A Community Website for Interactive Mobile Content Created by Children and Teenagers

Tobias Gritschacher
tobias@gritschacher.eu

Master's Thesis
Graz University of Technology

Adviser
Univ.-Prof. Dipl.-Ing. Dr.techn. Wolfgang Slany
wolfgang.slany@tugraz.at

STATUTORY DECLARATION

I declare that I have authored this thesis independently, that I have not used other than the declared sources / resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

…….. ……..

date (signature)

Abstract

The Catroid community website is part of the Catroid project[1] and provides a platform for all Catroid users to upload, exhibit, and share their own creations, and rate, comment, and remix the content of others. Catroid is a graphical, “LEGO”-style programming language for Android devices that allows children, teenagers, and users of all ages to easily and enticingly create animations and interactive apps directly on their smartphone, without the need of experienced programming skills or a PC. Catroid is inspired by the Scratch programming language developed by the Lifelong Kindergarten Group at the MIT Media Lab. Catroid is especially designed for mobile devices and will support a wide range of sensors available on those devices.

The Catroid community website has the goals to simplify the process of uploading, publishing, and remixing the content generated by users and to let an actively participating community of creators arise around Catroid.

This master’s thesis covers the early stages of development, various tools and methods which were used, and the current state of the community website project. It further describes how the success of such an online community can be measured, problems with remixing and how they can be solved, and gives future prospects based on literature research and best practice examples how the community website could evolve to maximize user participation and satisfaction.

[1] The Catroid project started in January 2010 and has grown until July 2011 to 57 active developers. In March 2011 Catroid was selected as a Google Summer of Code 2011 project.

Table of contents

1. Introduction

Young people let their creativity flow and create animations, music-videos, and even computer games and other applications, e.g., educational programs with their smartphones. They become creators of interactive content instead of just being consumers. They use their school breaks and spare time to let their ideas become visible and share these creations with their friends, class mates, and on social networks on the World Wide Web.

So the question is: can this scenario become reality? The answer is yes: it not only can become reality – it is already reality, at least to some degree. Back in 2004, the Lifelong Kindergarten Group at the MIT described a similar idea for PCs [Mal04]. It was the beginning of a successful programming platform called Scratch for young creators which today has over 830.000 registered users who uploaded more than 1.9 million projects to its community website by July 2011 [Lif11]. We want to take the next step and are developing a system similar to Scratch for mobile phones. Our system mainly consists of two parts – a mobile application which can be installed on an Android smartphone, and a community website where children can share their projects and are able to collaborate with others.

1.1. Catroid software

Catroid[2] is an on-device graphical programming language that runs on Android devices and that is intended for children. Catroid is inspired by the Scratch programming language developed by the Lifelong Kindergarten Group at the MIT Media Lab [Mal04]. As known from Scratch or Google App Inventor[3], Catroid programs are written in a graphical "LEGO"-style which makes it as easy as possible for children from the age of 8 to create their own Android apps without the need of having experienced programming skills. Nevertheless Catroid differs in important aspects from Scratch or App

[2] http://code.google.com/p/catroid

[3] http://appinventor.googlelabs.com/about

Inventor. There is no need for a PC – the apps can be written by solely using mobile Android devices. Compared to Scratch, Catroid will additionally also support special sensors which are only available on mobile devices. Figure 1 shows a demonstration of the software in action.

