



DEGREE PROJECT, IN COMPUTER SCIENCE , SECOND LEVEL
STOCKHOLM, SWEDEN 2015

Real time animation control in Android

IMPROVING THE WORKFLOW OF CREATING AND CONFIGURING ANIMATIONS

OSVALD IVARSSON

KTH ROYAL INSTITUTE OF TECHNOLOGY

SCHOOL OF COMPUTER SCIENCE AND COMMUNICATION (CSC)

Real time animation control in Android

Improving the workflow of creating and configuring animations

Osvald Ivarsson

osvald@kth.se

2015-06-10

Master's Thesis at CSC

Supervisor at CSC: Björn Thuresson

Examiner at CSC: Anders Lansner

Principal: Bontouch - <http://www.bontouch.com/>

Supervisor at Bontouch: Christoffer Hirsimaa

Master's Thesis coordinator at Bontouch: André Strindby

Swedish title:

Realtidskontroll av animationer i Android

Förbättrar arbetsflödet vid utveckling och konfigurering av animationer

Abstract

Smartphones nowadays are more powerful than ever, which means that the number of animations in apps have increased rapidly. But the way that animations are developed and tweaked to look good has not improved in the same pace.

In this thesis the typical workflow when developing animations for apps are investigated. This workflow is then improved upon by developing a prototype capable of configuring animations directly on the mobile device. The improved workflow has the consequence of decreasing development time needed to configure animations while also increasing developer satisfaction.

Sammanfattning

Smartphones har blivit kraftfullare än någonsin, vilket har gjort att antalet animationer i appar har ökat snabbt. Men sättet som dessa animationer utvecklas och konfigureras på har inte förbättrats lika snabbt.

I det här examensarbetet undersöks hur ett typiskt arbetsflöde ser ut vid utveckling av animationer till appar. Det här arbetsflödet förbättras sedan genom att ta fram en prototyp som gör det möjligt att konfigurera animationer direkt på den mobila enheten. Det förbättrade arbetsflödet innebär att utvecklingstiden för att konfigurera animationer sjunker samtidigt som utvecklarna känner sig mer nöjda.

Table of Contents

1	Introduction.....	1
1.1	Background.....	1
1.2	Bontouch	2
1.3	Problem.....	2
1.4	Delimitation	2
2	Method.....	4
2.1	Interviews.....	4
2.2	Surveys	5
2.3	Prototype.....	5
2.3.1	Development tools and strategies.....	6
2.4	Scientific result.....	7
3	Implementation and results	8
3.1	Phase 1 - Current situation at Bontouch.....	8
3.1.1	Interviews	8
3.1.2	Current workflow at Bontouch.....	9
3.1.3	Pre-survey questions	10
3.1.4	Method of asking the questions	11
3.1.5	Pre-survey results	11
3.2	Phase 2 - Development of a prototype.....	12
3.2.1	Development of the prototype.....	12
3.2.2	Technical result.....	12
3.2.3	Example of usage of prototype.....	15
3.3	Phase 3 - Evaluation of prototype and new workflow	19
3.3.1	Evaluation of prototype	19
3.3.2	Post-survey questions.....	19
3.3.3	Result of post-survey	20
4	Discussion	22
4.1	Higher satisfaction with the new workflow	22
4.2	Decrease of development time and number of contacts with designer..	22
4.3	Possibly allowing designers to directly fine-tune animations	23
4.4	Many configuration possibilities.....	23
5	Conclusion.....	24

6	Future work	25
6.1	Further improvements to the workflow	25
6.1.1	Let designers test Configuranimation	25
6.1.2	Investigate completely different alternative workflows	25
6.2	Technical improvements to Configuranimation	25
6.2.1	More default sets of parameters	25
6.2.2	More flexible adjustments in GUI	25
6.2.3	Include Javadoc directly in packaged library file	26
6.2.4	Open source the project	26
7	References	27
Appendix A.	Pre-survey questionnaire (in Swedish)	29
Appendix B.	Post-survey questionnaire (in Swedish)	30
Appendix C.	Answers to pre- and post-survey	31
Appendix D.	Page describing how to test prototype (in Swedish)	32

1 Introduction

1.1 Background

Smartphones nowadays are much more powerful than compared to just a few years ago, which means that the number of animations in apps have increased rapidly. But the way animations are developed and tweaked to look good has not improved in the same pace. This is a problem for the developers working with the creation of animations, since it is a very time-consuming task.

There are many different types of mobile platforms and animations to consider when talking about mobile apps. The animations considered in this project are only the animations of various interface elements like buttons, sliders, text views, etc. on the Android platform [1]. The animations performed are usually expansions of views or animated transpositions of views. Actual 3D animations as used in movies or games are not considered here.

There is also a distinction to be made between the creative part of creating animations and the more technical part of creating animations. Both of these parts are important, but in this report only the technical part of creating animations is considered. The technical part consists of practically configuring animations to behave and look good according to the design created by a designer. First the designer creates a beautiful looking design for a mobile app (the creative part) which includes both images and short videos of animations. After this the design is handed over to the mobile app developer who implements the design in the app (the technical part). After the initial version has been implemented by the developer, the designer provides feedback on the design and animations until the final design in the app is done.

An example of an animation is when a button is pressed which in turn expands a view showing some new information. The expansion of this view can be configured in many ways, but two main parameters are *duration* and *which interpolator to use* [2]. Duration is the time it takes for the expansion to finish, and interpolator is the type of curve which the speed of the animation should follow. For example a *Linear interpolator* would have a constant speed whereas an *Accelerate interpolator* would accelerate and become faster and faster during the execution of the animation. The fine-tuning of the duration and selection of which interpolator to use are typical tasks performed when configuring animations to make them look good. These tasks are usually performed by a developer.

When performing the very last tweaking of animations in order to make them look and feel good, it is important to do it directly on the target mobile device. In this case the target mobile device is a smartphone running Android. This means that the animation has to be configured in the programming code, after

which it is uploaded to the mobile device for actual inspection. This whole process takes a lot of time, since it is slow and usually has to be repeated many times.

The workflow as described above is one of the many ways that apps are developed nowadays. One of the companies developing apps like this is the Stockholm-based company Bontouch. The situation at Bontouch is used as an example case when trying to improve the general workflow. But any improvement to the workflow can be used by any developer who is creating animations, since an improvement generalizes to everyone.

1.2 Bontouch

Bontouch, the company that this work has been conducted together with, is a software consultancy firm located in the center of Stockholm. Bontouch develops mobile solutions with a focus on quality and long-term administration. Currently 40 people are employed at Bontouch, and all the development is done in-house. An office in New York was also recently opened.

1.3 Problem

As described above, the workflow when developing and configuring animations on the Android platform is a time-consuming task. The aim of this project is thus to find a way of improving the workflow when creating and configuring animations. This will be done by creating a prototype which makes it possible to try out a new workflow. The prototype will then be demonstrated to the developers at Bontouch in order to get quantitative data as feedback. An improvement to the current workflow would save many mobile developers a lot of time, which would be beneficial for everyone.

The goals when creating the improved workflow are the following:

- Make developers more satisfied with the new workflow (*Goal 1*)
- Decrease development time needed to create and configure animations (*Goal 2*)
- Create a way to allow designers to directly fine-tune animations (*Goal 3*)

If the improved workflow is successful it would also probably be possible to expand this to other use cases such as configuration of other design-related things directly on the mobile device.

1.4 Delimitation

This project is only concerned with trying to improve the workflow in Bontouch when practically developing and fine-tuning animations. A working, but possibly inefficient, workflow is assumed to already exist at Bontouch. Only this existing workflow is relevant for this project, and any reasoning about the bigger

organizational way of working in Bontouch is not included in the scope of this project.

