

<b>1 INTRODUCTION .....</b>	<b>2</b>
1.1 Objectives .....	2
1.2 Work method .....	2
<b>2 THEORETICAL BACKGROUND.....</b>	<b>4</b>
2.1 Development methodologies.....	4
2.1.1 LUCID .....	5
2.1.2 “School methodology” .....	6
2.2 The users.....	7
2.3 Design rules .....	8
2.3.1 “Eight Golden Rules of Interface design” .....	8
2.3.2 Layout.....	9
2.3.3 Icons.....	10
2.3.4 Colour .....	10
2.4 Usability testing.....	10
2.4.1 Usability testing methods.....	10
2.4.2 Testing the design with users.....	12
<b>3 TECHNOLOGICAL BACKGROUND.....</b>	<b>14</b>
3.1 Melody composers in Ericsson mobile telephones.....	14
3.2 Melody composers in mobile telephones of other brands .....	15
3.3 The iMelody format.....	18
3.4 The limitations of a mobile telephone .....	18
<b>4 THE DEVELOPMENT OF THE MELODY COMPOSER.....</b>	<b>20</b>
4.1 Vision.....	20
4.2 User definition .....	20
4.3 Design .....	21
4.3.1 The design of the application’s services .....	22
4.3.2 Conceptual design .....	22
4.3.3 The design of interaction.....	24
4.3.4 Graphical design.....	25
4.4 Implementation.....	26
4.5 Usability evaluation.....	27
4.5.1 Heuristic evaluation.....	27
4.5.2 Empirical testing.....	30
<b>5 CONCLUSIONS.....</b>	<b>37</b>
<b>GLOSSARY .....</b>	<b>39</b>
<b>REFERENCES .....</b>	<b>40</b>
<b>APPENDIX A.....</b>	<b>41</b>
<b>APPENDIX B.....</b>	<b>43</b>
<b>APPENDIX C.....</b>	<b>48</b>
<b>APPENDIX D.....</b>	<b>49</b>

# 1 Introduction

The mobile telephones of today are no longer devices that are used only for communication as the old fashioned telephones. Their functionality is rapidly moving towards the functionality of the PDA's (personal digital assistant) with applications as calculator, calendar, games and so on. To have a melody composer in the mobile telephone has become of great importance since it is very popular among the users to have a personal melody as the ring signal.

The big boom of mobile telephone users and the increasing amount of applications in the mobile telephones have increased the importance of a good user interface and a good human-computer interaction. Lack of knowledge of designing a user interface and a human-computer interaction among programmers often result in insufficiency in these areas when developing an application. This is a problem that a lot of companies have taken notice of and therefore started special departments with employees educated in this area.

Implementing applications on a mobile telephone is particularly difficult because of all the limitations. The complexity of an application is often determined by the processor and the size of the memory, which are very limited in a mobile telephone. A complex application that exceeds the processing power becomes very slow and causes feelings of irritation and disapproval towards the application. The small display size and the fixed number of keys also set great limits, especially when concerning the usability of an application.

The first melody composers in mobile telephones simply consisted of a text string with a more or less difficult syntax of letters, digits and a couple of symbols like "+" and "#". Many of these melody composers are still in use but are gradually being replaced with graphical user interfaces to supply the users with a more usable and appealing application.

## 1.1 Objectives

This Master thesis concerns the design, implementation and evaluation of an improved melody composer and its user interface and human-computer interaction in a mobile telephone. The improved melody composer should, among other things, consist of a graphical piano with key and sound reproduction to decrease the memory load and to give a nicer appearance to the application. The Master thesis was performed at Ericsson Mobile Communications AB in co-operation with the Department of Design Sciences at Lund Institute of Technology, Lund University.

## 1.2 Work method

The first period was used for preliminary study about how to design a user interface by reading the book "Designing the User Interface" by Shneiderman [1] and attending the course "Human-computer interaction" at Lund Institute of Technology, Lund University.

Some initial study was also made of the architecture and implementation of mobile telephones at Ericsson Mobile Communications AB. The work continued with a study of melody composers in different mobile telephones.

The development method used in a project during the attended course was adopted. This work consisted of the following stages:

1. Vision.

The initial idea of the application and its purpose is described.

2. User definition

Gathering of information about the users and conclusions about what they want.

3. Design

4. Implementation

5. Usability evaluation

It is an evaluation of the application. This can be done by user tests or by following some rules of user interface design like “The Eight Golden Rules”.

## 2 Theoretical Background

According to Löwgren [2], human-computer interaction can be defined as “a discipline concerned with the development of interactive systems for human use and with the study of major phenomena surrounding them”. The “major phenomena” are some of the complex things like the users, the development methodologies, the design rules and usability. Some of these subjects are shortly described here.

### 2.1 Development methodologies

During the years there have been a development of how to make human-computer interaction part of system’s development. Mainly four different approaches have arisen that roughly can be classified according to how much actual involvement and power the users get during the process. By reading the fig. 1 from the bottom to the top, the historical development of this field can also be seen.

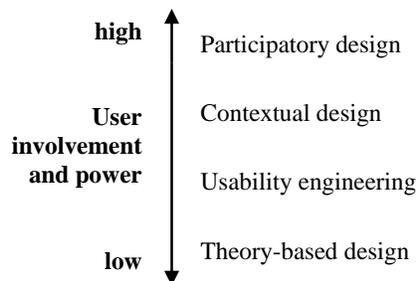


Figure 1. The historical development of the human-computer interaction in system’s development.

In the *theory-based design*, psychology played a major role. Scientific experiments were done to verify the usability of interactive systems as or after they were developed. General descriptions or theories of human’s interacting with computers were also created. All this resulted in analytical models or as general guidelines to help the developers in designing usable systems.

The basic idea of the *usability engineering* is to start with an analysis of the intended users, their tasks and their needs. Based on the analysis, the developers and the users formulate a set of usability goals for the new system. The goals have to be measurable by means of usability testing.

After the usability engineering an alternative approach called *contextual design* was formulated. The main goal with the contextual design is to use more of the users’ context and experience directly instead of making a formal analysis. The design also includes spending time in the users’ environment, looking at their work, discussing it with them

and trying to get as broad an understanding as possible of the whole work context. The new system is designed together with and tested by the users.

*Participatory design* is best described as a process of mutual learning, where the developers learn about the users' world while the users learn about what information technology could do for them and their work. Developers and users design a new work system together, in which technology might play only a minor role.

### 2.1.1 LUCID

A well-known methodology is the Logical User-Centered Interactive Methodology [1] more known as the LUCID that identifies six stages:

1. Develop product concept
2. Perform research and needs analysis
3. Design concepts and key-screen prototype
4. Do iterative design and refinement
5. Implement software
6. Provide rollout support

The more detailed description of the LUCID can be viewed in table 1. LUCID uses participatory design to solicit user input, construct workflow scenarios and define the objects that are central for the design.

A distinctive aspect of LUCID is its focus on the key-screen prototype that incorporates the major navigational paths of the system. The key-screen prototype is used to show users the design of the proposed system and allow them to evaluate and refine it. This is also called Lo-Fi prototyping.

The thoroughness of LUCID comes from its broad use in a lot of projects where the possibility has been given to improve it. Each project has special needs to which a development methodology can be adjusted.

**1. Develop product concept**

Create a high concept  
Establish business objectives.  
Set up the usability design team.  
Identify the user population.  
Identify technical and environmental issues.  
Produce a staffing plan, schedule, and budget.

**2. Perform research and needs analysis**

Partition the user population into homogeneous segments.  
Break job activities into task units.  
Conduct needs analysis through construction of scenarios and participatory design.  
Sketch the process flow for sequences of tasks.  
Identify major objects and structures which will be used in the software interface.  
Research and resolve technical issues and other constraints.

**3. Design concepts and key-screen prototype**

Create specific usability objectives based on user needs.  
Initiate the guidelines and style guide.  
Select a navigational model and a design metaphor.  
Identify the set of key screens using a rapid prototyping tool.  
Conduct initial reviews and usability tests.

**4. Do iterative design and refinement**

Expand key-screen prototype into full system.  
Conduct heuristic and expert reviews.  
Conduct full-scale usability tests.  
Deliver prototype and specification.

**5. Implement software**

Develop standard practices.  
Manage late stage change.  
Develop online help, documentation and tutorials.

**6. Provide rollout support**

Provide training and assistance.  
Perform logging, evaluation, and maintenance.

---

Table 1. The detailed process of the LUCID.

### 2.1.2 “School methodology”

The “school methodology” [3] is a method suited for quite small projects is used in the course “Human-computer interaction” given at Lund Institute of Technology, Lund University. The method can be considered as an reduction of the LUCID and consists of the following five steps:

1. Vision
2. User definition
3. Design
4. Implementation
5. Usability evaluation

The *vision* is a statement that defines the fundamental idea of the project and its purpose. Things like the product's functionality, variability and interactivity should be mentioned. Even the users' needs and symbolic values should be defined.

Methods for retrieving a vision can be to practice at the users' workplace, brainstorming or an investigation of similar existing applications for ideas.

A *user definition* should initially describe all different possible users and their needs. Then the group of users should be defined more specifically. A usability goal in the terms of REAL [2] (Relevance, Efficiency, Attitude and Learnability) should also be described. The relevance of a system is how well it serves the users' needs and the efficiency states how efficiently the users can carry out their tasks using the system. Attitude is the users' subjective feelings towards the system while the learnability of a system is how easy it is to learn for initial use and how well the users remember the skills over time.

A user definition can be made by practising at the users' workplace, observing the users or by small interviews.

The *design* process is divided into four stages: the design of the application's services, conceptual design, design of interaction and graphical design.

The design of the application's services mainly says what services the application will offer and what services it will not offer. By knowing what services will not be offered it is easier to concentrate on the more important services.

The application's overall structure and design is decided in the phase of conceptual design. Here are also the services grouped into their natural groups depending on their functionality and information. These tasks are solved with the help of some questions like, what services belong together and in what order will the services be used. Possible metaphors and analogies are also decided.

In the design of interaction the way of navigation, the appearance of the menus and error handling is designed.

In the final stage, the graphical design, the last fine adjustments are done. Things like layout and grid, colours, icons and sounds are handled.

An *evaluation* of the design can be made by following, for example, the eight golden rules (a so-called heuristic evaluation) or by user tests (empirical testing). More about these techniques can be read in chapter 2.4.