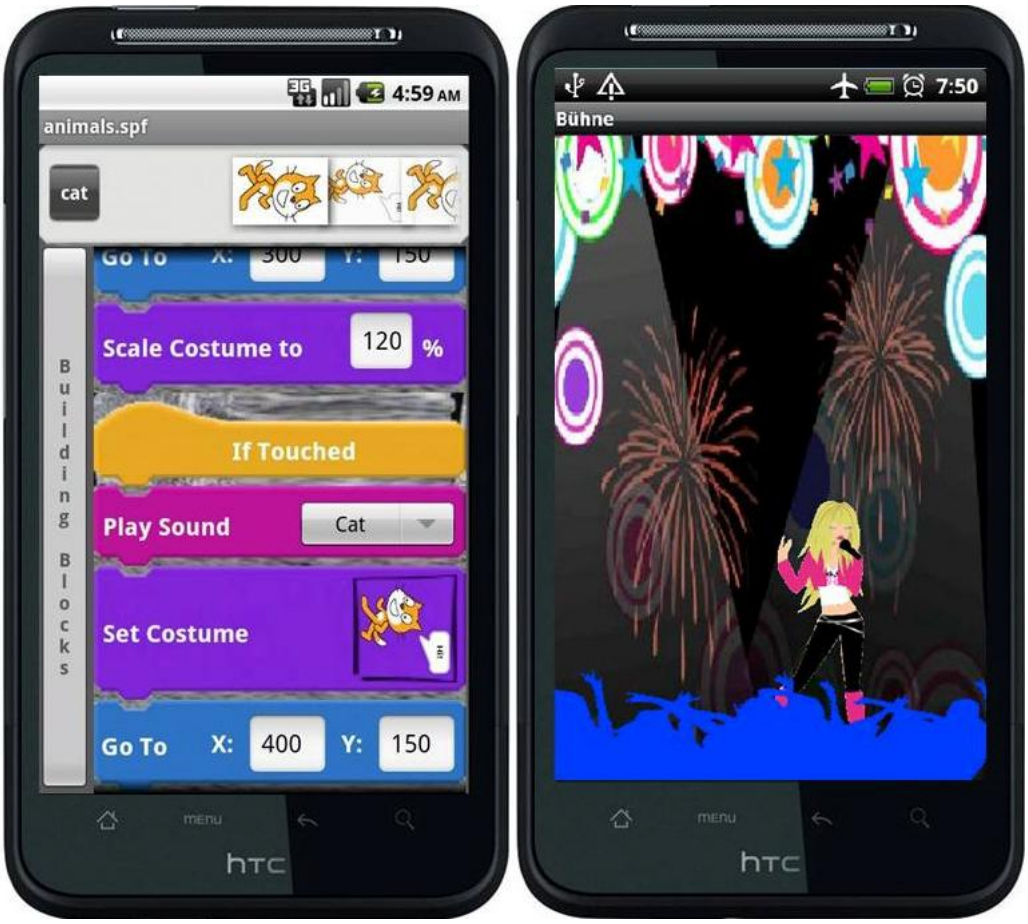

**Figure 1. Left: a project in development on the phone.
Right: execution of a Hannah Montana interactive music video animation created by children (remixed from an original Scratch project[4]).**

1.2. Community website

The Catroid project includes a community website[5] allowing children to upload and share their projects with others. All projects uploaded to the community website are open source and published under a free software license, though by default the license is restricted to non-commercial use.

[4] http://scratch.mit.edu/projects/tyster/443306

[5] http://www.catroid.org

Everyone can download and edit any particular project from the website, add new functionality or change the current behavior of the project, and upload the new version again. This is called “remixing” and was a core idea behind the Scratch online community [Mon09]. The principle of remixing, the problems associated with it, and possible solutions for those problems will be discussed in Section 6.4. The website also provides the possibility to rate and comment projects, to add other users as friends, to get updates on their activities, and it provides a wiki and a forum for discussion. See Figure 2 for some images of the community website on an Android smartphone.