The actual implementation in this project will be a small module where only a specific set of parameters can be configured. No other development of projects at Bontouch will be performed.

2 Method

In this chapter the method is explained, i.e. how the relevant information is gathered, how the implementation is performed and how the results are measured.

2.1 Interviews

First of all the relevant background information needs to be gathered from Bontouch in order to understand the current workflow at Bontouch, and what the eventual problems are. The definition of a workflow can be seen in Figure 1. Information about the workflow at Bontouch is needed in order to find out how the developers work and more specifically what kind of development workflow they have and what technical tools they are using. Getting to know how animations are created and configured is also important. This both includes the actual workflow when creating the animations, and also what type of parameters which are usually tweaked and fine-tuned.

Workflow:

“The sequence of industrial, administrative, or other processes through which a piece of work passes from initiation to completion.” [3]

Figure 1 - Definition of workflow.

A good way of gathering this information seems to be through the use of semi-structured interviews [4]. These kind of interviews typically contain a few set of questions to be begin with, but then allow for other questions too, depending on what the interviewee answers. The interviews in this case will be performed together with developers who are working with animation development at Bontouch, since they are the ones who can provide information about the current development workflow.

Semi-structured interviews are helpful in order to get a deeper understanding of the workflow at Bontouch. It is not suitable however to use only these interviews to try and measure and reason about the goals of this project. Daniel W. Turner III writes the following in *Qualitative Interview Design: A Practical Guide for Novice Investigators* [5]:

“Interviews provide in-depth information pertaining to participants’ experiences and viewpoints of a particular topic. Often times, interviews are coupled with other forms of data collection in order to provide the researcher with a well-rounded collection of information for analyses.”

It does thus seem reasonable to combine the interviews with another way of measuring the goals of this project, which is why surveys will also be performed.

2.2 Surveys

In order to measure various data points relevant to the goals of this project and what the developers working with animations think about various different workflows, two surveys in the form of questionnaires [6] will be performed.

The first survey will be performed with the developers before they have tried any new alternative workflow for developing animations. This survey is thus concerned with the current status at Bontouch and how they are usually working when developing animations.

The second survey will then be performed after the developers have tried a new alternative workflow for developing animations. The new workflow is the workflow that will be developed and evaluated in this project.

The goals of the surveys are to get enough data to be able to reason about whether or not the goals of this project have been fulfilled. The goals that will be measured are whether or not the developers are more satisfied with the new workflow, if the development time needed to create and tweak animations is decreased and if non-developers now can tweak animations themselves. The surveys will be performed electronically, since this provides a fast way of setting up the survey and then letting people answer it. The target group, developers at Bontouch working with animations, all have a high technical proficiency, which means that they will have no problem answering a survey online.

At least 5 developers will be asked to answer the pre-survey, try out the new alternative workflow and then finally answer the post-survey. This amount of test users should generate conclusive results, according to *Why You Only Need to Test with 5 Users* [7] by Jakob Nielsen.

To determine whether or not the difference in time estimates is statistically significant, the Paired t test [8] will be calculated. The Paired t test is relevant to use, since there are two samples with corresponding pairs in this case. Each developer will give two time estimates, one estimate before trying out Configuranimation [9] and one estimate after trying out Configuranimation. Configuranimation is the name of the prototype developed later in this project. If the p-value is below 0.05 when calculating the Paired t test, there is a statistically significant difference between the samples. The calculations will be made with the help of an online calculator [10].

2.3 Prototype

In order to be able to demonstrate a new workflow to the developers at Bontouch, it is also suitable to develop some kind of prototype. The goal of this prototype is to demonstrate a better workflow when developing and configuring

animations, for example by being able to configure animations directly on the mobile device. This prototype will be demonstrated to the developers as a new alternative workflow, after which it is possible for them to answer the post-survey described above.

2.3.1 Development tools and strategies

In this part the Android API documentation was researched together with 2 projects similar to what the prototype is supposed to achieve. The book Practical API Design [11] by Jaroslav Tulach was also consulted, in order to learn what to consider when creating a project which will be used by other developers.

2.3.1.1 Android API documentation

According to the Android API documentation [12], it is possible to create and configure animations both in code and by using XML files. Since everything is configurable in code it also means that it is possible to hook the configuration and do it directly on the mobile device. Assume that a parameter can be configured in code, like in Figure 2.

```
int animationDuration = 100;
```

Figure 2 - Manual assignment of variable.

Then it is also possible to configure it like in Figure 3.

```
int animationDuration = readLatestDuration();
```

Figure 3 - Assignment of dynamically retrieved value to variable.

This allows for the creation of a library where it is possible to directly configure parameters on the mobile device. The latest value of each parameter can then be read each time an animation is run for example. This is in practice how animations are configured right now too, just that the value is always set in the code.

In the API documentation several different ways of doing animations are mentioned, of which the most recent one is Property Animation [2]. It was introduced in Android 3.0 (API level 11) and allows for properties of any object to be animated. Two useful classes are *ValueAnimator* which calculates values for properties to be animated, and *AnimatorSet* which allows multiple animations to be grouped together. The animations can then be run in parallel. These classes will be used extensively during the development of the prototype. In the documentation of *ValueAnimator* both the duration and type of interpolators are mentioned as typical parameters to be changed.

2.3.1.2 Practical API Design by Jaroslav Tulach (2008)

Practical API Design [11] is a book written by Jaroslav Tulach, the main founder and architect behind the NetBeans IDE. In the book Tulach describes best practices and the theory behind creating a good and well-designed API, which he learnt after spending many years of trial and error developing NetBeans. He recommends treating the creation of normal classes as a kind of API design. This is in order to ensure that they will be usable by other developers for a long time. There is a lot of information in this book, which will be further studied during the implementation phase of this project. Even though the creation of a well-made API is not the main focus of this project, it is important to consider since it is going to be used by other developers too.

2.3.1.3 Similar projects

Tweaks [13] is a library created by Facebook. It is used for tweaking various parameters directly on iOS devices during development. The parameters to be tweaked are marked in the code, after which a debug menu is automatically generated. This is then available directly on the device, and is enabled by shaking the phone. Examples of useful parameters that can be tweaked are "animation timings, velocity thresholds, colors, and physics constants" [13]. Facebook apparently made heavy use of this during the development of the animation-heavy Paper app [14]. The idea of marking parameters in code and then showing a debug menu in the app will be used when developing the prototype in this project.

Mirror [15] is a library created by jim Labs, which allows for live previewing on physical devices when doing design work in Android. It automatically updates the design on the physical device when making changes in the code in the IDE on the computer. This can be useful when doing design work. Some drawbacks of Mirror is that it is not free, and a lot of extra code is needed to enable the live previews. To use Mirror to configure the design of the app, the relevant parts has to be marked in the code on beforehand. This idea will also be used in the development of the prototype in this project.

2.4 Scientific result

When the development of the library is done the average development time of an animation when using and not using this prototype will be measured by the surveys. The surveys will also ask whether or not the developers are satisfied with the workflow when using the prototype. The expected result is that it will be possible to create a library which makes it possible to decrease the development time needed to create and configure animations.

3 Implementation and results

The work is performed in 3 main phases, which are explained in this chapter. In the first phase the current situation at Bontouch is investigated, in the second phase a new prototype is developed and in the third phase the prototype is evaluated.

3.1 Phase 1 - Current situation at Bontouch

The main focus of the first phase is to assess the current situation at Bontouch, and what workflow they are currently using when developing and tweaking animations. A pre-survey is also performed in order to get quantitative data to later enable a comparison of the current workflow and the new alternative workflow.

3.1.1 Interviews

As described in the method chapter, performing an interview is a good way of getting in-depth information about a participant's experience and viewpoint of a subject. The aim of these initial interviews are to get to know the current workflow at Bontouch and whether or not the developers are satisfied with the current way of working.

