, the start date has not yet passed.</td>
</tr>
<tr>
<td>1</td>
<td>Ongoing, start date has passed and the competition has begun.</td>
</tr>
<tr>
<td>2</td>
<td>Finished, end date has passed and the competition has ended.</td>
</tr>
</tbody>
</table>

Table 3.2: The different kind of statuses a competition can have.

### 3.6.3 Server

The server’s purpose is to process and fetch information from the database depending on a client’s request. Due to time constraint there was a need to have a server up and running quickly, therefore the http.server [16] library from Python was chosen. Using the HTTPServer and BaseHTTPRequestHandler classes the different HTTP methods, GET, POST, PUT and DELETE, was handled accordingly. Because previous experience with Rest API it was chosen as means of communication between the client and server. The API path format was decided to begin with the route /api/ and then what table the client wanted to retrieve information from, for example participants or competitions. There are some exceptions like the login and search paths. To send data with the request the q={} format was chosen since it could easily be parsed and retrieved using Python’s urllib.parse [17] library. A “systemd” service for the Python script was also set up [29]. The different API routes can be seen in Table 3.3.

The encryption of a user’s password that is stored in the database is performed on the server. The encryption uses Python’s hashlib [18] and the pbkdf2 hmac function to generate a hashed version of the password. The function requires a salt, which should be randomized and a minimum size of sixteen bytes, and a hash digest algorithm. Salt used for the hash is randomly generated for each user, so two users should not have the same salt. As for the digest algorithm, the SHA-256, as suggested in the documentation, was used.
<table>
<thead>
<tr>
<th>API path</th>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>/api/participants?q={&quot;score&quot;: &quot;&quot;, &quot;user&quot;: &quot;&quot;}</td>
<td>PUT</td>
<td>Updates the score of a user.</td>
</tr>
<tr>
<td>/api/participants?q={&quot;user&quot;: &quot;&quot;}</td>
<td>GET</td>
<td>Used to get active competitions for a user.</td>
</tr>
<tr>
<td>/api/participants?q={&quot;comp&quot;: &quot;&quot;, &quot;user&quot;: &quot;&quot;}</td>
<td>POST</td>
<td>Used when a user joins a competition. The &quot;comp&quot;: field is the ID of competition.</td>
</tr>
<tr>
<td>/api/participants?q={&quot;comp&quot;: &quot;&quot;}</td>
<td>GET</td>
<td>Retrieve all participants for a competition.</td>
</tr>
<tr>
<td>/api/login?q={&quot;username&quot;: &quot;&quot;, &quot;pass&quot;: &quot;&quot;}</td>
<td>GET</td>
<td>Used to validate login for a user.</td>
</tr>
<tr>
<td>/api/login?q={&quot;username&quot;: &quot;&quot;, &quot;pass&quot;: &quot;&quot;}</td>
<td>POST</td>
<td>Creates a new user.</td>
</tr>
<tr>
<td>/api/competitions?q={&quot;user&quot;: &quot;&quot;}</td>
<td>GET</td>
<td>Retrieve all competitions a user can join.</td>
</tr>
<tr>
<td>/api/competitions?q={&quot;name&quot;: &quot;&quot;, &quot;public&quot;: &quot;&quot;, &quot;startDate&quot;: &quot;&quot;, &quot;endDate&quot;: &quot;&quot;, &quot;owner&quot;: &quot;&quot;, &quot;type&quot;: &quot;&quot;, &quot;image&quot;: &quot;&quot;}</td>
<td>POST</td>
<td>Creates a new competition without a contestant limit. Add field &quot;limit&quot;: to add a contestant limit. Start and end date is in DATE format.</td>
</tr>
<tr>
<td>/api/competitions?q={&quot;ID&quot;: &quot;&quot;}</td>
<td>GET</td>
<td>Retrieve all information about a competition.</td>
</tr>
<tr>
<td>/api/competitions?q={&quot;owner&quot;: &quot;&quot;}</td>
<td>GET</td>
<td>Get all competitions where a user is owner.</td>
</tr>
<tr>
<td>/api/invites?q={&quot;receiver&quot;: &quot;&quot;}</td>
<td>GET</td>
<td>Used to get all invites for a user.</td>
</tr>
<tr>
<td>/api/invites?q={&quot;comp&quot;: &quot;&quot;, &quot;sender&quot;: &quot;&quot;, &quot;receiver&quot;: &quot;&quot;}</td>
<td>PUT</td>
<td>A user accepts an invite.</td>
</tr>
<tr>
<td>/api/invites?q={&quot;comp&quot;: &quot;&quot;, &quot;sender&quot;: &quot;&quot;, &quot;receiver&quot;: &quot;&quot;}</td>
<td>DELETE</td>
<td>A user declines an invite.</td>
</tr>
<tr>
<td>/api/invites?q={&quot;comp&quot;: &quot;&quot;, &quot;sender&quot;: &quot;&quot;}</td>
<td>GET</td>
<td>Retrieve invites a user has sent for a competition.</td>
</tr>
<tr>
<td>/api/invites?q={&quot;comp&quot;: &quot;&quot;, &quot;sender&quot;: &quot;&quot;, &quot;receiver&quot;: &quot;&quot;}</td>
<td>POST</td>
<td>Creates an invite from a user to another user.</td>
</tr>
</tbody>
</table>
3.7 Technical Result