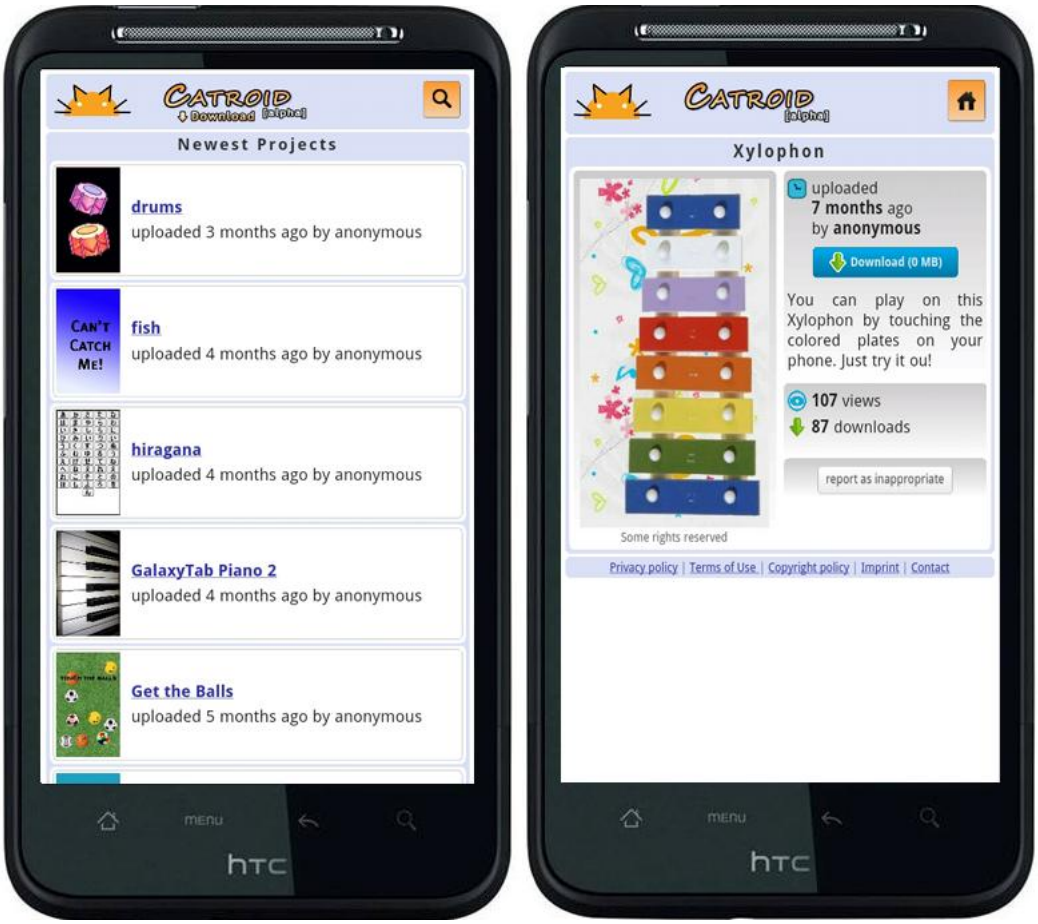

Figure 2. Left: a list of the newest projects on the community website. Right: the details page of a project.

1.3. Aim of the software

The aim of our software is to empower children to gather their first programming experiences in a playful way and to share those experiences

with others. The programming environment makes it easy for children from the age of 8 to create interactive content such as games or multimedia animations directly on their smartphones without requiring previous knowledge about software development. The website supports an online community composed of children and users of all ages who can share, rate, discuss, and remix their own projects and those of others in an active way. Furthermore the website should foster the cooperation between children and supports them to get inspired and learn from others.

1.4. Online communities

A very basic definition of an online community is a set of users who are communicating by using some sort of computer and who have common interests [Laz03]. But there has not really been an accepted definition of online communities for more than a century and there is still none today [Pre01]. The term means different things to different people [Pre00].

The supporting software might have slightly changed over the years from old fashioned chat rooms, bulletin boards, listservers or newsgroups to blogs and modern high-end Web 2.0 services like social network sites and content creation communities with extremely cross-linked contacts and content [Pre01] [Ted05] [Rhe93]. There is a trend to social network sites which are a combination of technologies like chats, message boards, wikis, and email and SMS reminders. According to Preece and Maloney-Krichmar especially teenagers are able to switch between different technologies without difficulties [Pre03]. The main intention of online communities remained the same: they are groups where people who may or may not meet one another face-to-face come together to exchange words and ideas [Rhe94]. They are places to share opinions, minds, and creations, or to just keep others up to date about one’s daily life.

Though research has shown that contribution to online communities is mainly done by intrinsic motives, these contributions generate new information

which benefits all members as it becomes a public good of the community [Ted05].

1.5. Target users

Social networks, content creation sites, and online communities have at least one thing in common. They all need users to be operational. Not just that – they need users who are willing to actively participate and generate content. In the majority of cases they do this for free and the only reward they get is what the community offers itself.

The developers of Scratch had Computer Clubhouses in mind when Scratch was in its early stages. Later they tested Scratch with and gathered feedback from youths, mentors, and coordinators at Computer Clubhouses in Boston, Los Angeles, and Dublin [Mal04]. Computer Clubhouses, co-founded by the MIT, are institutions freely accessible for youths where they can become designers and creators of computer-based products. Advice and support is given by adult staff and volunteer mentors [Res96] [Mal04].