Since Bontouch is a small company where developers often move around between different projects, only two developers were selected for these initial interviews. The reasoning behind only selecting two developers for the initial interviews is that many of the developers at Bontouch have knowledge about many of the current projects and should then be able to provide all the needed information about the current workflow.

The interviews were performed in-person as semi-structured interviews. First the questions below were asked, and then the interviewee was allowed to freely talk and describe the current workflow, how much time is needed to complete various tasks and anything else worth mentioning. Keywords and important sentences were then written down by the interviewer on paper, after which it was converted into fully written text.

The questions that were asked in the beginning of the interviews to get the interviews started can be seen in Figure 4.

- What is your current workflow when developing and configuring animations?
- Are you satisfied with this workflow?
- What would be an improvement to this workflow?

Figure 4 - Initial questions during interview.

3.1.2 Current workflow at Bontouch

The following information was obtained after interviewing two of the developers at Bontouch in order to find out how they work and what the current limitations of their workflows are.

Since this report is only concerned with the workflow when developing and configuring animations only two main actors are considered whenever discussing workflows. The two actors are a designer and a developer. The two roles might not always be clear-cut, and sometimes the developer assumes the role of a designer and vice-versa. But for the sake of this discussion the roles are assumed to be clear-cut. In that case the designer's job is to create the design of an animation, after which the developer implements this animation in the app using actual programming code.

A typical workflow at Bontouch is a process where the designer creates a design using images and sometimes videos, which the developer then has to re-create in the app. When there is a video available the developer tries to recreate the animations and designs from the video using methods available in the Android framework. But it is usually impossible to fully imitate the animations shown in the video. In the case where the developer only has a set of images describing the design, the developer has to personally think up and create all the animations.

When the developer has created the initial version of the animation, the fine-tuning and tweaking phase starts. The developer then does changes to the animation in the code, compiles the app and then walks over to the designer to show the new changes. The designer then provides feedback to the developer, who then proceeds to do new changes until both parties are happy with the design. This workflow can be seen in Figure 5.

1. designer creates a design and gives to developer
2. developer configures the animation in the code
3. and compiles the code
4. and uploads it to the smartphone
5. and navigates to the correct view in the app
6. and evaluates the animation and receives feedback from designer (is the speed good, how does the animation curve look etc.)
7. developer repeats from step 2 until animation looks good

Figure 5 - Current workflow at Bontouch.

Having to work through all these steps is a time-consuming task which many developers waste a lot of time performing. There is thus a need to find a faster and more convenient way of developing and configuring animations. One of the interviewed developers said that in a recent project at Bontouch a fairly

advanced animation took 20 roundtrips to the designer until the animation was finished and deemed good looking. This whole process took 2 full days.

There are also two other common workflows. The first is a workflow where the developer finds inspiration in other apps and imitate parts of their animations. In this case the designer is not involved, but much time is still spent doing the actual configurations of the animations.

The last workflow consists of the developer personally coming up with ideas for animations, after which they are implemented and tweaked. This could also be faster than going to the designer for feedback, but the feedback loop when practically configuring the animation is still too long.

Currently the fine-tuning of animations is usually done by the developer directly, since the designer usually does not have the necessary technical skill to be able to change animation properties directly in the code. The designer usually creates a design in some external program and then hands over the design to the developer for development and tweaking. The designer then has to provide feedback to the developer in order to tweak or change anything in the created app. This process takes time, which is why it would be very beneficial if there was a way for the designer to directly tweak the animations in the final app. A shorter feedback loop would both save time and allow the developer or designer to directly compare differences between different animations. Since the current workflow is so slow it is difficult to remember the difference between different animations, meaning you do not even know if the change was beneficial or not.

Both the interviewed developers said that it would be good if there was a way of configuring animations directly on the mobile device. One of the developers also said that he had already developed such a method in an earlier project, but that it was very specific for that project only. He also said that it would be better with a more general method which can be used in any project.

During the interviews it was apparent that the developers at Bontouch were not satisfied with the current workflow. Both of the developers agreed that it would be better to directly configure animations on the mobile devices, and that this would shorten down the development time considerably. Current estimates of development time of an animation is all the way up to 12 hours.

3.1.3 Pre-survey questions

The pre-survey questions were asked to the developers before they tried out the new alternative workflow. The questions were chosen because they, if asked before and after trying out the new workflow, allow for easy comparison of the two workflows. By collecting results concerned with these questions it is possible to reason about whether or not the goals of this project are fulfilled. The

following are the questions chosen for the pre-survey. Similar questions were also asked in the post-survey.

- Question 1. Are you satisfied with the current workflow when configuring animations?
- Question 2. Do you think it would be better to configure animations directly on the mobile device instead of in the code?
- Question 3. How much actual development time in average is spent per animation?
- Question 4. How many times do you estimate you have to contact the designers during the development and configuration of an animation?
- Question 5. Do you think it would be practical if the designer could fine-tune animations directly on the mobile device?

3.1.4 Method of asking the questions

A total of 6 developers were selected to evaluate this project, which meant that they had to answer the pre-survey questions, try out the prototype and then answer the post-survey questions. This number of developers was chosen in accordance to the theory presented in the method chapter.

All of the participating developers actively work with Android development and configuration of animations in their daily work. They are all experienced in the upsides and downsides of the current workflow and how much time various tasks usually take when developing and configuring animations.

The pre-survey questions were asked by performing a digital questionnaire, as seen in Appendix A. The survey was performed in Swedish, after which the results were translated into English. Question 1, 2 and 5 could either be answered with a "Yes" or a "No". In question 3 a time estimate in hours could be given, and in Question 4 a number estimate could be given. The post-survey was answered in the same way. When answering the pre-survey the developers only had a little bit of information about what this project was aiming to achieve, and they all answered the survey without needing any assistance or asking any questions about how to answer the survey.

3.1.5 Pre-survey results

When summarizing the results of the pre-survey, it was obvious that the developers were not satisfied with the current workflow when developing and configuring animations. A total of 5 out of 6 developers answered in pre-survey question 1 that they *are not* satisfied with the current workflow. They all agree that it would probably be better if it was possible to configure and fine-tune animations directly on the mobile device, and it would also be good if the designer directly could fine-tune the animations on the mobile-device. For the complete answers to all the questions, see Appendix C.

3.2 Phase 2 - Development of a prototype

In this phase of the project, a prototype which makes it possible to try a new workflow was developed. The aim was to demonstrate the new prototype and workflow to the developers, after which they could answer the post-survey questions.

The prototype was developed with respect to what the developers said were lacking in the current workflow, and what the developers think would be a better way of working. All development was done in Android using Android Studio [16] and the Java programming language [17].

3.2.1 Development of the prototype

The prototype is implemented as a modular library where the goal is for it to make it very easy to integrate in an already existing project. The goal of the library is to provide a usable and fast way of configuring animations directly on the device, thus hopefully improving upon the current workflow when developing and configuring animations.

The final library is called *Configuranimation*, which is a combination of the words *configuration* and *animation*. The author did all the programming and wrote all the code, while the developers at Bontouch provided constant feedback during the development. After each new feature was developed, the developers at Bontouch tried and evaluated the feature and then provided feedback. This feedback could then be used to improve or discard the feature, in order to finally end up with a usable implementation. All development was done in a methodical way, with development done using version control [18] and repeated test releases.

Consideration was also given to the theory in Practical API Design [11], especially where Jaroslav Tulach mentions that it is important to treat the creation of public classes as a form of API design. The public API of *Configuranimation* is designed to be as simple and straightforward as possible to use, without the need to constantly consult a difficult-to-read documentation.