## 2.2 The users

The users are a very important but difficult factor when evaluating usability. They are for whom the system will be made but they are also a group of individuals with different personalities, backgrounds and social norms privately and in their workplace. This is why it is important to study the group of users and their workplace before designing the application. Even though, psychologists have managed to find out some things about basic cognitive processes.

Memory is usually described as a combination of working memory and long-term memory. The working memory has a verbal and visual component and a very limited

capacity. The contents must be refreshed or else they will be forgotten. Long-term memory, on the other hand, has room to store a practically unlimited amount of information. The hard thing is finding it again. It is easier to remember things that have been generated by oneself. Further more, are pictures easier to remember than words. Human memory is amazingly good at recognising pictures that have been seen before. Recognition is generally easier than recollection.

The visual perception is highly used when using systems of today. Humans are good at perceiving patterns. Part of this comes from their concept-based processing, where the perception is guided by familiar concepts. Another factor in pattern perception is the Gestalt effects that make people group together objects that are visually similar, aligned in some way or just close to each other, see fig. 2.

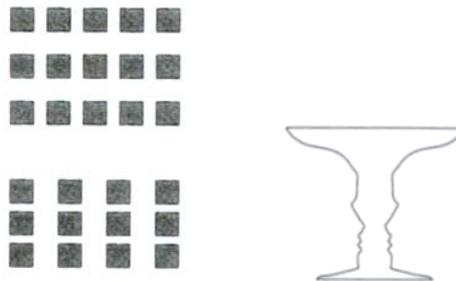


Figure 2. To the left, the Gestalt effect of proximity can be seen. One of the figures is seen as three rows and the other one as four columns. In the figure to the right, the vase is noticed before the profiles, illustrating the effects of closedness and size.

## 2.3 Design rules

A lot of rules have been established to facilitate the design of user interfaces. Very well known are Shneiderman's "Eight Golden Rules of User Interface Design" [1]. Except the Eight Golden Rules there are a lot of guidelines for more specific things, like colours and icons, that also are important to consider when designing a user interface.

### 2.3.1 "Eight Golden Rules of Interface design"

These rules must be interpreted, refined, and extended for each environment, but they are a very good pointer to follow when designing a user interface.

1. **Strive for consistency.**

This rule is the most frequently violated one, but following it can be tricky because there are many forms of consistency. Consistent sequences of actions should be required in similar situations and there should be a consistency of colour, layout, capitalization, fonts, and so on through the whole application.

2. **Enable frequent users to use shortcuts.**  
As the frequency of use increases, so do the user's desire to reduce the number of interactions. Abbreviations, special keys, hidden commands, and macro facilities are appreciated by frequent knowledgeable users. Short response times and fast display rates are also important.
3. **Offer informative feedback.**  
There should be system feedback for every user action. For frequent and minor actions, the response can be modest, whereas for infrequent and major actions, the response should be more substantial.
4. **Design dialogs to yield closure.**  
Sequences of actions should be organized into groups with a beginning, middle and end. The informative feedback at the completion of a group of actions gives users the satisfaction of accomplishment.
5. **Offer error prevention and simple error handling.**  
As much as possible, design the system in such way that users can not make a serious error. For example, do not allow alphabetic characters in numeric entry fields and prefer menu selection to form filling. If users make an error, the system should detect the error and offer simple, constructive, and specific instructions for recovery.
6. **Permit easy reversal of actions.**  
Actions should be reversible as much as possible. This feature relieves anxiety and encourages exploration of unfamiliar options, since the user knows that errors can be undone.
7. **Support internal locus of control.**  
Experienced operators strongly desire the sense that they are in charge of the system and that the system responds to their actions. The users should be the initiators to actions rather than the responders to actions. Surprising system actions or inability to produce the action desired build anxiety and dissatisfaction.
8. **Reduce short-term memory load.**  
The limitation of human information processing in short-term memory (the rule of thumb is that humans can remember  $7 \pm 2$  chunks of information) requires that displays be kept simple. Sufficient training time should be offered for codes, mnemonics, and sequences of actions. Where appropriate, online access to command-syntax forms, abbreviations, codes, and other information should be provided.

### 2.3.2 Layout

It is important to have a grid for the layout. A layout grid is a set of horizontal and vertical invisible help lines that break up the available space into visual units. These units will have visual as well as conceptual integrity in the eyes of the user. Putting things that belong together closer together is often enough to get a grouping effect (see chapter 2.2), sometimes borders are used to enhance the effect.

### 2.3.3 Icons

In computing, an icon is an image, picture, or symbol representing a concept. An icon can increase the comprehensibility since pictures sometimes say more than words. They are often smaller than text labels why icons often save valuable screen space. They can also contribute to better recollection and a visually more appealing user interface. Some icon specific guidelines should be considered:

- Represent the object or action in a familiar and recognizable manner.
- Limit the number of different icons.
- Make the icon stand out from its background.
- Consider three-dimensional icons; they are eye-catching, but also can be distracting.
- Ensure that a single selected icon is clearly visible when surrounded by unselected icons.
- Make each icon distinctive from every other icon.
- Ensure the harmoniousness of each icon as a member of a family of icons.
- Really good icons should work well on both black and white displays as on colour displays.

The “danger” with icons is that they can be very difficult to design and, for the users, difficult to interpret. Cultural differences can also result in different interpretations of the icons. Another problem is that there is an icon crisis, which means that it is getting more and more difficult to find icons to represent new things since most of the icons are already busy.

### 2.3.4 Colour

Colour can be used to make a user interface more appealing visually, to group information, to create some associations or to indicate importance or structure. But it is very important not to get carried away since it is very easy to get a cluttered result. A good idea is to design in monochrome and add colours afterwards. Some guidelines are to use five colours maximum and to keep the background colours very discrete. Not to forget is the psychology of colours where different colours have different meaning like red often means warning, hot or stop. These differences in the interpretation of colours can also depend on cultural differences. Another important issue that has to be considered when dealing with colours is that about 8% of the population are colour blind [4].

## 2.4 Usability testing

It can be very difficult to put the meaning of usability in words. Eason [2] states that “a system is usable if it is in fact used in practice”. The only drawback with this definition is that the usability can not be measured until the system has been installed.

Various methods have arisen to be able to test a system’s usability.

### 2.4.1 Usability testing methods

Different methods are suited for different phases of a project, see table 2.

Method	Phase				Resources			
	S	D	P	I	RU	SU	UE	DT
Concept test	Dark	Light	Light	Light	Dark	Light	Light	Dark
Heuristic eval.	Light	Light	Dark	Dark	Light	Light	Dark	Light
Checklists	Light	Light	Dark	Dark	Light	Light	Light	Dark
Walkthroughs	Light	Light	Dark	Dark	Light	Light	Dark	Light
Empirical test	Light	Light	Dark	Dark	Dark	Light	Light	Dark
Field trial	Light	Light	Dark	Dark	Dark	Light	Light	Dark
Follow-up study	Light	Light	Light	Dark	Dark	Light	Light	Dark
Field study	Light	Light	Light	Dark	Dark	Light	Light	Dark

Table 2. The S, D, P and I phase refers to study, design, prototyping and implementation. The different resources are real users (RS), simulated users (SU), usability experts (UE), and the development team (DT) itself. In both phase and resource dark grey cells mean that the method/resource is very suitable while light grey cells mean that it could be used.

The purpose of a *concept test* is to evaluate the results of a user and work-study, and to test whether the terminology and the identified domain objects and tasks are appropriate. Scenario storyboards and paper mock-ups are useful tools since it is generally done before the detailed prototyping has started.

A *heuristic evaluation* is a technique where a small group of usability experts evaluate a design, using a specified set of design guidelines.

*Checklists* can be constructed early in a project for later use in the evaluation phases, or they can be found in some books on user interface design. Checklists are long lists of aspects to think about in a user interface design.

There are some *structured walkthrough* methods that can be used by the development team and/or by the usability experts. An example is the cognitive walkthrough that is based on a theory of human learning and action. These methods are usually quite tedious since they require attention to the smallest details of the user's tasks.

The most valid results are obtained by testing a design on real users, so called *empirical testing*. Many companies today have special usability labs with all sorts of equipment needed for empirical testing. An important thing about empirical tests, that is often forgotten, is to formulate the test tasks carefully. Telling the users to play around with the system will not give any useful evaluation results.

A *field trial* is kind of an empirical test. The only difference is that the users test the system in their workplace instead of in the lab.

The study of a system's usage in the workplace over a longer period of time is called a *field study*. This evaluation method comes closest to Eason's definition of usability.

## 2.4.2 Testing the design with users

Before testing a design it has to be decided what information will be gathered. It is useful to distinguish process data from bottom-line data. Process data are observations of what the test users are doing and thinking as they work through the tasks. These observations show what is happening step by step and will hopefully give a suggestion to *why* it is happening. The bottom-line data explain what happened: how long time did users need to perform the tasks, where they successful, how many errors did they make. The satisfaction of the users is also very useful information.

Depending on what information is considered to be of greatest importance to gather there are some different techniques. Simple observation has always been a common way of recording errors and usage patterns. Task completion times can be collected with a stopwatch. The observation can also be made with the help of a video camera. Having the test on tape makes it easier to go back and review what the users did. However, the video camera can make the users a bit nervous.

Another technique for collecting data on times, errors and action patterns is to have the system log the entire interaction. The log can later be replayed and analyzed. To get to know what the users think while using the system there is a technique called the “thinking out loud” method. The idea is to let the user talk while he/she is testing the system and tell the observer what he/she is thinking. Another way to collect the users’ personal opinion about the system is to use questionnaires or interviews.

User observations through the thinking out loud method, where the users tell the observer what they are thinking while they are testing the application, will not give statistical results. It will give the developers a good idea of where people have difficulties in using the application and the possibility to improve these parts. When testing users a lot of ethical things have to be considered and the users must be comfortable and feel that they are not the ones that are being tested. To assure oneself that the procedure is being done according to the “rules” it is good to follow some guidelines written by Gomoll and Nicol [5].

### **1. Introduce yourself.**

### **2. Describe the purpose of the observation (in general terms).**

Set the participant at ease by telling that he/she is really helping to improve the product by trying it out in the early stages. That it is the product that is being tried out and not the participant, so probable troubles with some of the tasks are the product’s fault and not the participant’s.

### **3. Tell the participant that it is OK to quit at any time.**

Never leave this step out. The participant should know that he/she can quit at any time if feeling a little bit uncomfortable. This not only an ethical testing procedure, it is required by the American law. Participants should not feel like they are locked into completing tasks.