In general Catroid as well as the Catroid community are intended to target users from all ages. Nevertheless the main focus lies on teenagers and children from the age of 8. One interesting field of application would be schools. Catroid can be used in computer science classes to let children gather their first mobile software programming experiences. However, kids are encouraged to use Catroid and its community at home and in their spare time too. One of our goals is that they become creators of digital content instead of just being consumers.

A so called *critical mass* is necessary for the diffusion of interactive media such as online communities. Online communities are extremely vulnerable to start-up problems and discontinuance if they do not reach that critical mass [Mar87]. Thus it will be important for the Catroid community website to quickly acquire such an adequate number of users after it has launched. A glance at the Scratch statistics shows that the community reached the critical

mass after a few months. From that moment on the number of newly registered members increased every month [Lif11].

The following section will examine whether there are potential users for our software.

1.6. Mobile phone usage among children

The use of mobile phones is spreading rapidly all over the world. For example in the US the wireless penetration, which means the number of active mobile phones divided by the total population, reached 96% in 2010 [CTI10]. In the world's most populous country China this number is quickly growing with almost 900 million mobile phones in use which means a wireless penetration of 61% in May 2011 [Tho11]. There are countries like Russia [ACM11], Germany [Tec09], Saudi Arabia [Ara10] or Argentina [Bet10] where this number is over 100%. These trends are not limited to adults only. In industrial countries a growing number of children have their own mobile phone.

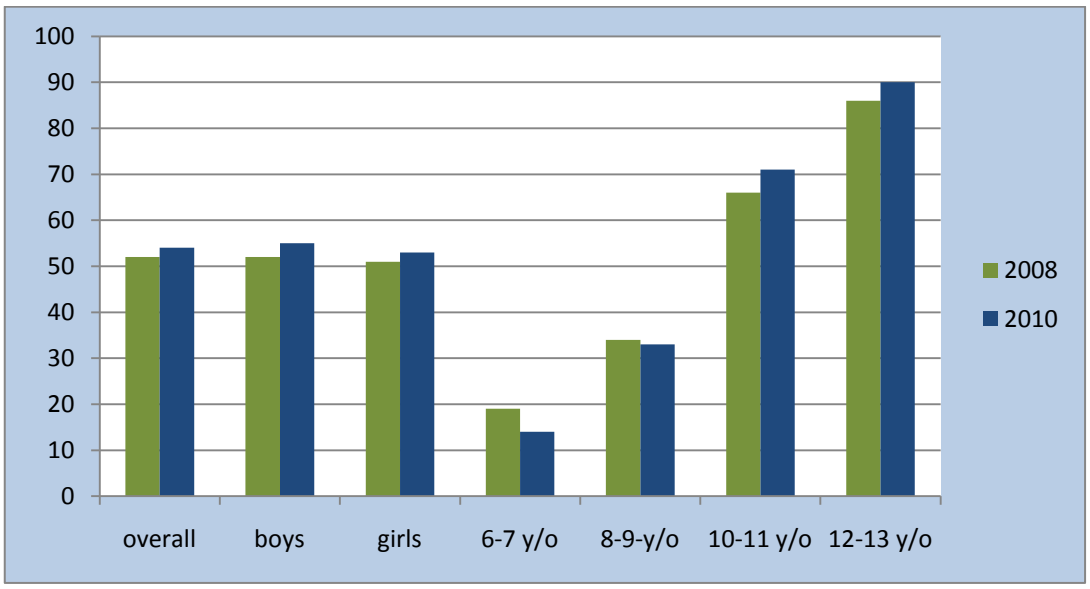

Figure 3. Percentage of German kids who own a mobile phone.

As illustrated in Figure 3, more than 50% of the children aged between 6 and 13years have their own mobile phone in Germany, which is representative for countries in the European Union. In the group of kids aged between 12 and

13 years this value reaches 90% [Med10]. Similar results can be seen in Asian countries like Japan, Korea, China, or India [GSM10]. More and more mobile phones have a fast Internet connection, leading to the fact that children are able to connect to the Internet wherever they want. In Japan almost 70% of the children who own mobile phones use them to access the Internet. The reason for this high percentage is the relatively cheap cost of Internet use in Japan. In China and Korea the percentage is higher than 30% [GSM10]. It is assumed that in Europe the number of kids who access the Internet through their mobile phone will increase soon due to decreasing costs and the availability of flat rates. Today about 15% of the kids in Germany have an Internet connection available on their mobile phones [Med10]. Beside the use of mobile Internet other popular activities are making calls, writing messages, taking photos, and recording and exchanging videos. The trends reflected in these statistics show that the basis for our Catroid software and the associated community website for children already exist. The number of potential users grows quickly and will rise even more in the near future. It is clear that these developments provide perfect conditions for our project.