Configuranimation is meant to be used in an already existing Android project, and is released as an .aar file [19]. This format is a binary distribution of an Android Library project, and is meant to be as easy as possible to simply drop into an existing Android project.

3.2.2 Technical result

Configuranimation provides a way to easily experiment with different values of any numerical parameter, directly on the mobile device. It is also possible to choose between a given set of Interpolators. This is achieved by the developer adding the parameters to be configured, directly in the code. After adding the parameters in the code, *Configuranimation* dynamically populates a list with a

combination of seekbars or drop-down selectors, depending on what type of parameters are to be configured. This list is then shown in the GUI of the app, where it is possible for the developer to adjust the values of the parameters using the seekbars or drop-down selectors.

3.2.2.1 Possible parameters to configure

Any parameter in the form of any of the following Java Number [20] classes *Float*, *Double*, *Integer* or *Long* can be configured using *Configuranimation*. Each parameter is named when it is created, and a minimum and maximum value also has to be specified. It is also possible to optionally specify the starting default value, and the step size of a parameter. Adding a numerical parameter to *Configuranimation* creates a seekbar in the GUI, as circled in Figure 6. By moving the seekbar in various directions, the underlying value of the parameter changes after which the new value can be used in any way desired. It can for example be used as a new value for the duration when running an animation, thus giving the developer a convenient and fast way of finding a value of the parameter which produces a satisfying animation.

After each adjustment is made, the new value of the parameter can be retrieved using the name that was used to create it. This value can then be used in any way the developer wants, but typical use cases are updating a view with new width and height values, repositioning views or updating the duration or delay of animators. It is also possible to associate a callback with each parameter, which will be run each time the parameter is updated in the GUI. This makes it possible to see the changes in real-time.

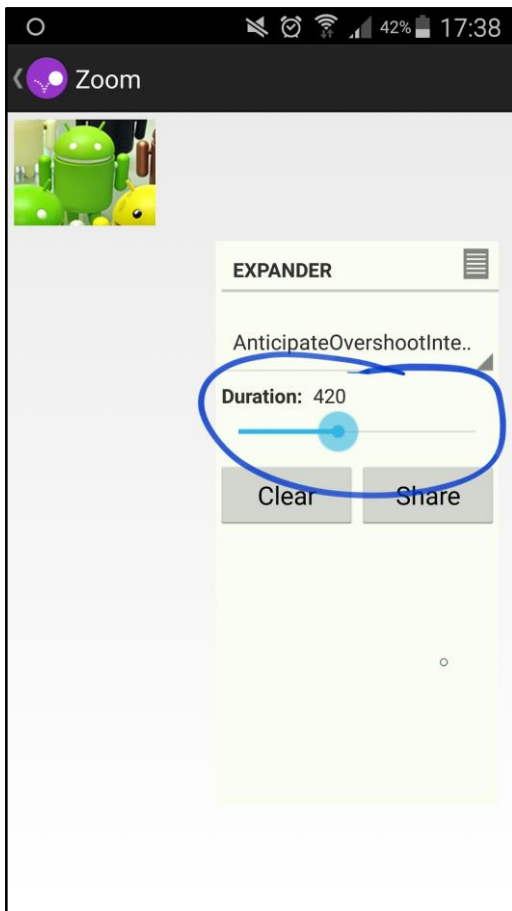


Figure 6 - A seekbar (circled for emphasis) in the GUI which is connected to an underlying number parameter.

Interpolator parameters can also be configured in *Configuranimation*, i.e. what type of interpolator to use when running the animation. There are multiple types of interpolators, but two common ones are the *linear interpolator* and the *accelerate interpolator*.

Assume that a view is to be moved from point A to point B during a fixed amount of time.

- If the move of this view is animated using a *linear interpolator*, the view would simply move in a constant speed from point A to point B.
- If however an *accelerate interpolator* was used to move the view, the view would start moving with a slow speed but accelerate more and more the closer it got to point B.

When adding an interpolator parameter the only thing that has to be specified is the name. Optionally the default value can also be specified. A drop-down selector will then be added to the GUI where it is possible to select any interpolator from the list of available interpolators in Android, see Figure 7. The selected type of interpolator can then be used to run the animation.

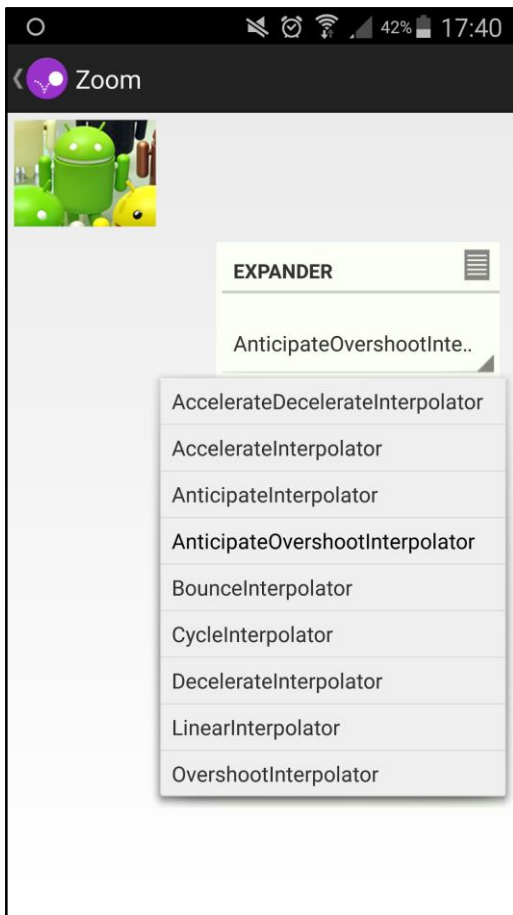


Figure 7 - An interpolator being chosen in the GUI from a list of available interpolators.

3.2.2.2 Download and documentation

Configuranimation can be downloaded at <http://osvald.org/exjobb/configuranimation-release-0.8.aar> [9]. The automatically generated Javadoc documentation [21] for Configuranimation is hosted online at <http://osvald.org/exjobb/docs> [22], since it would be too big to attach to this report.

3.2.3 Example of usage of prototype

The following is a complete example of usage of Configuranimation, meant to demonstrate a new workflow when configuring animations. In the example, the app consists of an image thumbnail (Figure 8), which can be clicked (Figure 9), after which it expands (Figure 10). Both the thumbnail and expanded image have many different values which can be configured. In this case the developer

has already received a design from the designer and implemented a basic, but not yet good-looking, animation.



Figure 8 - Thumbnail before expansion.

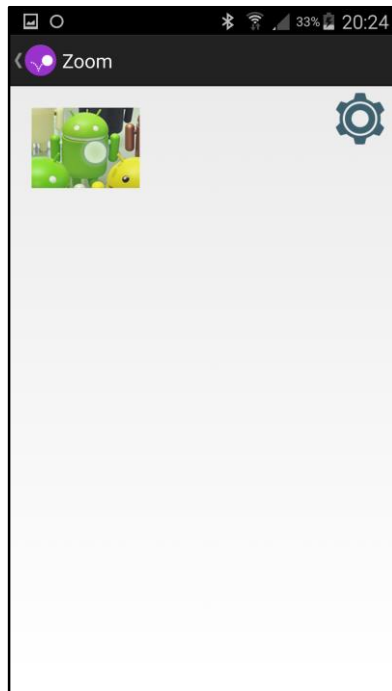


Figure 9 - Thumbnail getting clicked.