To answer the questions asked in this thesis, stated in the introduction, the first problem that had to be solved was how a stairstep could be counted. After some research and reading articles, machine learning and the machine learning framework TensorFlow came into the picture as a possible solution. But first the decision of what sensor(s) and dataset to use had to be made. The hardware choices depended heavily on what was available. Since a mobile phone is something most people have it resulted in that an Android smart phone and the accelerometer sensor was used for development.

When it came to what dataset to use for the machine learning the final decision was to create a custom dataset. With a custom dataset also came the need to examine the amount of data needed for best result using machine learning in this context. A model also had to be implemented for the machine learning. After some searching a Neural Network LSTM-model released under the MIT license was found and after some modification used to create the stairstep counter.

To solve the questions regarding gamification several articles were read and evaluated. The game elements were then chosen depending on the context and how they would work together. The amount of time for development when deciding the game elements was also taken into consideration. This resulted in two major game elements, an animation element and a competition element, as well as some minor elements like high score. To keep it simple the animation element was created using a circle and some lines. The competition element was created so a user could create, join and compete in competitions, they could also search and send invites to other users.

A resulting prototype application was developed with above game elements and a stairstep counter model that classifies if a stairstep was taken or not with a rather high accuracy. A user can create and log in to their account, keeping track of what competitions they are active in, view score and use the animation of a jumping ball when walking in a stair. Developed application might perhaps be the first application using machine learning to count a stairstep with TensorFlow, since nothing similar has been found at this moment. There is also no known application on the market today that applies chosen game element the same way and in this context.
3.7.1 Final Stairstep Counter

The result of the stairstep counter, tested on four different people can be seen in Table 3.4. Final dataset used consists of 670 samples of data, taken from seven different people from the ages 23 to 50. The accuracy of the model during training using a training dataset did not reflect the "true accuracy" of the model when actually used in a stair walking so it was more of a guideline. Hence testing the model had to be done manually, walking in a stair to control how many steps it counted. Each person walked up and down a few times in a stair with a total of 10 steps. So if steps taken are 40 the person walked up, down, up and then finally down a stair. The result of when Person 2 walked in the result is a second attempt testing the counter.

<table>
<thead>
<tr>
<th>Person</th>
<th>Stairsteps taken</th>
<th>Stairsteps counted</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>40</td>
<td>40/40</td>
</tr>
<tr>
<td>1</td>
<td>40</td>
<td>40/40</td>
</tr>
<tr>
<td>2</td>
<td>20</td>
<td>20/20</td>
</tr>
<tr>
<td>3</td>
<td>40</td>
<td>40/40</td>
</tr>
</tbody>
</table>

Table 3.4: Test results of trained model using the final dataset.

The result from the test in Table 3.4 shows an accuracy of 100%, however when continuously testing the counter with the same person it did count the correct number of steps for most part but sometimes it could count one or two steps more or less. So you could say that the accuracy overall is between 85% to 100%.

3.7.2 Final Application

The final result is a working application where a user can compete in competitions and use animation to walk in a stair that uses a stairstep counter. As seen in Figure 3.12, when a user start the application the "Home" screen or the login activity will be visible depending on if the user is already logged in or not. At "Home" the user can choose to start walking in a stair or to enter competition element, there are also the possibilities to view profile, change settings and log out.

Figure 3.13 shows the stair activity with the animation element enabled and the main view of the competition element. The animation element has the ball that jumps up on what is supposed to be stairs, displaying a counter that count each stairstep a user has taken. When the user has finished walking they can press the done button and will be directed back to the "Home" screen. If they are competing in any competitions their score will be updated. The main view of the competition element, Figure 3.13b, is the first thing the user will see after pressing the competition button. Here the user can see their "active" competitions, any competitions the user is currently participating in. As an overview of an active competition, information displayed is its name, the owner, its status and what score the user presently has.
3.7. TECHNICAL RESULT

CHAPTER 3. METHOD

(a) The login activity.