- 4. Talk about the equipment in the room.**  
Explain the purpose of each piece of equipment (hardware, software, and so on) and how it will be used in the test.
- 5. Explain how to “think aloud”.**  
Ask the participants to think out loud during the testing of the product, and tell him/her that this is the best way for you to get a lot of important and useful information. Listening to the users while they are working makes it easier to understand their expectations for the product, as well as their intentions and their problem-solving strategies.
- 6. Explain that you will not provide help.**  
It is very important not to provide the users with help when they get stuck or if they try to solve the problem the wrong way. If you provide a user with help in such a situation you will lose a lot of valuable information – where users have trouble and how they figure out how to do (if they figure it out). If any help should be given this should be decided before the tests start so all the participants will get the same help. It is important to try not to give the participants any more information than the true users of the product would have.
- 7. Describe the tasks and introduce the product.**  
**Explain what the participant should do and give him/her written instructions for the tasks. If the prototype has to be demonstrated before the user observation begins, be sure not to demonstrate the tasks that will be tested.**
- 8. Ask if there are any questions before you start; then begin the observation.**
- 9. Conclude the observation.**  
When the test is over explain for the participant what you were trying to find out during the test and answer any remaining questions the participant may have. Discuss any interesting behaviour you would like the participant to explain.
- 10. Use the results.**  
The purpose of observing users is to see what parts of the product might be difficult or ineffective. Review all your notes carefully and thoroughly. Look for places where the participants had trouble, and see if you can determine how your product could be changed to alleviate the problems. Also look for patterns in the participant’s behaviour that might tell you whether the product is understood correctly.

## 3 Technological Background

Several melody composers in different mobile telephones were examined. Examining other, in this section similar applications, that work for users is a good way to gather ideas. Below are the melody composers in Ericsson's mobile telephones described very thoroughly, while the melody composers in mobile telephones of other brands are only described in the whole with some good and bad things about them. The iMelody format and the limitations of a mobile telephone had to be considered before designing the application. Both of the subjects are mentioned here.

### 3.1 Melody composers in Ericsson mobile telephones

Most of the melody composers in Ericsson mobile telephones are based on an old fashioned input method, the text string, see fig. 3. The melody format consists of the basic seven notes 'c', 'd', 'e', 'f', 'g', 'a', and 'b' with the possibility to raise or lower the notes half a step according to the rules of music. There are only two note durations, two octaves and one kind of pause in the format.



Figure 3. The text string in previous melody composers in Ericsson mobile telephones.

A regular note is represented as its corresponding lower-case letter in the text string. A longer note is represented with a capital letter, a higher octave with a plus in front of the note, and a flat or sharp note gets its representing musical sign 'b' or '#' in front of it. Every time a new note is put in, changed or walked past with the cursor, the user can hear its sound and length. This makes it easier to compose a melody since the user can hear what he/she is doing.

In Ericsson's regular mobile telephones the interaction is made through the keys where the keys "one" through "seven" represent the notes 'c' through 'b', the keys "eight" and "nine" represent the notes 'c' and 'd' in the higher octave. The toggle between a basic/flat/sharp note is made with the "hash" key while the octave is toggled with the key "zero". A pause is put in with the "star" key.

A special mobile telephone made by Ericsson is the R380, see fig. 4. The R380 has a much bigger screen than the regular mobile telephones, and this screen is also a touch screen. These qualities make it possible for the melody composer to look differently. The melody composer has a one-dimensional piano display that covers the keys of two octaves. Due to the touch screen the piano is used as a regular piano by pressing the keys of the piano when composing a melody.



Figure 4. The Ericsson R380.

### 3.2 Melody composers in mobile telephones of other brands

Most of the mobile telephones' melody composers (if they had any) simply required the user to write the melody as a text string, just like the melody composers in Ericsson's mobile telephones. The more interesting mobile telephones were the ones with a graphical user interface. Some of them are described in this chapter with some personal reflections about positive and negative things with the user interfaces and the interaction.

#### **Sagem MC932**

The user interface of the melody composer in the Sagem MC932, see fig. 5, consists of a nice looking staff and three softkeys; options, listen, and help. The options menu offers the possibilities to "Change the tempo", "Save", "Quit" and so on. The help menu describes each key's function like "C. Previous note" and "7. Delete a note".



Figure 5. The Sagem MC932.

- + The help function is very useful since it is difficult to remember what each key does.
- + An error message pops up when an illegal action according to the musical format is being done.
- The keys' functions are not intuitive. For example, the function of the key "C" is to delete letters and digits when writing in the mobile telephone, but in the melody composer the function of the key "C" is to go to the previous note. The key seven is used as delete.
- Since the keys' functions are not intuitive the memory load is very high on the user when trying to write a melody. If the melody composer is rarely used it is almost impossible to remember how it works between the times of usage.
- There is no sound response when writing a new note or when passing it with the cursor. The only possibility to listen to one note is to write it and only it and push "listen". Otherwise there is only the possibility to listen to the whole sequence of the melody written.
- There is very little possibility to play around in the melody composer by listening to the notes or more or less play a melody like on a piano. There are no keys representing the seven basic notes as in the Ericsson's melody composers.

### Samsung SGH-800

The melody composer in this mobile telephone, see fig. 6, uses a regular staff.



Figure 6. The Samsung SGH-800.

- + The keys “one” through “seven” are used as the basic notes ‘c’ through ‘b’ and sound with the corresponding sound when used.
- + The function of the “up”/”down” key is to raise/lower the note half a step (if possible otherwise it is raised/lowered a whole step).
- The “left arrow” key is used to change a note’s duration and the “right arrow” key is used to change a pause’s duration. These keys functions are usually to back and forth among the written notes, this is not the case.
- There was no function found that makes it possible to go back through the written notes a couple of steps and change a written note. The solution is to delete everything written after the note to be changed.

### Siemens S40

The melody composer in the Siemens S40 (see fig. 7) consists of an unprofessional kind of staff, see fig. 8. Instead of the regular five lines there are four lines to differ the different octaves, but there are not any lines in between to differ the notes. Since there are no lines to simplify the reading of the notes there is a box on the size that shows the values of the note on which the cursor is pointing. The values shown in the box are the note's basic value, the note duration and the octave.



Figure 7. The Siemens S40.

There are no conventional notes, only circles representing them. To show a note with a longer duration the circle becomes more like an ellipse, the longer the note, the longer the ellipse. Pressing the key corresponding to the note value several times changes the duration of a note. It can also be changed with the key nine.

The volume keys on the side of the telephone are used to change the octave of the note.

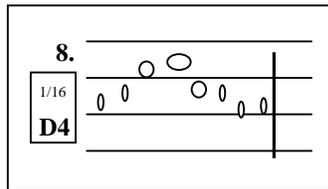


Figure 8. A replica of the graphical user interface of the melody composer in the Siemens S40.

- + The keys “one” through “seven” are used as the basic notes ‘c’ through ‘b’ and sound with the corresponding sound when pressed.
- + Walking through the written notes the note’s sound is heard.
- + The “star” key carries a help function. When it is pressed a nice looking graphical display is shown over the telephone’s keys and their function in the melody composer.
- + The function of the “up”/“down” key is to raise/lower the note half a step (if possible otherwise a whole step).
- + This kind of graphical representation can be easier to understand for users without any knowledge about notes. They do not have to know the actual syntax for a note with a long duration, they can see it on the ellipse’s width.
- This graphical solution can be very annoying for a user with musical skills.
- In one of the corners there is a counter counting the number of notes put in. A good idea but it does not fill its purpose since it starts to count from the number one and up. If the user does not know the maximum limit, this counter does not help at all. It should be counting down instead so the user can see how many more notes can be written.

### Siemens SL45

In the Siemens SL45’s (see fig. 9) melody composer there are dashed lines under and above the regular staff to make it possible to use a higher amount of octaves. When the user goes far under or above the regular staff so the regular staff disappears, the dashed lines are supposed to help the user.



Figure 9. The Siemens SL45.

- + The keys “one” through “seven” are used as the basic notes ‘c’ through ‘b’ and sound with the corresponding sound when pressed.
- + The function of the “up”/“down” key is to raise/lower the note half a step (if possible otherwise a whole step).
- + It is possible to walk through the notes already written and change a note. While walking through the notes the corresponding sound sounds.
- + There is a counter that counts down from the maximum amount of notes and inform the user about how many more notes can be written.

- The dashed lines do not help much when the regular staff comes out of sight since it is very difficult to know which of the dashed lines is which.

### 3.3 The iMelody format

Several mobile phone companies have designed a new more complex melody format [6]. This format is supposed to emulate the real musical format as much as possible, so that the users can compose advanced melodies. Furthermore, the co-operation makes it possible to send the melodies between mobile telephones from different brands.

The format consists of the seven basic notes and all the flat and sharp notes existing in the regular music notes. Each note can be toggled nine octaves and it can have six different duration with three duration specifiers. The pause can also have all the different durations.

The format also holds a couple special effects that can be set in a melody. It is possible to set the back light, the led, and the vibrator on and off. The volume can also be modified to a completely different volume or it can just be increased or decreased one step from the previous volume. A melody's name, composer, beat, style, and an overall volume can also be set if wanted by the user. According to the format there is no limit to a melody's size.

### 3.4 The limitations of a mobile telephone

A mobile telephone's display is much smaller than for example a computer screen. This is one of the reasons why it is much more difficult to design a user interface for a mobile telephone. The small size of the screen limits the possibilities to display a lot of functions at the same time. Menus, for example, must often be scrolled although they are not very long. A smaller display makes great demands on the disposition of it and what functionality to display or hide. Studies have proved that tasks are usually performed faster and with less error rate on a larger screen comparing to a small one [7].

Comparing to a computer screen's resolution, the resolution of a mobile telephone's display is quite bad. The resolution determines the size of the icons and how detailed they can be drawn. The small display makes it desirable to have small icons to be able to fit as many things as possible on the display, but that can be almost impossible to accomplish because of the bad resolution.

Another difficulty comes with the limited number of keys. A key can have different functionality depending on in which application it is used (sometimes a key's function is to input digits and sometimes it puts in letters, when playing a game the key "two" can for example be used as "up" and so on). The lack of space on and around the keys on the mobile telephone reduces the possibility to mark them with all their functions. Even if there were space enough this would be a very bad solution. The mobile telephone's design would become very daub and there would be more text than anyone would have the patience to try to read. A good solution would be to have softkeys as is common in applications for computers. The softkeys are used by choosing them with the mouse. The previously mentioned small display and the lack of a mouse repudiate this solution. The

solution would be applicable on a bigger touch screen as it is on the R380. A touch screen is unfortunately an unusual thing on mobile telephones of today.