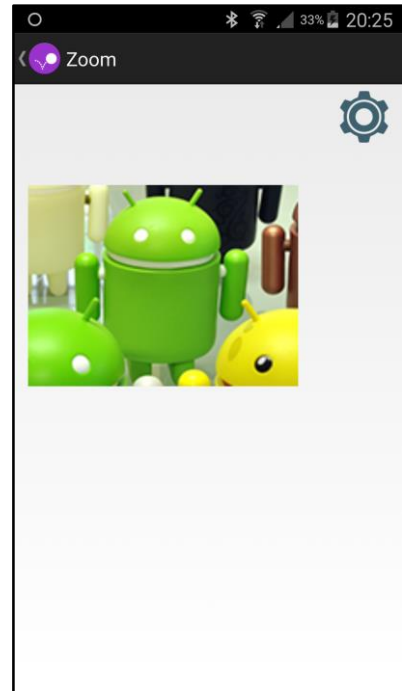


Figure 10 - Thumbnail after expansion.

Since the basic animation has already been implemented, it is assumed that an expansion animation object called “*expansionAnimator*” of type *Animator* [23] exists, which was created before using this tool. The *Animator* object is what a developer usually uses to create an animation, and the object has methods for setting various properties of the animation. The methods that will be used here are *setDuration(duration)* and *setInterpolator(interpolator)*, which simply configures how long duration and what type of interpolator to use when running the animation. To actually run the animation, there is also a method called *start()*.

These are some of the most common parameters to configure:

- position (x and y coordinates)
- width and height of original view
- width and height of view after the animation is run
- duration of animation (when expanding or closing view)
- delay before running animation
- what type of interpolator to use

All of these could easily be configured using *Configuranimation*, but here only the duration of the expansion animation and type of interpolator will be

configured. The following is then a complete example of all the steps needed to configure these two parameters:

1. Add `configuranimation.aar` [9] to the existing Android project.
2. Instantiate a `Configuranimation`-object in the `onCreate` [24] method of the current Android Activity, see Figure 11.

```
Configuranimation configuranimation =  
    new Configuranimation(this); // this == current Activity
```

Figure 11 - Object instantiation.

3. Add parameters to be configured, see Figure 12.

```
final AnimationConfig animationConfig = configuranimation.addAnimationConfig("Expander")  
    .addLongProperty("Duration", 0, 1000)  
    .addInterpolator("Interpolator");
```

Figure 12 - Creation of grouped properties.

This creates a named group of properties called "Expander" with a long property with values between 0 and 1000 called "Duration" and an interpolator property called "Interpolator".

4. Add a callback to be run each time the properties are changed in the GUI, see Figure 13:

```
animationConfig.setCallback(new Callback() {  
    @Override  
    public void run() {  
        animator.setDuration(animationConfig.getLongProperty("Duration"))  
            .setInterpolator(animationConfig.getInterpolator("Interpolator"));  
        animator.start();  
    }  
});
```

Figure 13 - Setting of callback.

Each time the callback is run it updates the duration and type of interpolator of the animator, after which it is started. This means that the animation will be run with the new values after each update of the property.

5. Compile and start the app.
6. Make changes in the GUI using the seekbar and the drop-down selector, and observe the results. See Figure 14 for example.

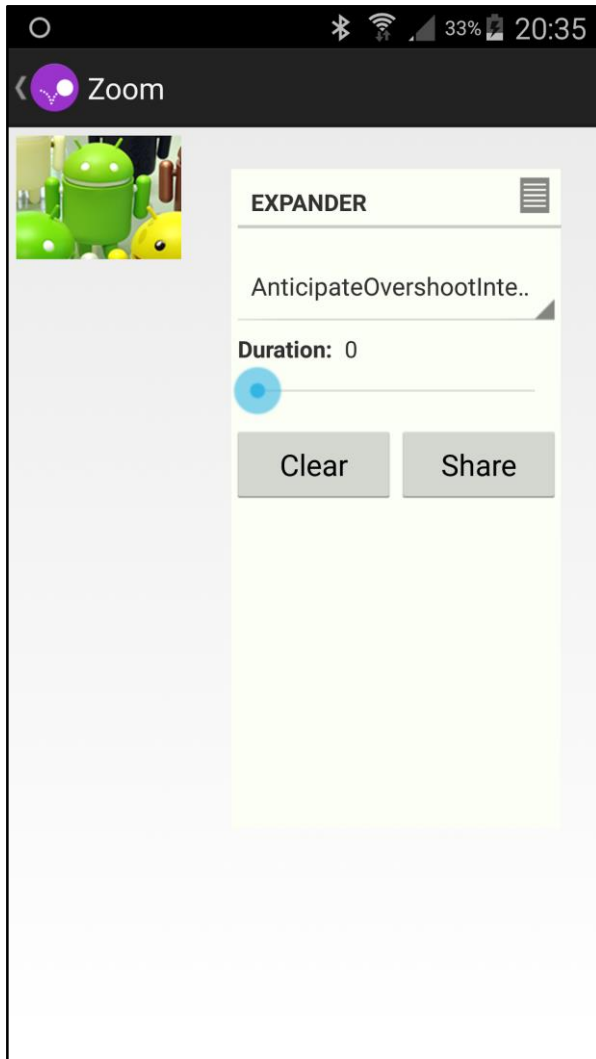


Figure 14 - Expanded configuration menu showing two configurable items, an interpolator and duration.

Repeat this until satisfying values are found for the duration and the type of interpolator. In this example the values might end up being 700 milliseconds for the duration, and LinearInterpolator for the interpolator, or any other valid value (final values not shown in any figure).

7. Use the final values to directly configure the animator in the code.
8. Delete the code of Configuranimation that was added in the previous steps, also delete configuranimation.aar from the Android project.
9. The app can now be recompiled and used with the final good-looking values of the parameters found using Configuranimation.

3.3 Phase 3 - Evaluation of prototype and new workflow

In this final phase the prototype is evaluated by the developers after which they answer a post-survey concerned to determine whether or not a workflow when using the prototype is more or less satisfying than the old workflow. The development time needed to develop and configure animations when using the old workflow is also compared to this new alternative workflow.

3.3.1 Evaluation of prototype

All the developers who answered the pre-survey also evaluated the prototype and finally answered the post-survey. When it was time to evaluate the prototype, each developer received access to a page (in Swedish, since they all speak Swedish) describing how to download and test the prototype. The page describing the test can be found in Appendix D.

The developers were not given a specific task to perform, but included in the page was an example of a configuration of an animation. It is assumed that they looked at this example and used the prototype to configure an animation of their own, which they also confirmed that they did in informal discussions after the test. All of the developers performed the test alone, in their own pace using their own computers. Only one developer asked for help, since he considered the example in the description page to be too vague. The other developers tested the prototype without needing any help. After practically testing the prototype the developers answered the post-survey as described below.

3.3.2 Post-survey questions

The post-survey was, just like the pre-survey, performed as a digital questionnaire, see chapter 3.1.4 for full description of the method. The questions asked are shown below, and the full survey as the developers saw it can be seen in Appendix B. The survey was performed in Swedish, after which the results were translated into English.

- Question 1. Are you satisfied with the new workflow when configuring animations?
- Question 2. Is it better to configure animations directly on the mobile device instead of in the code?
- Question 3. How much development time in average is spent per animation when using Configuranimation?
- Question 4. How many times do you estimate that you have to contact the designers during the development and configuration of an animation using Configuranimation?
- Question 5. Do you think it would be practical if the designer could fine-tune animations directly on the mobile device using Configuranimation?

3.3.3 Result of post-survey

When gathering and summarizing the results of the post-survey, it was immediately clear that the developers were more satisfied with the workflow when using the prototype presented in this project (Question 1). All 6 developers were satisfied with the workflow when developing and configuring animations, as compared to earlier when only 1 out of 6 was satisfied.

They all also agree that it is better to configure animations directly on the device (Question 2) and that this also probably makes it practical for the designer to directly fine-tune animations on the device (Question 5).