(b) The main activity.

Figure 3.12: The first view a user will see depending if they are already logged in or not.

For easier navigation between the active competitions and any invites a user might have from other users a "tab-layout" is used. The user can switch to see the invites from the active competitions by swiping left or by pressing the invites text. The view of the invites and be seen in Figure 3.14a. In a similar manner as the view of active competitions, information displayed for an invite is name of competition, who sent it, competition type and its start date. If the user want to accept or decline the invite, they simply press the invite and a dialog as seen in Figure 3.14b pops up where the user can choose to decline or accept the invite. By swiping right the user can come back to the active competition view. If the user want to have more detailed information about a competition, they can press it and an information view as in Figure 3.15a will show.

Except from viewing the active competitions and invites, a user can go to the join activity, seen in Figure 3.15, by pressing the "join" button in the active competitions view. In the join activity it is possible for the user to search after a competition to join using its name. A user can also create competitions. Pressing the "owned" button in active competitions will take the user to all competitions they are owner of, see Figure 3.16a. There it is possible for the user to create a competition. If they create a private competition the user will be able to choose to invite a user directly or later, as in Figure 3.16b. If they decide to press "now" they will be directed to the invite activity, seen in Figure 3.16c, there they can search for a user by their username and send an invite.
3.7. TECHNICAL RESULT

(a) Stair activity with animation element.

(b) The competition activity.

Figure 3.13: The main view of the two major game elements.

(a) The view for a user’s invites.

(b) User can accept or decline invite.

Figure 3.14: How invitations for a user are displayed and accepted/declined.
3.7. TECHNICAL RESULT

CHAPTER 3. METHOD

(a) Displayed information for competition in (b) Display of possible competitions to join active view.

Figure 3.15: How more detailed information about competition is displayed and view of possible competitions to join.

(a) All competitions owned (b) Private competition created.

Figure 3.16: A user can view, create competitions and invite users to private competitions.
Chapter 4

Evaluation

The questions that was asked for this thesis were

- How can a stairstep be counted?
- What game elements are best in this context to induce a behavioral change?
- What requirements are there to a gamified system?
- What techniques are most suitable for the system?

To solve how to count a stairstep, a stairstep counter was developed. The counter was developed using machine learning, or more specifically a LSTM network, with a custom dataset collected using an accelerometer. The counter was a rather crucial part of the application since without it the application itself would be rather meaningless. The resulting accuracy of the stairstep counter is quite high, but more testing with additional people should be done.

The game elements used in the developed application were derived after reading several studies that had previously been done about gamification. Regarding requirements of the game elements, it was discovered through the studies that the context should not be overshadowed by the game elements so the context should still be the main focus of the application. It was also found that different game elements attracts different people depending on their personality type and what game elements attracts which personality. The used game elements were based on this research. The animation element can be viewed as a reward, and then combined with the counter, a kind of visual progress that could possibly engage the ”achiever, objectivist and goal setter”. The competition element is fairly obvious which personality type it engages, the competitive one. By providing the possibility with private competitions and invites to other users it also engages the more social personality type.

To further evaluate the game elements in the application a questionnaire was sent out with a screenshot of each of the game elements, asking the participants on a scale from one to ten how engaged the image made them feel.
The questions were asked this way because the first impression a user has is important. In the app store a user’s decision to download an app or not depends mainly on screenshots and the description of the application. The ideal would of course be if they actually interacted with the application but sometimes it is just not possible.

Over 300, or 336 to be exact, people participated in the questionnaire. The people chosen to the questionnaire were students at Luleå University of Technology. The students questioned were not students in any specific area, they could study everything from the teacher program to the economy program. Why specifically they were chosen was because the easy access to their email which could be retrieved through the university by being a student. It was also desired to get responses from a larger amount of people for a more accurate result. The questions that were asked in the questionnaire can be seen in appendix A. For a better overview of the result the responses from the questions regarding the game elements, with a scale from one to ten, were arranged into three different groups. The answers from one to four were labeled as ”not that or not engaged”, the value five was labeled ”neither” and the range from six to ten was labeled ”somewhat or very engaged”.

The response from the participants on the question about their feel of engagement concerning the animation element can be seen in Figure 4.1. The first column is people that answered they felt ”not that or not engaged”, second column is ”neither” and third column is of people that felt ”somewhat or very engaged” when seeing the animation element. With a 336 participants the result gives us that 29% has answered that they felt ”somewhat or very engaged” by the animation element and approximately 15% responded ”neither”.