The processor in a mobile telephone determines the complexity of an application. Very complex applications consume a lot of processing power and can result in very slow applications. Nobody likes a slow application and therefore should these applications be avoided.

The mobile telephones' memory is also limited. The iMelody format's lack of size limitation is not applicable in the reality. The same goes for the format's range of nine octaves. The speakers in the mobile telephones do not have as large range as the format demands.

## 4 The Development of the Melody Composer

The development of the melody composer was carried out as much as was possible according to the school methodology described in chapter 2.1.2.

### 4.1 Vision

The user interface in the melody composer will have a piano as an analogical metaphor. When pressing a key representing one of the basic notes, the corresponding key on the piano will become pressed for a while, as the sound of the note will be heard and the note will be written on the display. The visual response of the piano and the sound reproductions are supposed to simplify the users' work while composing a melody. This should make it easier for the users, especially those with little musical skill, to write a melody since they can hear what they are doing. They can for example hear that an eighth note is shorter than a half note. Since the applications in a mobile telephone should be possible to use anywhere the user may not have a music handy. This is also when these factors should help the user to manage anyway. The graphical display is also supposed to attract the users just to play with the application without having a serious goal to write a melody. A new melody may arise from the playing. The users should get a nice experience and be tempted to use the melody composer again another time.

As mentioned above, will the melody be put in with the mobile telephone's keys where the most necessary functions will be available. More advanced functions will be hidden to be used by the more advanced users.

### 4.2 User definition

Defining possible users result in four different groups. The users with technical interest mean the ones that use their telephone more than just for calling people. The four groups would be:

- Users with minor technical interest and minor musical skills
- Users with technical interest and minor musical skills
- Users with minor technical interest but with musical skills
- Users with technical interest an musical skills

Users with minor technical interest that do not use the applications in mobile telephones will (unfortunately) not suddenly start to use the melody composer, so these groups can be excluded. A small possibility would be that very musically skilled users would start to use the melody composer. This is not very likely since users with minor technical skills often feel intimidated with the applications or think that they are too detailed to use. The group, of the two remaining groups, of users with minor musical skills will still be kept as a target group since there is a lot of help for them on the Internet with written melodies to copy.

For the users with minor musical skills the most important functions for writing a melody have to be easily found. The application should also be fun to use to encourage them to use it.

For the users with musical skills there should be functions to be able to write a more advanced melody.

After asking people with and without musical skills if they would prefer a melody composer with a staff or a piano there was no doubt about that the piano was preferred. The usual reaction was that the piano would be more fun and not as serious as a staff. A lot of people start to play around on a real piano even though they do not know how to, but very few of them sit down and write music. Another reaction was that the old text string is not very much fun to use. The decision was made to continue with the old idea with a graphical piano, this with the intention to tempt more users.

The usability goals according to REAL were set.

*Relevance:* The users should be able to write an easy melody after a short time of practise.

*Efficiency:* The most important functions should be found directly through the mobile telephone's keys. The more advanced functions and special effects that can be used but will probably be used more rarely will take some more time to retrieve since they will be put further down in the system. Since the melody composer is an application of amusement rather than of necessity there are no specific efficiency goals but the fact that the application should not be too slow. The time to complete a task like writing a single note should be very short. The time to complete a whole melody will mostly depend on the users musical skill and the melody's length. There will of course be an error rate because of the lack of key labelling, which will give the users a certain memory load. Connecting the mobile telephone's keys to the graphical piano's keys should reduce this memory load.

*Attitude:* The user should think that the melody composer looks nice, that it is quite easy to use, and that it is fun to use it. All this should result in that this is an application that they can imagine to use.

*Learnability:* Users should be able to understand and use the most important functions after a short time of trial and error. The more advanced functions will be found after a time of use. It can be difficult to remember a key's exact function from one time to another but it should be easy to retract their functions from memory after a very short time of trial and error.

## 4.3 Design

The design can be a never-ending process because of all the ideas that arise during the development of the application or after user tests. This is why the design often is an iterative process where things in the design are changed when flaws with the design are

discovered or when the designers or users come up with better solutions for the user interface. The design for the first prototype of the melody composer, that is described here, is the result of such a process. The design has been made in co-operation with several employees at Ericsson Mobile Communications AB, Daniel Helgesson, Magnus Ernstsson, Per Ögren and Robert Najdenovski.

#### 4.3.1 The design of the application's services

Since the melody composer had to support the iMelody format there was not much to discuss. The melody composer had to support everything the iMelody format requires and as much as possible of the options (like beat, style, etc).

The melody composer should have the functionality of the seven basic notes and all the flat and sharp notes existing in the regular music notes. Each note should have the possibility to be toggled between different octaves and it should be able to have six different durations with three duration specifiers. The pause should also be able to have all the different durations.

A couple of the special effects (back light, led and vibrator) that can be set in a melody should be available. Furthermore should the possibility be to modify a melody's volume to a completely different volume or increase or decrease it one step from the previous volume. A melody's name, composer, beat, style, and an overall volume are more optional functionality.

There will be no possibility to write polyphonic melodies in the melody composer and there will be no special effects beyond the format because that would ruin the compatibility with mobile telephones of other brands.

#### 4.3.2 Conceptual design

The functions that had to be put so they can be used directly through the keys of the mobile telephone are the functions that are most basic for writing a melody. The functionality of the seven basic notes is the most important one. Other important functions were, to be able to put in a pause, change the duration of a note or pause, change the octave and raise or lower a note half a step (flat or sharp note). This way of input was later on called the "standard mode".

The "piano mode" (chosen by the "volume up" key) inactivated the keys "zero" through "nine" and the "hash" and "star" keys. In this mode, the "left" and "right error" keys are used to walk through the keys of the piano (with sound reproduction) instead of walking through the melody string. When the user comes to the desired note and chooses it with the key "Yes", the note will be written in the melody string. To be able to change a note or something else in the melody string the user has to go back to the standard mode.

More advanced things about the melody's behaviour and the special effects were put in the option menu. The option menu turn up when the new "option" key on the mobile telephone is pressed. The things put in the option menu was the:

- melody's overall volume
- volume modifiers
- change of the volume to a specific value anywhere in the melody
- led on/off
- back light on/off
- vibrator on/off
- repeat start/stop
- tempo
- style
- duration specifiers

The name and composer of the melody were optional in the format and therefore skipped in the user interface but they are supported in the implementation. The user will be able to set a name (this could be the name of the melody) but the name will only be a way to identify the melody in the telephone.

The title bar that exists in most of the applications and menus to facilitate the orientation in the mobile telephone was removed to increase the space in the window to be used. The always-existing piano on the top of the window had to do as an orientation mark. The piano is static except the fact that the keys can look pressed (one at the time). This is achieved when a note is put in, when changing a note's octave, duration or when toggling it basic/flat/sharp. Even when passing a note in the melody string with the cursor, or when walking around in the piano in piano mode, will the keys of the piano look pressed. The white keys of the piano will be labelled with their note values to make it easier for the users with less musical skills, see fig. 10.

The note 'b' was initially named 'h' as it is called in, among other places, Sweden. After receiving the knowledge that American people do not know the note 'h' (they call it 'b') it was changed. For a Swede the note 'b' is the note 'h' lowered half a step, but Swedes do understand that a note 'b' can mean the note 'h'.



Figure 10. The piano in the melody composer with the labelled keys.

The melody will be represented as a string with letters and icons. A note will consist of three parts: a regular note as the note duration, a letter to show the note's value and a raised digit to show the octave, see fig. 11. The different durations of the pause could be represented with the actual musical signs or with a note to describe the duration and a 'p' for pause. A lot of things were in favour of the latter solution. Having the actual musical signs would be more difficult for the users with no musical skills. Not knowing what duration a note represents, its length can be heard due to the sound reproduction when writing it. It is more difficult to hear the length of silence when writing a single pause

since the melody composer also will be silent before and after the writing. The meaning of the notes will be learned after some time of usage and will be easily adapted to the duration of a pause. It would be more difficult for the user to have to learn six more musical symbols.

Another problem was the two durations of a pause that are represented by the same symbol and are only differed by what line they are situated on in a staff. They would be difficult to differ since there will not be a staff in the melody composer.

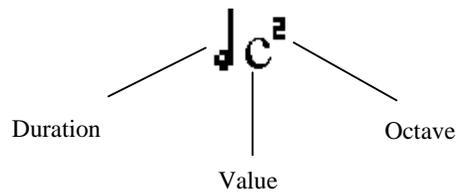


Figure 11. The group of icons that describe a note.

### 4.3.3 The design of interaction

A lot of the basic interaction has been solved as in the melody composers in previous Ericsson mobile telephones, see fig. 12. The only difference are the keys “eight” and “nine” that previously carried the note ‘c’ and ‘d’ in a higher octave and now function as ‘-’ and ‘+’ to be able to decrease or increase the duration of a note.

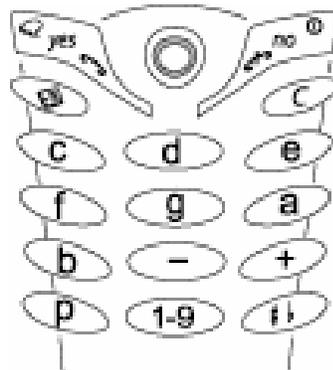


Figure 12. The mobile telephone’s key set and the keys’ functions in the melody composer.

The special effects that can be used anywhere in a melody, were grouped together and put in a menu as icons. The icons in the menu can be chosen and thereby inserted into the melody string. These special effects were the back light, led and vibrator on and off, repeat start and stop, volume modifiers and volume. The volume in the melody can according to the iMelody format be set to sixteen different values. This was a problem. All the sixteen icons could not be inserted into the menu since it would make the menu

very long. Only one of the symbols was inserted and then the value can be changed with the keys “eight” and “nine” (‘-‘ and ‘+’). This solution was expected to give the users quite some problems.