All the developers also answered that the estimated time needed to develop and configure an animation decreased when using the prototype and new workflow (Question 3). In Figure 15 the estimated time needed when using and not using the current workflow is shown.

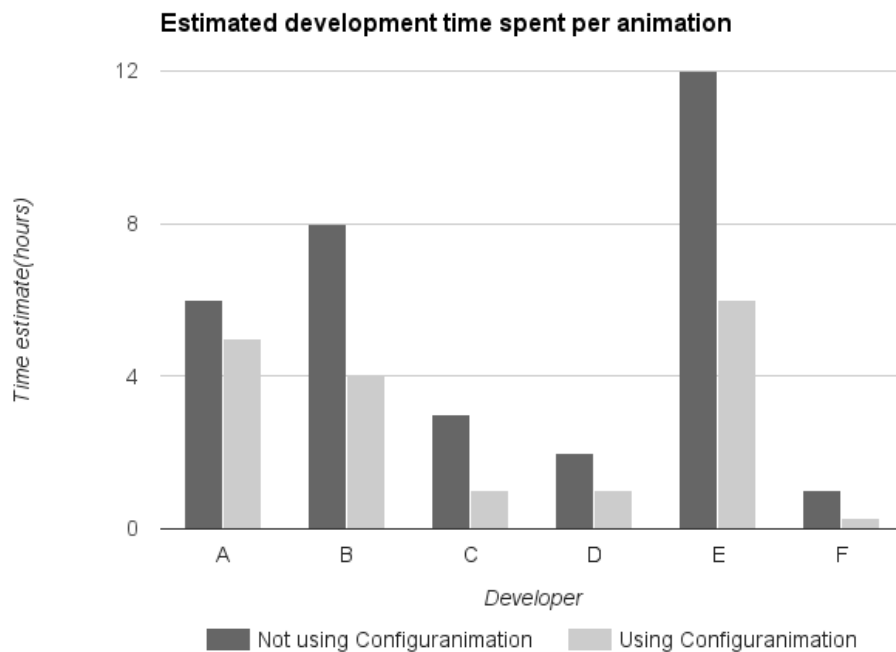


Figure 15 – Estimated development time of an animation when using and not using Configuranimation.

By calculating the p-value of the Paired *t* test using GraphPad Software’s online calculator [10], a p-value of 0.0366 is retrieved. Since this is below 0.05, the decrease in estimated development time is of statistical significance.

The developers also reported that the number of times needed to contact the designer in order to get feedback during the development and configuration of an animation decreased, as seen in Figure 16.

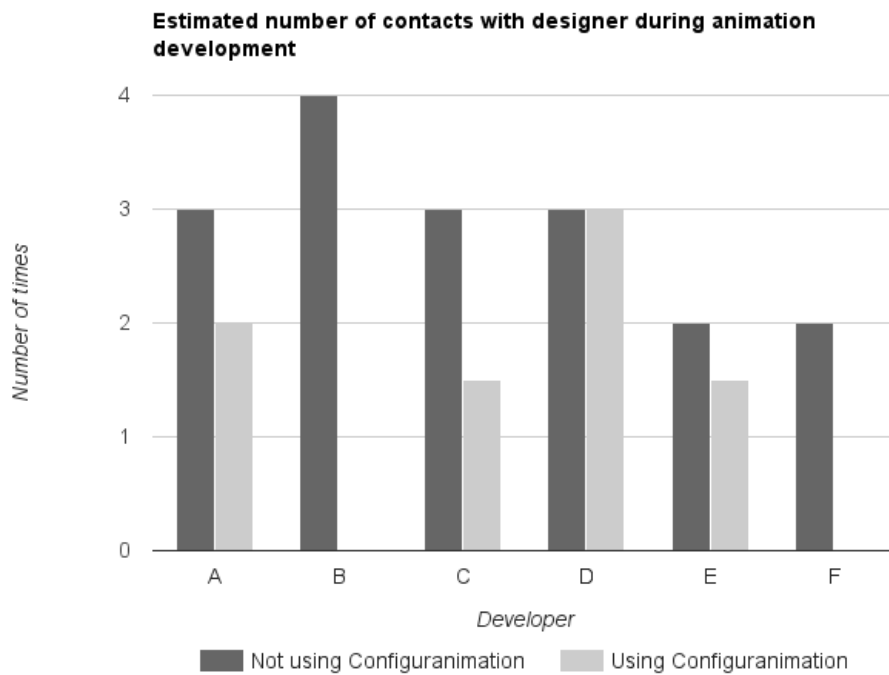


Figure 16 - Estimated number of times needed to contact designer when using and not using Configuranimation.

By using the same calculator [10] as above, the obtained p-value of the Paired t test for the estimated times needed to contact the designer is 0.0484. This means that this result too is of statistical significance.

For the complete answers to all the questions, see Appendix C.

4 Discussion

In this chapter the results are discussed and interpreted.

4.1 Higher satisfaction with the new workflow

In the pre-survey, only 1 of the 6 developers was satisfied with the current workflow when developing animations as answered in Question 1. But after trying out Configuranimation a total of 6 out of 6 developers answered in the post-survey Question 1 that they were satisfied with the new way of configuring animations. After trying out the prototype one of the developers said the following about the project: “[Configuranimation is a] very nice and useful tool which I definitely will continue using”. Another developer wrote the following in an email after trying out the prototype: “It’s really good and I’m convinced that it will be really useful when developing [animations].”

This is very good feedback, and Goal 1 as outlined in the Problem chapter is definitely achieved.

No survey was made to measure *how much less* satisfied the developers were with the old workflow when developing and configuring animations, and *how much more* their satisfaction improved when using the new workflow.

Developing and configuring animations is also only a small part of a much bigger development job, which in turn is just a part of the job satisfaction when working in a development company. But if using Configuranimation makes for a more satisfying workflow, it hopefully means that the developers in the end feel higher job satisfaction at work too, even if it is only a small increase. And having higher job satisfaction is very important, since most people spend at least 8 hours per day at work every workday. Therefore it is important to work using as good workflows and tools as possible.

4.2 Decrease of development time and number of contacts with designer

As seen in Figure 15, every developer also reported that the time needed to develop and configure animations decreased when using Configuranimation. The number of times needed to get feedback from the designer also decreased (Figure 16), since the developers could tweak the animations and get better results in a faster way than earlier. These are positive results, since it means that this new workflow actually saves the developers time when creating and configuring animations. Both of these results were determined to be of statistical significance, as seen in chapter 3.3.3.

A decrease in development time needed and also time spent contacting the designer means that both the developers and designers will have more time to focus on what is really important. In the case of the developer it will be possible

to create better-looking animations faster than before, thus giving more time to other tasks like creating more animations or working on other features of the app. And in the case of the designer less time will be spent giving feedback to the developer, and more time can be spent creating more beautiful designs.

Since the estimated development time needed to configure animations was decreased, Goal 2 was also achieved.

4.3 Possibly allowing designers to directly fine-tune animations

In the post-survey, all the developers answered that they believe that it would be practical if the designer could fine-tune animations directly on the device (Question 5). As outlined in the Problem chapter, Goal 3 was to create a way to allow designers to directly fine-tune animations on the device. However, this was never tested, so Goal 3 cannot be considered achieved in this project. It seems likely that the prototype can successfully be used by the designers, but an investigation of this is left to eventual further work.

4.4 Many configuration possibilities

During the testing phase of the project multiple developers reported that they used Configuration to configure other parameters not even concerned with animations. One developer used it to configure the depth of a button, since the live feedback meant that he could simply use the seekbar to find a good-looking depth in a quick manner.