![Figure 4.1: The response from the question regarding the animation element.](image)
CHAPTER 4. EVALUATION

What the participants answered regarding the competition element can be seen in Figure 4.2. The same labels for the columns are used here as with the animation element. The result of the responses was that around 37% of the participants answered that the competition element made them feel "somewhat or very engaged" and 14% that it made them feel "neither".

![Figure 4.2: The response from the question regarding the competition element.](image)

Another question that was asked was how the participants motivation were affected by high score. What they answered on that question can be seen in Figure 4.3. From the result we can see that almost 51% of the participants felt that high score made them more motivated. Approximately 47% stated that high score did not affect their motivation in any way and only 3% responded that it made them less motivated.

![Figure 4.3: The response from the question regarding the high score element.](image)
What was also asked in the questionnaire was how important it was for the participants that a stairstep counter is accurate when counting stairstep. The result from that question can be seen in Figure 4.4. Since the question was asked using a scale from one to ten it is also here possible to group different ranges together for a better overview of the result. Lets say that the range 7 to 10 is labeled as ”rather or very important”. This gives us the result using responses from 336 people that approximately 90% of the participants considered the accuracy of a stairstep counter to be ”rather or very important” to them. Using this result you could say that a requirement for a gamified system is that the technology used in the application works properly.

![Figure 4.4: The response from the question regarding the stairstep counter.](image)

From the questionnaire it is possible to draw the conclusion that the high score element is indeed something that is beneficial to apply in a gamified application. Since around half of the participants were motivated by it, the other half were not affected by it and only a small percent were not motivated by it there is no drawback in using it. The animation and competition were engaging for around a third of the participants, give or take, which was to be expected since they could not interact with the application. Especially an animation is difficult to convey in a proper manner using only screenshots.
Chapter 5

Conclusion

5.1 Future Work

There is always more that can be implemented in these kind of applications. For example the current application is not that scalable, only one server is used and there is no load balancing. To make the application more scalable in that aspect there for example could be more servers and then a load balancing method, like round-robin to distribute clients that want to access it. The application also only supports Android for the moment so developing an application in iOS is something that could be done in the future.

The stairstep counter shows potential, but there is still a lot that could be done with it. For example more testing should be performed on the current counter to determine how accurate it really is, even if the current result is promising. Testing here means trying it on more people and see how many stairsteps it counts. Depending on the result of that there could be a need to train a model specifically for a user during a configuration phase which could be done on the server. If however the result is good then there should be enough to increase the dataset and use some more variety in other data since the dataset at the moment is rather small.

One thing that could be done with the competition element is to add many-on-many competition, so there is the possibility of group competition instead of only the one-on-many that is currently used. This could improve the social part of the application and motivate more users. To improve the animation element it could be something more intriguing, like animation when the score changes and better animation when the ball jumps. To make it more natural a little bounce when the ball lands should be added. The application should also be tested on for example a focus-group for better feedback on its gamification. Unit tests should also be implemented.

Additional competition element that could be added to the application is something like a map with levels, similar to Candy Crush Saga or AlphaBetty Saga. Each level could have goals that a user has to finish before they can advance to the next level. The goals are something that the user can decide if they want to create themselves or use pre-made ones. This would draw in the users that are not that competitive in their nature.
5.2 Ethics

Since gamification is about manipulating a person to a behavioural change there are some ethics that has to be discussed. There is no clear answer though when it comes to ethics since everyone have their own morals. One person might think one thing is fine while another might disagree completely. There are however some guidelines you can follow. Andrzej Marczewski [33] proposed a framework that can be used to control the moral of a gamified application. The suggested framework is:

1. Does the system offer a choice?

2. What is the intention of the designer?

3. What are the potential positive and negative outcome for being in the system?

4. Are the beneficial outcomes weighted toward the needs or desires of the user or the designer?

If we begin with question number one, does the system offer a choice? Yes, you could say that it does. There is the possibility for the user to turn off the game elements if they want to and there is no need for the user to use the application if they do not want to. There is however, as the article mention, risk that the users feel compelled to join the system in ”fear of missing out” or that they feel socially compelled to join the system. This might induce extra stress for people, that they feel they have to exercise more. By using anonymous accounts in the application this reduces the risk slightly since people do not have to use their real name, but there is only so much that can be done and there will always be a risk.