There were a lot of discussions of making the duration specifiers available by several presses of a key representing a note, just like when writing alphabetical letters in the mobile telephone and the user has to press the key “two” twice to get the letter ‘b’. This solution would require a small window in a corner, as in the rest of the mobile telephone, which pops up when pressing one of those keys to inform the user that more choices can be made. This corner window slows down applications a little bit. Another argument against this solution was that the duration specifiers will be used very rarely and should be put further down in the system. The duration specifiers were put in a multiple-choice window where the name or explanation of the different duration specifiers was written. The user can choose one of these to attach on a note. The different styles of a melody are also chosen through a multiple-choice window.

Scrolling a scrollbar to the desired volume sets the melody’s overall volume. The input of the tempo could not be solved in the same way because of its wide range (25-900). The tempo has to be written by hand.

There was a lot of error prevention done in the design as the mentioned scrollbar that only goes between the existing volume limits and the multiple-choice windows that naturally only offer existing choices. The window that handles the tempo only accepts input in the available range.

#### 4.3.4 Graphical design

The display of the melody composer only consists of two areas, the piano on the top of the display and the text area below. The Gestalt effect of proximity was used to group several icons to describe for example a note. In this case there was a note icon for the note’s duration, a letter for the note value and a risen digit for the octave. The same effect was used to describe a pause and a couple of the symbols for the special effects.

Since the piano plays an important role of the application it was made quite big. A bigger piano reduces the text area. The smaller text area the smaller will the displayed part of the melody string be, but the sacrifice of the text area was considered worth it. The piano’s important role was to reduce the memory load of the keys and to increase the users subjective feeling towards the application.

The size and font of the letters were adjusted to be the same as in the rest of the telephone. The icons could not be made as small because they would be very difficult to see and were made just a little bit bigger which resulted in an increased row height.

The colours of the application were not considered at all since they will be adapted to the rest of the mobile telephone. The sounds sounded at the same places as in the melody composers in previous mobile telephones of Ericsson. The sound could be heard every time a note was written, if something was changed on a note or when passing a note with the cursor.

The icons, which can be viewed in fig. 13, got a final look for the tests.

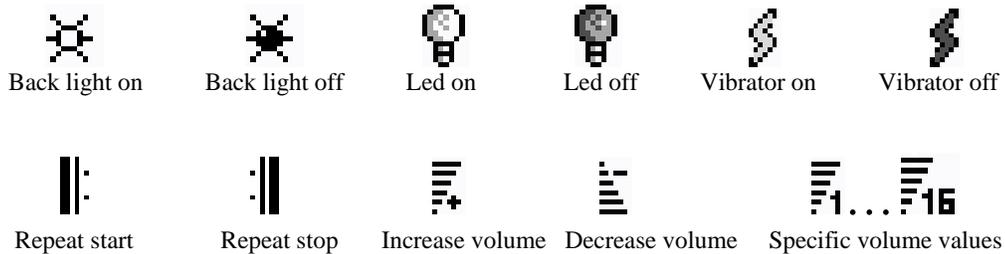


Figure 13. The different icons in the melody composer and their intended meaning.

## 4.4 Implementation

The implementation of the Master thesis was limited to the display object and was supposed to result in a prototype. Some difficulties and solutions that occurred during the implementation are described here. The implementation was made in C.

To be able to have melodies with unlimited length, according to the iMelody format, there was a requirement to save the melodies with as little memory usage as possible. The length of the melodies should also be dynamic. To meet the possibility of a dynamic length of a melody and not to allocate more memory than needed a linked list was used. Each element of the list could hold any of the possible things that can be written in a melody, a note with its duration and octave, a pause with its duration, led on/off, repeat and so on. The information of each list element was put into 32 bits. Eight bits were used for the type identification and the rest of the bits held necessary data, see fig. 14. The identification bits make the format easy to enlarge in case some new special effects would be defined in the iMelody format. The format was capsulated with “get” and “set” functionality so it would be much easier to use. The detailed definition of the bits can be viewed in appendix B.



Figure 14. The 32-bit format, where the head decides what type of the possible elements in the iMelody format it is, and the tail contains the data.

The new 32-bit format did not correspond to any of the existing formats why several parsers had to be written. The different formats were:

1. the user interface’s format with icons and fonts
2. the melody composer’s internal 32-bit format

3. the iMelody format consisting of a text string
4. the audio module's internal format

A very small parser was needed to convert the melody composer's internal format to the audio modules internal format when playing separate notes. Another parser constructed the user interface's format with icons and letters from the 32-bit format. The parsers between the iMelody format and the 32-bit format were the biggest and most difficult ones, one parser for each way. All the parser from the 32-bit format to the any of the other formats had to have functions that counted the amount of memory to be allocated before parsing.

A small problem was met when the already written functions for handling text scrolling and arrows, that pop up when scrolling is possible, were used. These functions assumed that the entire window was used for text only. The result was that the text was scrolled too late (the function thought that more lines would fit the window than it actually did) and that one of the arrows popped up over the piano. The scrolling and arrow handling functions had to be specially written in the melody composer.

A nice feature would have been to have a piano key pressed just as long as its note's duration. An implementation would probably not be as accurate as it would need to, to make the sound and key press simultaneous. This would also slow down the application when playing long notes. A constant time was set for all the piano key presses.

## 4.5 Usability evaluation

Different evaluation methods are suited for different stages of a development. A heuristic and an empirical evaluation were suitable methods, according to table 2, since the melody composer was in the prototype stage.

### 4.5.1 Heuristic evaluation

A heuristic evaluation is a way for designers to examine an application without users. The prototype of the melody composer is evaluated, by the author, according to the Eight Golden Rules of Interface Design, see chapter 2.3.1.

#### 1. **Strive for consistency.**

The missing title in the application is a violation of consistency comparing to most parts of the mobile telephone. This is a minor violation since the solution to increase the space of the window by omitting the title is used in a couple of places in the mobile telephone, as in the different games. The colours and fonts fulfil the consistency requirements since they are set as the default values in the entire mobile telephone.

All the icons specially made for the melody composer's melody string have the same height and are quite simple in their construction. Most of the icons are plainly black and white or are filled with a dark grey colour to imply that the function that the icon describes is turned off. The light bulbs are the only icons among the icons in the melody string that are made in grey scale and shaded. The notes also have a three-

dimensional touch according to the piano but almost no of the icons in the melody string are made three-dimensional in the same way. The icons in the melody string should be a bit more consistent. The piano can be kept three-dimensional since it makes it more eye-catching as it should be since it is an important part of the application.

**2. Enable frequent users to use shortcuts.**

This rule has not been accommodated to any great extent. Some help for frequent users may be the feature that a new note receives the previous note's duration and octave, since it is usual that similar durations and octaves are used throughout a melody. Unfortunately there are no shortcuts for frequent users to the special effects or some of the other special functions.

**3. Offer informative feedback.**

The melody composer follows this rule very well. There is a sound reproduction as soon as something is being done with a note. If the note's duration or octave is changed the user can hear the differences between the different durations or octaves. The icon representing the duration or the octave is immediately replaced with a new icon when changed. The sound is also heard when passing a note with the cursor. Every time any of the mentioned things are done the piano's key corresponding to the note becomes pressed for a while.

Good feedback is also given when walking around in the window with the icons for special effects where it is very easy to see which icon is "active". When choosing an icon the window is closed and the icon appears immediately in the melody string.

The multiple-choice windows inform the user about the choices made in a very nice way by delaying the choice when closing the window. The user will get a confirmation of the selection made.

**4. Design dialogs to yield closure.**

When writing the melody string an action is considered finished every time an icon or a set of icons pops up on the display. Other closures are the same as the feedback's mentioned under the rule number three.

**5. Offer error prevention and simple error handling.**

The melody composer has some obvious error prevention in the multiple-choice windows for the duration specifiers and the style. Only the choices available in the format are offered. The same goes for the volume bar. The input for the tempo only accepts the values in the available range.

Less obvious is the error prevention when making a note flat or sharp. Most notes can be toggled basic/flat/sharp but the notes 'e' and 'b' that only can be toggled basic/flat and the notes 'f' and 'c' that only can be toggled basic/sharp. This is made according to the basic rules of music and the iMelody format. Using the keys for toggling an octave or basic/flat/sharp will not work on other symbols but on notes. That means that these functions will not work on a pause or on a volume symbol.

When choosing one of the duration specifiers when the cursor is at a symbol that is not a note or a pause the duration specifier is not set. This is good error prevention, but it can also be confusing for the user since he/she may not understand why it did not work.

The iMelody format does not allow nested repeats and these are not taken care of in the melody composer. This means that there is no error message to inform the user of the problem, and this is a drawback in the same fashion as if the user presses the key “No” he/she leaves the melody composer. Leaving the melody composer this way will not save the written melody.

**6. Permit easy reversal of actions.**

A mobile telephone does not offer undo functionality as for example the Microsoft Word does. The undo function in the Microsoft Word remembers the changes made several steps back and allows the user to undo these changes. A similar function is not possible in a mobile telephone since the function uses a lot of memory to remember what has been done, and memory is a very limited resource in a mobile telephone.

Even though there is no undo function it is very easy to remove a written note or another symbol anywhere in the melody string. Some processes are reversible as a flat or sharp note. To change the note to a basic note, the note does not have to be deleted and rewritten. It is enough to toggle the note back to a basic note with the “hash” key (it toggles a note basic/flat/sharp).

A duration specifier can be removed by choosing the menu item “duration” in the option menu and set it to normal instead of one of the duration specifiers. The reversal of actions is overall very easy.

**7. Support internal locus of control.**

The only actions that work automatically in the mobile telephone are the back light that is turned off and the keys that are locked after a time of inactivity (these two automatic actions can be turned off if wanted). The user is otherwise in control of all the actions in the melody composer.

There are some actions that can result in another way than the user expects. The icons for back light and led can be difficult to know which is which. It can be very surprising for the user to expect that the back light will go on and see the led be turned on instead. The fact that the volume can be changed in several places can also be confusing. The volume can be set for the entire telephone, for the entire melody or anywhere in the melody where the volume can be changed to another value or just be increased or decreased one step from the previous volume in the melody. This is to many different ways to make it easy to understand for the user. The different volumes may be misinterpreted and give another result than the user has expected.

**8. Reduce short-term memory load.**

To reduce short-term memory load is the most important function of the piano since the keys of the piano (which are labelled) are mapped to the keys of the mobile telephone. The user does not have to remember which key of the mobile telephone represents which note. The piano mode reduces the memory load even more. The only thing the users have to remember is the functionality of the “left” and “right arrow” keys and the key “Yes”. The “left” and “right arrow” keys are used to walk around in the piano in piano mode while the key “Yes” is used to chose a note with. These features have reduced the memory load from the function of twelve keys to five keys.