Allowing for the configuration of any parameter while also greatly reducing the feedback loop is a very important consequence of using this new workflow of configuring parameters directly on the mobile device. This allows the developer to instantly receive feedback when making small changes to an animation for example. Assume that the feedback loop normally takes 2 minutes, meaning that it takes 2 minutes before you see the result of a change. And the task is to change the duration of an animation from 200 milliseconds to a new value, say 180 milliseconds or maybe 240 milliseconds. With a feedback loop of 2 minutes it is almost impossible to even remember the differences between these durations, but when getting live feedback directly on the device it is fully possible.

But since any parameter can be configured directly on the device, not only the duration and interpolator of an animation, this result can be used in many more different ways. It is possible to use it to quickly try out many different variants of animations directly on the device, making it possible to play around with different versions and getting a good feel for what type of animation feels good for the user. It is also possible to expand it in order to be able to try out other design-related changes on any type of element in the GUI, since it is not locked to animation use only.

5 Conclusion

In conclusion the new workflow when using Configuranimation as proposed in this project turned out to be very useful to developers who are doing development and configuration of animations on Android. With relatively simple measures it was possible to create a library which reduces the time of the feedback loop and thus reduces the development time in total. The developers who tried Configuranimation also agreed that it is a more satisfying way of configuring animations.

The development time needed to develop animations is now reduced. And by using Configuranimation it is possible to create better-looking animations than before in a faster way. It is also possible to freely experiment with the look-and-feel of apps directly on the mobile device, in order to quickly find what looks and feels good for the user. This allows the developers to focus even more on other crucial parts of the apps, like the business logic or other features.

6 Future work

There are many future improvements which are possible to make to this project. The below are some of the more important ones, in no particular order.

6.1 Further improvements to the workflow

6.1.1 Let designers test Configuranimation

As mentioned in the Discussion chapter, Goal 3 of this project was not achieved, since the prototype was never tested on the designers. The only result concerned with Goal 3 was that the developers answering the post-survey all thought that it probably would be a good idea to let the designers use Configuranimation to do the final fine-tuning of animations themselves. A real study should be made where the designers get to test the prototype and the total time needed to configure and fine-tune an animation is recorded. This could then be compared to the current workflow when creating and configuring animations.

6.1.2 Investigate completely different alternative workflows

In this report only one new workflow is suggested, which is to use the developed prototype to configure animations directly on the mobile device. But this is only one of many possible workflows, and there are probably even better ways of creating and configuring animations out there. For example, there might be a way of creating a workflow where the designer takes charge of the whole process of creating and configuring animations. It could be possible to automatically convert the designs created by the designer into working animations directly. This is a suggestion for something to investigate in future work.

6.2 Technical improvements to Configuranimation

6.2.1 More default sets of parameters

By adding more default sets of parameters, it would be even easier and faster for a new developer to include and start using this project. This could include various commonly used parameters like a default set of *start scale*, *end scale*, *animation duration* and *animation start delay*. By including this default set of parameters the corresponding configurable elements would automatically get included in the configuration GUI.

6.2.2 More flexible adjustments in GUI

Another useful feature would be the ability to configure the minimum and maximum values of the ranges of the parameter directly in the GUI. Instead of having to configure a minimum and maximum value in the code, this could be directly adjusted in the GUI. This is useful if the developer realizes that the initially configured range is too narrow for example.

6.2.3 Include Javadoc directly in packaged library file

Right now the packaged .aar-file of Configuranimation does not include the Javadoc documentation, since it is not supported in the aar file format. The documentation is hosted externally, but having it directly included in the library package and thus available in the IDE would be useful.

6.2.4 Open source the project

To open source the project and make it available to everyone would also be a good thing to do. This would both allow even more developers to use it and hopefully provide feedback in order to make the project even more useful.

7 References

- [1] Wikipedia, "Android (operating system)" 2008. [Online]. Available: [http://en.wikipedia.org/wiki/Android_\(operating_system\)](http://en.wikipedia.org/wiki/Android_(operating_system)). [Accessed 20 May 2015].
- [2] Android, "Property Animation | Android Developers" 2015. [Online]. Available: <http://developer.android.com/guide/topics/graphics/prop-animation.html#object-animator>. [Accessed 3 May 2015].
- [3] Oxford University Press, "Workflow - Oxford Dictionaries" 2013. [Online]. Available: http://www.oxforddictionaries.com/us/definition/american_english/workflow. [Accessed 14 May 2015].
- [4] Food and Agriculture Organization of the United Nations, "Semi-structured interviews" 1990. [Online]. Available: <http://www.fao.org/docrep/x5307e/x5307e08.htm>. [Accessed 10 June 2015].
- [5] D. W. T. III, "Qualitative interview design: A practical guide for novice investigators" *The qualitative report*, vol. 15, no. 3, pp. 754-760, 2010.
- [6] Wikipedia, "Questionnaire" 2005. [Online]. Available: http://en.wikipedia.org/wiki/Semi-structured_interview. [Accessed 12 May 2015].
- [7] J. Nielsen, "Why You Only Need to Test with 5 Users" 19 March 2000. [Online]. Available: <http://www.nngroup.com/articles/why-you-only-need-to-test-with-5-users/>. [Accessed 3 May 2015].
- [8] Wikipedia, "Student's t-test" 2015. [Online]. Available: http://en.wikipedia.org/wiki/Student%27s_t-test#Paired_samples. [Accessed 23 May 2015].
- [9] O. Ivarsson, "Configuranimation" 2015. [Online]. Available: <http://osvald.org/exjobb/configuranimation-release-0.8.aar>. [Accessed 13 May 2015].
- [10] GraphPad Software, "Paired t-test calculator" 2015. [Online]. Available: <http://graphpad.com/quickcalcs/ttest1/>. [Accessed 23 May 2015].
- [11] J. Tulach, *Practical API design: Confessions of a java framework architect*, Apress, 2008.
- [12] Android, "Animation and Graphics Overview" 2015. [Online]. Available: <http://developer.android.com/guide/topics/graphics/overview.html>. [Accessed 16 February 2015].
- [13] Facebook, "Tweaks" 2014. [Online]. Available: <https://github.com/facebook/Tweaks>. [Accessed 3 May 2015].
- [14] Facebook, "Paper" 2014. [Online]. Available: <https://www.facebook.com/paper>. [Accessed 3 May 2015].

- [15] jimu Labs, "Mirror" 2012. [Online]. Available: <http://jimulabs.com/>. [Accessed 3 May 2015].
- [16] Android, "Android Studio Overview" 2014. [Online]. Available: <http://developer.android.com/tools/studio/>. [Accessed 13 May 2015].
- [17] Wikipedia, "Java (programming language)" 2004. [Online]. Available: [http://en.wikipedia.org/wiki/Java_\(programming_language\)](http://en.wikipedia.org/wiki/Java_(programming_language)). [Accessed 13 May 2015].
- [18] Wikipedia, "Mercurial" 2005. [Online]. Available: <http://en.wikipedia.org/wiki/Mercurial>. [Accessed 13 May 2015].
- [19] Android, "AAR Format - Android Tools Project Site" 2015. [Online]. Available: <http://tools.android.com/tech-docs/new-build-system/aar-format>. [Accessed 3 May 2015].
- [20] Oracle, "Number (Java Platform SE 7) - Oracle Documentation" 2012. [Online]. Available: <https://docs.oracle.com/javase/7/docs/api/java/lang/Number.html>. [Accessed 13 May 2015].
- [21] Oracle, "Javadoc" 2004. [Online]. Available: <http://www.oracle.com/technetwork/articles/java/index-jsp-135444.html>. [Accessed 14 May 2015].
- [22] O. Ivarsson, "Configuranimation documentation" 2015. [Online]. Available: <http://osvald.org/exjobb/docs>. [Accessed 14 May 2015].
- [23] Android, "Animator | Android Developers" 2011. [Online]. Available: <http://developer.android.com/reference/android/animation/Animator.html>. [Accessed 13 May 2015].
- [24] Android, "Activity | Android Developers" 2015. [Online]. Available: <http://developer.android.com/reference/android/app/Activity.html>. [Accessed 3 May 2015].