2. Related work

2.1. Programming environments with online communities oriented towards children

Beside Scratch other community sites supporting and encouraging the sharing of interactive animations and/or games created by kids include those associated with Nintendo's Wario Ware D.I.Y.[6], Microsoft's Kodu[7], Flipnote Hatena[8], and Game Maker[9].

[6] http://www.wariowarediy.com

[7] http://www.ditii.com/2011/03/03/kodu-with-community-game-sharing-released

[8] http://flipnote.hatena.com

[9] http://www.yoyogames.com

2.1.1. Wario Ware D.I.Y.

With WarioWare D.I.Y. it is possible for Nintendo DS players to design their own microgames. They can create their own graphics and music for the games. These games can further be played on Nintendo Wii as well as shared online [Nin10] [1UP09].

2.1.2. Kodu

Kodu is a programming language developed by Microsoft. It is designed especially for young children from the age of 10 enabling them to create complete 3D games including terrain and level design running on the Xbox 360 and on Windows PCs. The language is simple and entirely icon-based, and programming is done solely with a gamepad. Programs are composed of pages, which are broken down into rules, which are further divided into conditions and actions [Mac09] [Mac11] [Mic10]. In March 2011 Kodu received a new release which enables kids to share their games in an online community [DIT11].

2.1.3. Flipnote Hatena

Flipnote Hatena is an online service that allows uploading of flipbook-like animations created with Flipnote Studio. Flipnote Studio is a free DSiWare software title for the Nintendo DSi provided by Nintendo and is preinstalled on many newer Nintendo DSi devices. It is possible to add comments and to star one’s favored Flipnotes. The service was launched in Japan in November 2008 and worldwide in August 2009 [Yam10] [Hat11].

2.1.4. Game Maker

Game Maker is a rapid-application-development tool for young people from the age of 8 to be used at home and in schools to create two-dimensional and isometric games. Users can create games just with drag-and-drop techniques without writing a single line of code. The produced stand-alone games can be distributed freely [Ove04]. By making use of the community

website users can upload, share, and play the games they created with Game Maker. Further the best content created by the community is published and distributed on mobile devices [YoY11].

2.2. Other related systems

Among the jungle of nowadays' social networks there also exist some sites which are especially intended for or extremely popular amongst young people. To mention a few there is SchülerVZ[10] in Germany, bebo[11] in Great Britain, Hyves[12] in the Netherlands, Tuenti[13] in Spain, and Habbo[14] in Finland. In Japan there exist three popular networks, Gree[15], Mixi[16], and MobageTown[17] which are all rather popular among teenagers [CTI10]. Another popular online community designed for children from 6 to 14 years is Club Penguin[18] which is owned by The Walt Disney Company and is available in many countries and languages.

YouTube can also be seen as a platform to share user contributed multimedia content though it is not primarily oriented towards children and the contributed content cannot be made interactive. In June 2011 YouTube started to encourage the remixing of videos published under a creative commons license[19].

2.3. Theoretical work

A lot of theoretical work related to topics discussed in this thesis has been done by the creators of Scratch. This research mainly targets Scratch or the

[10] http://www.schuelervz.net

[11] http://www.bebo.com

[12] http://www.hyves.nl

[13] http://www.tuenti.com

[14] http://www.habbo.com

[15] http://gree.jp

[16] http://mixi.jp

[17] http://www.mbga.jp/.pc

[18] http://www.clubpenguin.com

[19] http://youtube-global.blogspot.com/2011/06/youtube-and-creative-commons-raising.html