**4.5.2 Empirical testing**

Two different tests were made with users. One of the tests was to check if the icons for the special effects were interpreted as they were intended to. In the other test was the functionality of the melody composer tested on real users.

**The icon test**

Twenty persons, most of them employees at Ericsson Mobile Communications AB participated in the test to check the meaning of the icons for the melody composer. The test simply consisted of a paper with the icons on one side of the paper and the explanations in a random order on the other side of the paper, see fig. 15. The participants had to connect the icons to the, in their opinion, correct explanations. The results can be viewed in table 3.

There was no trouble at all caused by the volume icons. They were interpreted as they were supposed to. The repeat icon, on the other hand, caused some trouble for participants without any musical skills.

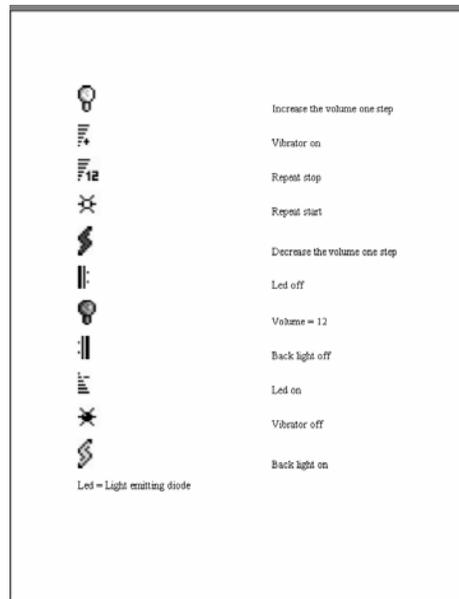


Figure 15. The icon test.

					No answer
Back light	15	3	1	-	1
Led	4	13	1	2	-
Vibrator	-	2	12	1	5
Repeat	-	1	2	14	3

Table 3. Some of the different icons and explanations. The numbers show how many of the participants connected an explanation with an icon.

The “light bulb” and the “sun” icons were not interpreted, as the designer wanted them to. The light bulb was supposed to represent the led and the sun was supposed to represent the light in the back. The majority of the participants interpreted these two icons in the opposite way.

Many of the participants did not like the vibrator icon (🔊) at all and said that this icon represents voltage and not vibration. In spite of the great hostility there was a high amount of correct answers, but there was an explanation to this. The participants solved all the other icons first and then there was only the vibrator icon left. Some of them even refused to connect this icon to an explanation.

A small problem over all was to know which of the filled or unfilled icons symbolised on and off. This problem would probably not exist as much if the test would be done on the mobile telephone. The back light would flow through the icons not filled and help the users to interpret the icons as they are supposed to.

### **User tests**

The time of the Master thesis was about to end so the user tests had to be accomplished although the prototype still had some flaws. This was a bit unfortunate since an important rule during user tests is to have an error free software because it causes feelings of uncertainty and can make the participant nervous. To minimize these possibilities the tests were done on the simulator.

Following the steps of the thinking out loud method the simulator had to be introduced thoroughly. It was explained for the participants that the simulator works just like a regular mobile telephone with the difference that the keys of the mobile telephone have to be clicked on with the mouse. Having the mouse, the display with the piano really looked as if it could be clicked on. The participants were informed of the fact that even if it looks like that in the simulator the display is not a touch-screen and that all the functions had to be accomplished through the keys.

The problems with the sounds in the simulator were also explained since the simulator could not reproduce the sounds correctly. The participants were told that the sounds in a real mobile telephone will be reproduced with the correct sound of the note’s value and with the right duration so the user will not have to know which note is long and which note is short since this could be heard.

They were also informed that no help would be offered during the test, and that if they felt that they could not manage one of the tasks and that they did not want to try any more they should say that and continue with the next task.

The participants received a paper with the following tasks:

1. Play around and test the application for a while
2. Write the following melody: see fig. 16
3. Delete the note 'c'
4. Set the tempo of the melody to 200
5. Insert "back light on" in the melody
6. Make the melody change the volume to five in the middle of it
7. Listen to the melody
8. Save the melody
9. Questionnaire



Figure 16. The piece of melody that the participants in the user test should write.

The playing around with the melody composer was important since the user can not possibly know which key of the mobile telephone does what since the keys are not labelled. Furthermore would the participants not be equipped with a manual so this first step was almost a necessity.

Since it is important to always give the participants real tasks, they received a template of a melody string. This would focus the observation on the interaction and the user interface instead of on the participants' musical skills and whether they can write a melody without a music or not. The melody string was not a real melody since it would not sound correctly in the simulator anyway. It was a short string that demanded that all functions available directly through the keys of the mobile telephone would be used. During this task the error rate and the difficulty to remember the different keys' function would be observed.

The following task with the deletion of the note 'c' was made to certify that changes are made easily in the melody composer, because tasks that the designers consider as very easy to solve are sometimes the most difficult tasks to solve for the users.

Even though the more advanced functionality of the melody composer is for advanced users and not for users that see the melody composer for the first time, some of these functions had to be tested. To set the tempo of the melody was an easier task to see if the participant would find the option menu at all. The following insertion of the back light was to see if the users would find the function and understand how to choose an icon.

Since the icons already had been tested no further notice would be taken to the users choice of icon.

The task to change the volume in the middle of the melody was to see if anyone would figure out the solution since it was not considered optimal and very difficult. Listening to the melody and saving it was tasks to give the participant a closure.

The intention of the questionnaire, which can be viewed in appendix C, was to make sure that the participants met the characteristics for the target group. It was also a way to really let the participants evaluate the melody composer.

When playing around with the melody composer most of the participants started to test the functionality of the keys systematically from the left to the right (from the key “one” to the key “hash”). One might think that this is a very good idea, that nothing can go wrong and that all the most important functionality will be found immediately. This was unfortunately not the case. The keys on the bottom of the mobile telephone start with the function of a pause then the key “zero” for changing an octave and the key “hash” for toggling a note basic/flat/sharp. When the participants tried the keys “zero” and “hash” after just writing a pause the keys appeared to them as having no functionality since a pause can not have different octaves or be toggled basic/flat/sharp. They had to try these keys later on again when looking for the two functionalities and by then they had often written a couple of notes in between.

There were some reactions on the keys “eight” and “nine” which decrease respectively increase the duration of a note or a pause. Several of the participants were a bit surprised that the two keys had the same functionality, some of them noticed the difference after some more testing. All the participants got confused for a split second when they saw the ‘p’ (for pause). Every one of them also said, “Oh, it must be a pause” right after.

All the participants understood the mapping between the keys of the piano and the keys of the mobile telephone. Some of them even counted the keys of the piano from the left to the right loudly till they reached the desired note. They did that to know what key to press on the mobile telephone. One of the participants suggested that it would have been easier to map the keys if there was a small mark after every three white keys of the piano. The mapped keys and the fact that the users understood the mapping almost immediately resulted in a very small, almost none-existent, error rate when writing the melody. The participants that had a bit higher error rates were the ones that did not have the patience to find all the basic functionalities in the first step and had to test a bit more in this step. A higher error rate in one case depended on a very nervous participant. One of the participants that sometimes compose melodies said that he/she usually use the manual to remember which key of the mobile telephone represents which note. This participant did not make a single error when writing the notes, even though there was no manual provided.

There were no problems to delete the note ‘c’ but the fact that the cursor was a bit difficult to see since it did not flash as it is supposed to. This feature had not been implemented yet.

Those of the participants that had found the options menu earlier had no problems at all with the changing of the tempo. The other ones looked for a menu in the “up” and “down arrow” keys or by pressing the key “Yes”. All these participants found the option menu after a while and had no troubles with the changing of the tempo.

The inserting of the “back light on” caused some troubles. The participants understood that they had to search in the option menu but it was more difficult to know which of the items to choose. They excluded the “Add symbol”, “Tempo”, “Volume”, and “Duration” items and only had the “Style” left which they did not understand. The less musically skilled participants chose “continuous” under style, said that they were not sure if that was the correct solution but they left it at that. When the more musically skilled participants saw the choices under “Style” they understood that this was not the solution. After a couple of tests it was more and more obvious that the problem was the label “Add symbol”. After the completion of all the tasks the participants were asked what they thought “Add symbol” meant. They answered that they thought that it only had to do with musical symbols, perhaps another way to insert notes or pauses, that was why it was not picked.

The most difficult task was, as expected, the change of the volume in the middle of the melody. Almost every one of the participants chose the “Volume” in the option menu and changed it to five. When they saw that nothing appeared on the display a lot of them suspected that they only had changed the volume of the entire melody. The ones more convinced of the fact and that had found the symbols continued to try to find another solution. Only two of the participants managed to solve it. A couple of the other had some ideas of increasing/decreasing the volume with the icons as many steps as needed to get the volume five.

The problem was discussed with the participants afterwards. One of the suggestions that came up was, that the symbol for a specific value of the volume (in the “Add symbol” window) could have the sign of amount (#) instead of a digit. The intention was to make it easier for the users to understand that the value could be changed. This suggestion was rejected by the same person with the explanation that only technically skilled users would understand the sign.

There were no problems with the listening and saving of the melody. Some of the users stumbled on the piano mode but no one understood the purpose of it or used it. When they were told of this feature after the test, they thought it was useful since the keys would not have to be used in such an amount. One of the participants said that the piano mode would be a slower way since more key presses are needed to get to the desired note.

Another thing that was considered difficult was that the special effects, that can be set anywhere in the melody string, and those concerning the whole melody were mixed in the option menu.

One part of the questionnaire was an assurance that the participants in the test met the requirements as the target group. None of the participants were regular Ericsson

employees. Some of them were newly arrived summer workers, others were from the outside of Ericsson and had to sign a promise of secrecy agreement. The agreement can be viewed in appendix D (in Swedish). The eight participants were in their twenties and three of them were female. All the users used their mobile telephone for more than just calling at least once a week, and the majority of them used it daily.

The result of, how often the participants compose melodies, was very poor according to the questionnaire. That depended on the answer possibilities to the question and not on the participants. The participants that compose melodies quite often still, do not do it as often as once a week or daily so they had to answer “at least once a month”. The participants that only compose melodies from time to time but not as often as once a month had to answer “almost never”. The answer possibilities should have been often, sometimes and never.