Appendix A. Pre-survey questionnaire (in Swedish)

Undersökning innan test av Osvalds exjobb

Till Android-utvecklare:

Mitt exjobb (Real-time animation control) går ut på att skapa ett generellt verktyg för att under utvecklingsarbetet av Androidappar kunna finjustera animationsparametrar och övriga parametrar direkt på telefonen.

För att få ett vetenskapligt resultat till exjobbet får du gärna hjälpa mig genom att svara på frågorna nedan.

Tack på förhand,
Osvald

* Required

Är du nöjd med det nuvarande tillvägagångssättet för att utveckla och sedan finjustera animationer under utvecklingsarbetet av Android-appar? *

- Ja
 Nej

Skulle det vara bättre om man kunde justera parametrar direkt på telefonen? *

- Ja
 Nej

Hur mycket tid uppskattar du att du lägger på utvecklingen av en animation? *

Tänk dig att du ska göra en någorlunda avancerad animation. Hur mycket tid tar det ungefär från start till slut, inklusive finjusteringar och åtgärdande av feedback från t.ex. en eventuell designer?

Hrs ▼ : Mins ▼ : Secs ▼

Under utvecklingen av en animation, hur många gånger är du i kontakt med designern för att få feedback? *

Skulle det vara praktiskt om designern kunde göra finjusteringar av animationer själv, direkt på telefonen? *

- Ja
 Nej

Ditt namn *

(Kommer inte att användas/visas i resultatet någonstans)

Submit

100%: You made it.

Never submit passwords through Google Forms.

Appendix B. Post-survey questionnaire (in Swedish)

Undersökning efter test av Osvalds exjobb

Till Android-utvecklare:

Efter att ha laddat ned och testat mitt exjobb får du gärna svara på frågorna nedan!

Tack på förhand,
Osvald

* Required

Är Configuranimation ett bättre sätt att utveckla och justera animationer? *

- Ja
 Nej

Är det smidigare att justera animations-parametrar direkt på den mobila enheten? *

- Ja
 Nej

Hur mycket tid uppskattar du att utvecklingen av en animation tar om du använder Configuranimation? *

Tänk dig att du ska göra en någorlunda avancerad animation. Hur mycket tid tar det ungefär från start till slut, inklusive finjusteringar och åtgärdande av feedback från t.ex. en eventuell designer?

Hrs ▼ : Mins ▼ : Secs ▼

Under utvecklingen av en animation med hjälp av Configuranimation, hur många gånger är du i kontakt med designern för att få feedback? *

Skriv "ingen designer", om du inte har någon designer.

Skulle det vara praktiskt om designern kunde använda Configuranimation själv, direkt på telefonen? *

- Ja
 Nej

Övriga kommentarer

Ditt namn *

(Kommer inte att användas/visas i resultatet någonstans)

Submit

100%: You made it.

Never submit passwords through Google Forms.

Appendix C. Answers to pre- and post-survey

Are you satisfied with the current workflow when configuring animations?	Do you think it would be better to configure animations directly on the mobile device instead of in the code?	How much actual development time in average is spent per animation? (Hours)	How many times do you estimate you have to contact the designers during the development and configuration of an animation?	Do you think it would be practical if the designer could fine-tune animations directly on the mobile device?
No	Yes	6	3	Yes
No	Yes	8	4	Yes
No	Yes	2	3	Yes
No	Yes	12	2	Yes
No	Yes	3	3	Yes
Yes	Yes	1	2	Yes

Table 1 - All answers to the pre-survey.

Are you satisfied with the new workflow when configuring animations?	Is it better to configure animations directly on the mobile device instead of in the code?	How much development time in average is spent per animation when using Configuranimation? (Hours)	How many times do you estimate that you have to contact the designers during the development and configuration of an animation using Configuranimation?	Do you think it would be practical if the designer could fine-tune animations directly on the mobile device using Configuranimation?
Yes	Yes	4	0	Yes
Yes	Yes	1	3	Yes
Yes	Yes	5	2	Yes
Yes	Yes	6	1-2	Yes
Yes	Yes	1	1-2	Yes
Yes	Yes	0.3	0	Yes

Table 2 - All answers to the post-survey.

Appendix D. Page describing how to test prototype (in Swedish)

Test av Osvalds exjobb

Hej!

Mitt exjobb (Real-time animation control) går ut på att skapa ett generellt verktyg för att under utvecklingsarbetet av Androidappar kunna finjustera animationsparametrar och övriga parametrar direkt på telefonen.

För att hjälpa mig att få ett vetenskapligt resultat, skulle jag verkligen uppskatta om du kunde göra följande 3 steg:

1. Svara på följande frågor innan du testar mitt projekt (Configuranimation):
<http://goo.gl/forms/QnkFHx1vbv>
2. Testa mitt projekt enligt instruktionerna nedan.
3. Svara på följande frågor efter att du har testat mitt projekt:
<http://goo.gl/forms/elzhxLKw4w>

Instruktioner för att testa Configuranimation

Att lägga in Configuranimation i ditt egna Android-projekt är enkelt. Configuranimation låter dig välja vilka parametrar du vill kunna justera direkt i den dynamiskt genererade menyn som skapas. Se kodexempel nedan för hur du väljer vilka parametrar som ska kunna ställas in, samt hur du läser ut de igen senare.

1. Ladda ner senaste aar-filen här: Configuranimation 0.8
2. Lägg filen i libs-katalogen i Android-projektet.
3. Lägg till följande rader i build.gradle:

```
repositories {
    flatDir {
        dirs 'libs'
    }
}

dependencies {
    compile(name:'configuranimation-release-0.8', ext:'aar')
}
```

4. Lägg t.ex. till följande i onCreate:

```
Configuranimation configuranimation = new Configuranimation(this);

// Add what you want to configure like this
configuranimation
```

```

.addAnimationConfig("Expander")
.addDuration(0, 1000, 0, 20)
.addInterpolator()
.setCallback(new Callback() {
    @Override
    public void run() {
        expandView();
    }});

// Or maybe like this
configuranimation
.addAnimationConfig("Thumb Start")
.addFloatProperty("Start Scale X", 0f, 2f, 1f, 0.1f)
.addFloatProperty("Start Scale Y", 0f, 2f, 1f, 0.1f)
.addDpProperty("Start X", 0, 400, 0, 4)
.addDpProperty("Start Y", 0, 1000, 0, 10)
.setDebouncingEnabled(false);

// Retrieve values again
AnimationConfig config = configuranimation.getAnimationConfig("Thumb Start");

thumbView.setX(config.getDpPropertyInPx("Start X"));
thumbView.setY(config.getDpPropertyInPx("Start Y"));
thumbView.setScaleX(config.getFloatProperty("Start Scale X"));
thumbView.setScaleY(config.getFloatProperty("Start Scale Y"));

```

Dokumentation följer tyvärr inte med .aar-filen, men finns online på följande adress:
<http://osvald.org/exjobb/docs/>

5. Testa din app och justera de parametrar som du har valt att konfigurera. När du har hittat bra värden kan du sedan ställa in dessa direkt i din kod eller XML-filer och ta bort Configuranimation igen.

6. Efter att du har testat projektet får du gärna svara på de 5 korta frågorna på följande länk: <http://goo.gl/forms/elzhxLKw4w>

7. Känn dig bra för att du har hjälpt mig med mitt exjobb!

Tack för all hjälp! Om du undrar något eller om något är oklart så är det bara att höra av dig till mig!

Hälsningar,
Osvald