Next question, what is the intention of the designer? For this application it is simply to enhance a person’s experience when walking in a stair to motivate them to increase their daily exercise. Even if this is the intention of the application there is always the risk that users will use it in other ways. There have been some cases where the developers for a system had something different in mind than what the people used their system for. Regarding this application it is hard to imagine any other situation it could work in but it is probably something that could be revealed through people using the system. This also applies to the third question, what are the potential positive and negative outcome for being in the system? Except from what has previously been discussed, that of being socially compelling, there is a risk of people hurting themselves using the application. There is always a danger of for example falling when any kind of movement is involved and especially with stairs. To prevent this there should probably be some kind of warning text before a person start walking in a stair, to be careful and think of safety first.

The final question, are the beneficial outcomes weighted toward the needs or desires of the user of the designer? As the developer you could be rather biased and can only see the application as good for people. However, in this case the system is a bit more beneficial for the developer, since it is a thesis work.
5.3 Discussion

Have I answered all the questions asked for this thesis? If we begin with the first question, how can a stairstep be counted? The development of a stairstep counter has been described in the method section and also its accuracy when tested on four different people, which can be seen in Section 3.7.1. Regarding the accuracy of the counter, from the tests that was made with it a 100% accuracy was achieved, but this is when a person is walking in the stair in a certain speed. Because the window size is 30 samples, or 0.6 seconds, if the person walks faster or slower there is a risk that the counter miss counting a step. To fix this the model just have to be trained with another window size.

Since this was a first experience with machine learning a significant amount of time went to just understanding how it works and how it could be implemented in this context. The stairstep counter alone was something that easily could have been the entire thesis time-wise. A lot of time went to collecting data, testing different datasets and combinations of datasets in an attempt to gain high accuracy. To compensate for this the accuracy of the stairstep counter was limited, since there had to be time to develop the application. It does what it was supposed to do but for the moment you could say that it is a glorified step counter because it can count steps while you are walking or shaking the phone in the hand as well. To fix this more data for the dataset is required, like data from a person walking to set as other so the model learns the difference between a walk and a step. The size of final dataset is very small compared to those activity recognition dataset you can find on the Internet, for example the dataset from the WISDM Lab has 1 098 204 samples of data. Although those datasets are usually for five to six different activities so some difference in size is expected. Anyway, the purpose was to examine how a stairstep could be counted and this stairstep counter does count the stairsteps in a stair rather well so I would say that the question is answered.

What game elements are best in this context to induce a behavioral change? When it comes to the gamification of the application it is difficult to know if the game elements picked were the best for this particular context since more testing is required. For the moment the game elements are mostly based on previous studies and even though the research answered what the requirements for a gamified system are, to know if game elements chosen are the best they should be tested with actual users and see their response. Although the questionnaire gave a good indication how engaging the elements are to different people a focus-group might have provided better feedback since it would be possible for people to interact with the application. So this question would need additional information before a more developed answer can be concluded.

Now to discuss the application itself. The application might feel a bit thin. The primary reason of this was that more time than expected had to be designated to the stairstep counter before an acceptable result was achieved. The amount of time left for the development of the application after that landed on around five weeks and since this was my first larger Android application the development of the application did not go as fast as desired. Even though the application could use more
game elements those that are implemented works. A user can join a competition and compete using the stairstep counter. The animation element works rather well with the stairstep counter that is like a point system, there the stairsteps are the points. Both the competition and animation element are something that has been used before in application but they have not been implemented in this way using the walking in a stair context.

5.4 Conclusion

It is finally time for the conclusion of this thesis. The result of this work was a working prototype application which it is not perfect and a lot more can be done. A few examples was said in the future work section, Section 5.1 and how it can be corrected in the discussion, Section 5.3. Still, the resulting application uses machine learning with TensorFlow in a way that has not been done before, to my knowledge. It is a gamified application to motivate people to walk in stairs using animation or competitions with a stairstep counter classifier, that is not the standard step counter health application that can be found on today’s market.
Bibliography


Appendix A

Questions Asked in Questionnaire

The questions were initially asked in Swedish so a translation to English has been made.

1. Do you want to increase your daily exercise?
2. Do you usually take the stairs instead of the elevator?
3. If no or sometimes, what would make you take the stairs more instead of the elevator?
4. Do you think a mobile application with game elements would make you walk more in the stairs? With game elements means ex points, animation or competitions.
5. On a scale 1-10, how engaging does the picture below feel?
6. On a scale 1-10, how engaging does the picture below feel?

7. How is your motivation affected by "high score"?

8. If you would use an application with a stairstep counter, how important on a scale from 1-10 is it for you that it counts the accurate number of staisteps?