The graphics of the melody composer received really high grades. All the participants thought that it looked nice or even really nice. Most of them also were of the opinion that it was quite easy to use. No one thought that it was very difficult to use but some of the participants graded the usability to be neither easy nor difficult. The lower grade depended on the functions in the option menu, not on the basic functions. The basic functions were considered quite easy to learn and to remember. Most of the participants could imagine using the melody composer. A good thing was that some of the participants that never had used a melody composer graded the possibility for them to use it as quite high. Only one of the participants could not imagine using the melody composer at all. This person said that it had nothing to do with this particular melody composer but with the fact that he/she thought that it is too much work and that he/she would never take the time. Furthermore, this person did not use a mobile telephone daily.

	What do you think about the melody composer's graphics?	Do you think it is easy to use?	Is the melody composer an application you could imagine to use?
Participant 1	4	3	5
Participant 2	4	4	3
Participant 3	4	4	3
Participant 4	5	4	5
Participant 5	5	5	5
Participant 6	4	2	1
Participant 7	4	3	5
Participant 8	5	4	4

Table 4. The results of the three last questions in the questionnaire are represented in the scale 1-5 since there were five alternatives when answering the questions. The grade one is the worst and five is the best.

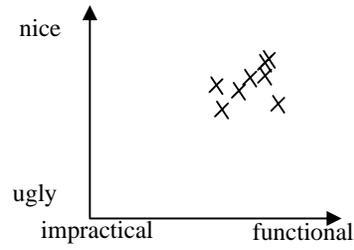


Table 5. The participants' opinions about the melody composer.

The participants' opinions regarding the melody composer are put together in a table and a diagram (see table 4 and 5). The result was very positive.

## 5 Conclusions

During the empirical and heuristic evaluations, several things were discovered that could be improved in the prototype of the melody composer. One of the things, which was not seen in previous Ericsson mobile telephones and was therefore excluded in the design to remain the consistency, was an error message when quitting the application without saving first. The error message helps the user to remember to save before he/she quits. This message has been put in some of the applications in the new mobile telephone and should exist in the melody composer as well.

An informative message could help the users to find all the functions available through the keys of the mobile telephone. This message should appear when the user tries to change the octave or toggle basic/flat/sharp on another symbol but a note, and it should inform the user that the particular key only can be used on a note. The same solution could be used on the duration specifiers since they only can be attached on a note or pause, but there is an even nicer solution. This solution would be to deactivate the choice of the duration specifiers in the option menu if the cursor was on another symbol but a note or pause.

The label “Add symbol” in the option menu should be changed since it was misinterpreted by most of the participants in the user test. “Features” or “Special effects” could be better labels. The possibility to change to the piano mode could be given in the option menu as well where more users would find it. To be able to change the piano mode with the “volume up” key could be kept as a shortcut.

The icons for the special effects really need to be improved since they did not do very well in the test. The meaning for the “light bulb” and the “sun” should at least be changed and the icon for the vibrator should be redone.

Even though the solution for the volume that can be set anywhere in the in the melody is a bit tricky it is acceptable since it is a special effect that only the most advanced users will use.

One thing that was thought of early in the development but was not adopted in the design, because of lack of knowledge of how to make it consistent with the rest of the mobile telephone, was a help function. The help function should describe the different keys’ functionality. It would be nice to have such a function where even the “volume up” key would be described and therefore the piano mode would become easier to discover.

An idea for future development could be to have an external small piano that can be attached to the mobile telephone just like the chatboard.

The prototype having a lot of things that can be improved it is still interesting to see how well the usability goals, which were set during the user study, were met.

*Relevance:* The user tests showed that the users had no trouble to write a melody, which was the goal.

*Efficiency:* The most important functions for writing a melody can be retrieved directly through the keys of the mobile telephone and the more advanced functions and special effects are retrieved from the option menu. The melody string was easily written during the user tests with a very small error rate. The tests also proved that the memory load was reduced, as it was intended, due to the graphical piano when the users looked at it to know what key of the mobile telephone to press.

*Attitude:* The results of the questionnaire with the diagram showed that the users liked the graphics in the melody composer and that they thought that it was quite easy to use.

*Learnability:* It was proved during the user tests that the most important functions were found quite easily and quickly. The difficulty to remember the keys' functions after some time was not tested but the fact that it was so easy for the participants of the test to learn them the first time this is not considered to be a problem.

Although there are several things that can be improved in the melody composer, many of the design rules for a user interface and human-computer interaction have been fulfilled. Furthermore, the usability goals were met, which can be considered very good since the tested application still was a prototype. The finished melody composer will be a very nice application that hopefully will be used quite a lot.

## Glossary

Chatboard	A small external keyboard that can be attached to a mobile telephone
Parser	In computer technology, a parser is a program, usually part of a compiler, that receives input in the form of sequential source program instructions, interactive online commands, markup tags, or some other defined interface and breaks them up into parts (for example, the nouns (objects), verbs (methods), and their attributes or options) that can then be managed by other programming (for example, other components in a compiler). A parser may also check to see that all input has been provided that is necessary.
Personal digital assistant (PDA)	PDA is a term for any small mobile hand-held device that provides computing and information storage and retrieval capabilities for personal or business use, often for keeping schedule calendars and address book information handy.
Snap on cover	Plastic covers of special mobile telephones, with different colours and motives, that can be replaced.
Softkey	A virtual key on a display that, on computers often can be clicked on, and in mobile telephones often have a physical key connected to it.

For musical terms, please see appendix A.

## References

- [1] Shneiderman, “Designing the User Interface: Strategies for Effective Human-Computer Interaction”, Third Edition, Addison-Wesley, 1998, ISBN 0-201-69497-2
- [2] Löwgren, “Human-computer interaction”, Studentlitteratur 1993, ISBN 91-44-39651-1
- [3] Haake, “Människa-datorinteraktion”, <http://www.eat.lth.se/Kurs/Material/MAM060/>
- [4] Mullet and Sano, “Designing Visual Interfaces”, SunSoft Press, Prentice Hall, PTR, 1995, ISBN 0-13-303389-9
- [5] Tognazzini, “TOG on Interface”, Addison-Wesley, 1992, ISBN 0-201-60842-1
- [6] Scales, “iMelody”, Infrared Data Association, 24 of October 2000
- [7] Nygren, “Grafiska användargränssnitt, några tips” Report no 60, CMD, Uppsala university, 1995, <http://www.hci.uu.se/papers/60>
- [8] Thinkquest, “Music Notes”, <http://library.thinkquest.org/15413>

# Appendix A

## Musical theory [8]

For readers with less musical skills here is a very short introduction about some musical terms and their meaning beginning with the most basic things.

A **staff** is made up of five horizontal lines and four spaces, see fig. 17.

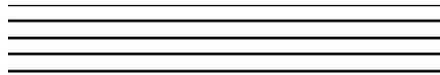


Figure 17. A musical staff.

The notes are **named** after the first seven letters of the alphabet (A B C D E F G)

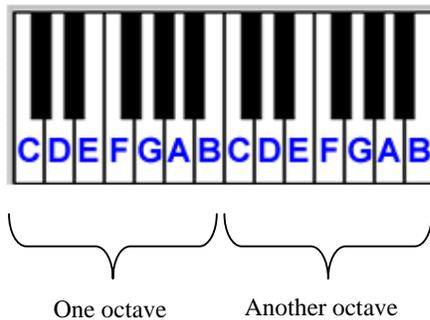


Figure 18. The keys of a piano in two octaves.

A note with twice as many vibrations as another note is named the same but sounds differently and is said to be in the next **octave**, see fig. 18.

Different notes and pauses have different **durations** which means for how long a note is played (or for a pause, how long the silence is kept before continuing playing). The different durations can be viewed in fig. 19.

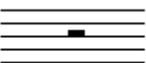
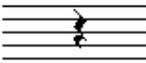
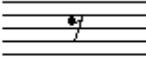
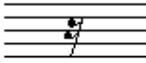
Name	Note	Pause
Whole Note		
Half Note		
Quarter Note		
Eighth Note		
Sixteenth Note		

Figure 19. The different durations of notes and pauses except the thirty-secondth note.

Beside the notes and pauses in fig. 19 there is another note. The thirty-secondth note and pause looks like the sixteenth note and pause but they have a third flag.

A **sharp** symbol ( $\sharp$ ) raises a note half a step while a **flat** symbol ( $\flat$ ) lowers a note half a step.

When a note has a dot after it is called a **dotted note** ( $\bullet$ ). The dot increases the note's duration with half of the note's duration (for example, a whole note gets the duration of one and a half note).

**Continuous** style means that there is no rest between notes and **staccato** style means that the notes are shorter and the rest periods longer.

## Appendix B

The different items that a melody can consist of according to the iMelody format and their 32-bit representation in the melody composer are described below. The grey areas, in the 32-bit bars, represent the occupied bits. All tables show the numerical values and the different items that they are representing. The different types and their numerical representation can be viewed in table 6.

Type	
0	Note
1	Pause
2	Volume
3	Volume modifier
4	Led
5	Vibrator
6	Back light
7	Repeat

Table 6. The different types and their numerical representation.

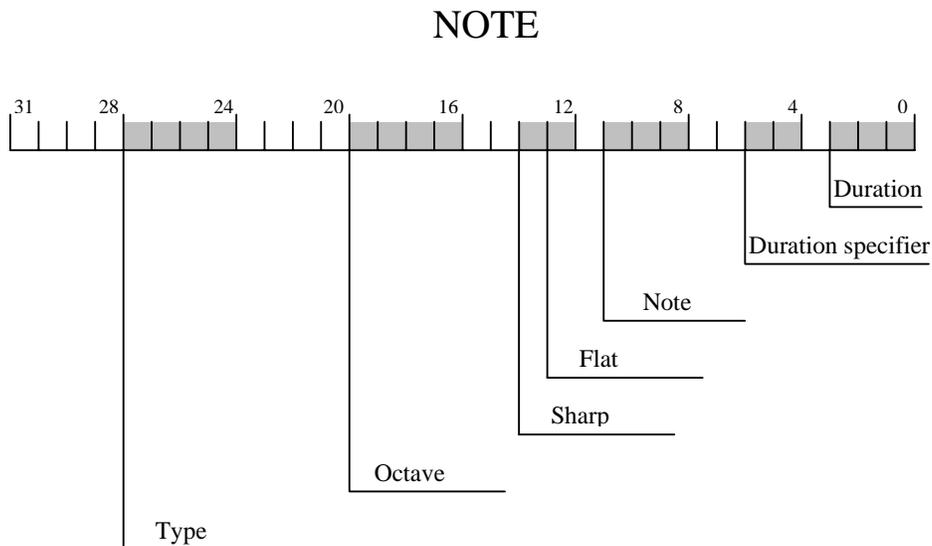


Figure 20. The 32-bit representation of the note.

Octave	
0	Octave = 0
1	Octave = 1
2	Octave = 2
⋮	⋮
8	Octave = 4

Sharp	
0	No sharp note
1	Sharp note

Flat	
0	No flat note
1	Flat note

Note	
0	'c'
1	'd'
2	'e'
3	'f'
4	'g'
5	'a'
6	'b'

Duration specifier	
0	No special duration
1	Dotted note
2	Double dotted note
3	2/3 length of a note

Duration	
0	Full-note
1	1/2-note
2	1/4-note
3	1/8-note
4	1/16-note
5	1/32-note

Table 7. A note's different items and their possibilities with the numerical representations.

### PAUSE

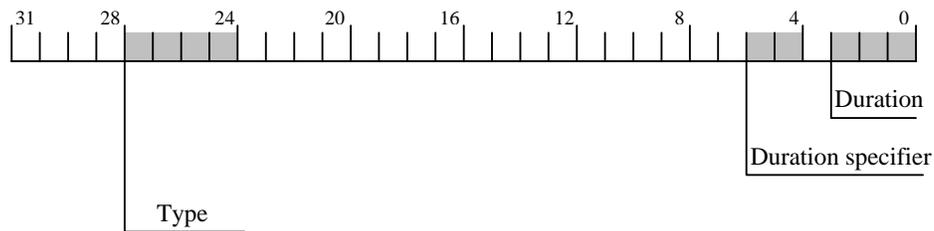


Figure 21. The 32-bit representation of the pause.

See tables for duration and duration specifiers under note

### VOLUME

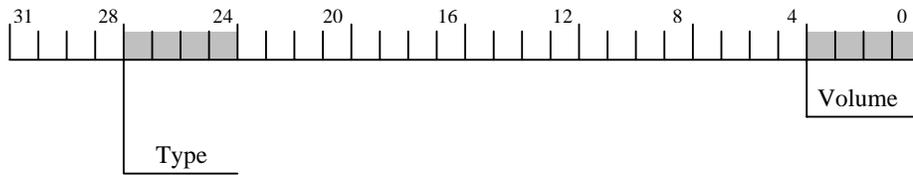


Figure 22. The 32-bit representation of the volume.

Volume	
0	Volume = 0
1	Volume = 1
⋮	⋮
15	Volume = 15

Table 8. The different possibilities for volume.

### VOLUME MODIFIER

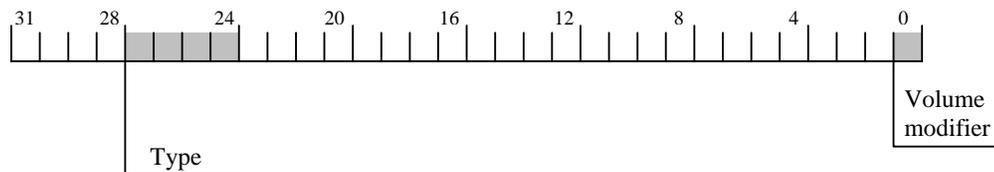


Figure 23. The 32-bit representation of the volume modifier.

Volume modifier	
0	Decrease volume
1	Increase volume

Table 9. The different possibilities for volume modifiers.

## LED

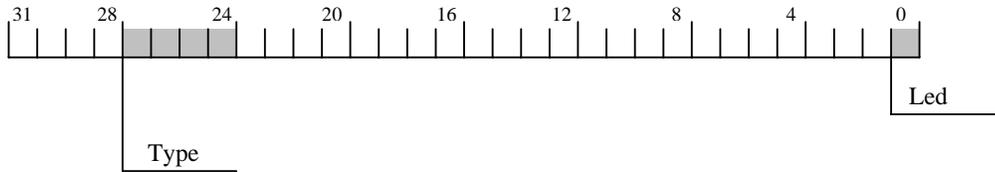


Figure 24. The 32-bit representation of the led.

Led	
0	Led off
1	Led on

Table 10. The different possibilities for led.

## VIBRATOR

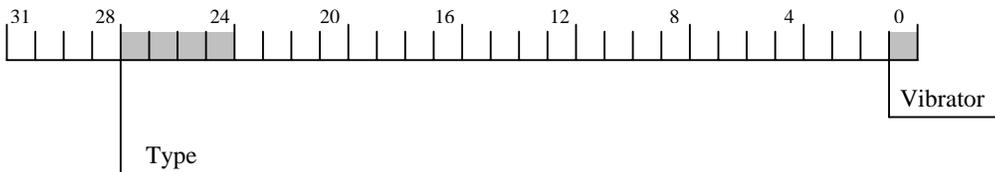


Figure 25. The 32-bit representation of the vibrator.

Vibrator	
0	Vibrator off
1	Vibrator on

Table 11. The different possibilities for vibrator.

## BACK LIGHT

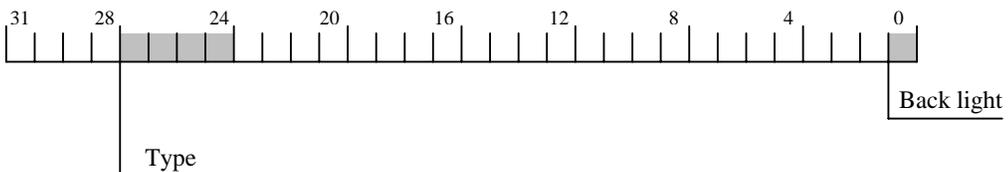


Figure 26. The 32-bit representation of the back light.

Back light	
0	Back light off
1	Back light on

Table 12. The different possibilities for back light.

### REPEAT

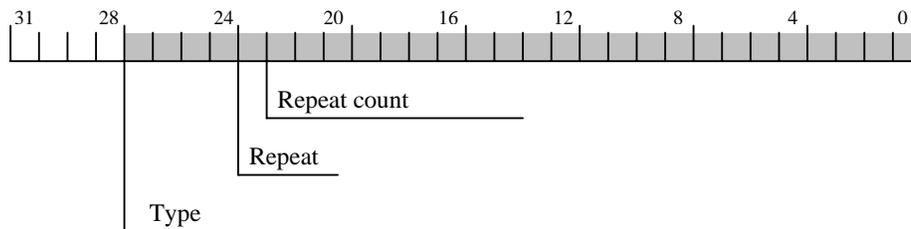


Figure 27. The 32-bit representation of the repeat.

Repeat	
0	Repeat stop
1	Repeat start

Repeat count	
0	0
1	1
⋮	⋮
8388607	8388607

Table 13. The different possibilities for repeat and repeat count.

## Appendix C

Female       Male

Age \_\_\_\_\_

How often do you use a mobile telephone?       Daily  
 At least once a week  
 At least once a month  
 Almost never

Do you usually use your mobile telephone for more than just making calls like SMS, games, phonebook etc.       Daily  
 At least once a week  
 At least once a month  
 Almost never

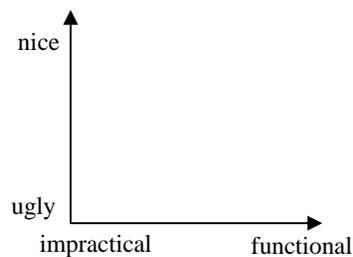
Do you usually compose/ download melodies in your mobile telephone?       Daily  
 At least once a week  
 At least once a month  
 Almost never

What do you think about the melody composer's graphics?      very ugly                              very nice

Do you think it is easy to use?      very difficult                              very easy

Is the melody composer an application you could imagine to use?      no, not at all                              yes, gladly

Please evaluate the melody composer's functionality and its appearance.



Thank you for your participation!

## Appendix D

### Avtal rörande användbarhetstest vid Ericsson Mobile Communications

Jag, \_\_\_\_\_ (Namn) \_\_\_\_\_ (Personnummer)  
intygar härmed att jag har accepterat att delta som testpilot i ett användbarhetstest. Som  
testpilot kan jag komma att ta del av konfidentiell information rörande Ericsson Mobile  
Communications (Ericsson) eller andra bolag inom Ericssongruppen.

Jag är medveten om att jag inte under några omständigheter får lämna några uppgifter om  
Ericssons produkter och verksamhet till obehörig person utan Ericssons skriftliga  
godkännande. Med *obehörig* avses här alla personer inom och utanför Ericsson, vilka ej direkt  
är engagerade i testet.

Efter utfört test skall jag återlämna all utrustning och dokumentation som erhållits av  
Ericsson.

Jag förbinder mig att följa dessa föreskrifter, och är medveten om att överträdelse av  
tystnadsplikten beträffande konfidentiell och/eller företagshemlig information kan medföra  
laga påföljd.

Jag godkänner vidare att Ericsson dokumenterar testet genom en videoinspelning. Jag förstår  
att denna inspelning endast kan visas för personer involverade i produktutvecklingen och inte  
kan användas för något annat syfte utan mitt skriftliga medgivande.

Jag ger också Ericsson äganderätt och exklusiv nyttjanderätt till de eventuella idéer och  
förslag som jag bidrar med i samband med testet.

Jag förstår att mitt deltagande är helt frivilligt och att jag kan avbryta när jag vill.

Ericsson Mobile Communications behåller originalet av detta dokument och såsom testpilot  
har jag fått en kopia.

Lund den .....

Bevittnas av ansvarig person:

\_\_\_\_\_  
Underskrivande testpilot

\_\_\_\_\_  
För Användbarhetslabbet

Learning to compose melodies is really about learning to free your own personal melodies from the chains of inhibition and censorship. You have to be able to accurately describe, and understand what you hear in your head. This can only happen if you: Have a framework for describing the melody that is deeply ingrained in your mind. Have the ability to connect that framework with what you hear. I can probably transcribe a melody that goes from G to F, and then, F to E – but that means very little at this point. It is only when I back that up with understanding of how single lines can actually outline harmony, and how that harmony dictates the way notes resolve to other notes! This is when I start to understand my melody. `lstm_melody_composer_train.py`. `README.md`. Melody Composer. This is the implementation of a LSTM Recurrent Neural Network that composes a melody to a given chord sequence. This work is based on konstilackner's thesis work. © 2019 GitHub, Inc.