associated Scratch community. Early papers discuss the advantages of having an easy to understand programming language for children like Scratch, the impact of the ability to develop skills like problem solving and technological fluency on the wider spheres of children's lives, and some design principles for construction kits for kids [Mal04] [Res05]. Zuckerman, Blau and Monroy-Hernández analyzed the Israeli Scratch community and investigated types of participation and gratification mechanisms identifying "project creators" and "social participants" as two distinct participation types. Further they pointed out the long tail distribution in the Scratch community and the importance of specific design features to increase participation [Zuc09]. Several papers by Monroy-Hernández – the leading developer of the Scratch community website – target cooperation and the mechanism of remixing of interactive content which was always a core idea behind Scratch. They address problems with remixing like the lack of acceptance of automatic attributions as well as the importance of giving manual credit to content contributors [Mon09] [Hil10] [Mon10] [Mon11].

Additional related research apart from Scratch includes work about online communities and social networks in general and how to measure success of online communities [Cot00] [Ted05] [Far07].

3. Development

3.1. Our goals

Inspired by the strong community of Scratch and also by the success of Flipnote Hatena by Nintendo, we consider the availability of an online community to share projects and to support the collaboration between users as essential.

In general our main goal from the very beginning was to create a community website that is intimately connected with Catroid, is highly attractive for children, and has great potential to be as successful as the Scratch

community website. We identified the following basic characteristics such a community website should have before we started its implementation.

(1) It should be as easy and intuitive to use as possible.

(2) It should be optimized for mobile devices but should also be pleasant to use on a desktop browser.

(3) The barriers for users to join and use the site should be as low as possible.

(4) The boundary between the Catroid app and the community website should be fluid.

(5) It should be suitable for children from the age of 8 years.

While these requirements provide a good foundation to start, it is clear that there are further success factors for a website for children that should be considered. Those additional success factors will be discussed in Section 6.

Considering the requirements above the development of the Catroid community website started in August 2010.

3.2. Implementation and development methods

3.2.1. Development methods

In August 2010 we started to make the first paper mockups for the website. Our guidelines were clear: not to add too many features at once but instead making small steps where each step leads to a fully functional version of the website. Each new version should have a small amount of new features and/or improvements of the user experience. The advantage of this method is that the development cycle of the releases is kept rather short and we are able to release almost anytime. Following these guidelines the very first “version” of the website was just a simple design draft. Screenshots of this first version were printed on our poster for the Scratch@MIT conference[20] in

[20] http://events.scratch.mit.edu/conference/index.php/Scratch/2010

August 2010 where our project got a lot of positive and constructive response. An illustration can be seen in Figure 4.

Figure 4. Early design draft of the website.

Another important development method we are using is Test Driven Development (TDD). As we are a free software and open source project, following this principle ensures that we can produce code which stays maintainable over time. Extending and refactoring the code can be done without having to meet the original developers and without having to read special design documents. Our only documentation for developers is the source code of the actual program and the source code of the automatic tests, both refactored many times and optimized for clarity, simplicity, and intuitive understandability. Section 3.2.4 gives further information about tools we use for testing.

To organize tasks and to keep an overview of the project's progress we use an agile software development method called Kanban. It should help to find bottlenecks and other problems and should prevent unexpected delays

[Kol11]. Kanban was originally applied in process administration and was adopted for information technology in 2007 by David J. Anderson [And10] [Kol11]. Figure 5 shows the Kanban board with tickets of the Catroid community website project. It can be seen that there are four lanes on the board, each defining a particular phase in a task's progress. The *Backlog* lane contains all tasks which have been created but not started yet. When a developer pair starts with the implementation of a particular task, the ticket is moved to *Development* and on completion to the *Done* lane. The last lane contains the completed and accepted tasks. To be accepted means that a task has to be reviewed by another member of the team who was not involved in the development of that particular task and that it complies with special requirements such as good source code style as well as existence and positive results of test cases.

Figure 5. Kanban board of the Catroid community website project (June 2011). White tickets are regular tasks, yellow tickets indicate bugs.

The introduction and the use of Kanban during the development of Catroid was target of research leading to the result that Kanban *"[...] can have a positive effect on the value of reliability and maintainability"* [Kol11].

3.2.2. Personas

For the design process our team makes use of so called personas. The personas method is a user centered design tool that was adapted for the online business by Alan Cooper in 1999. The idea behind the concept is to have a small group of fictional but very concretely defined persons – the so called personas – rather than considering an amalgamation of all target users. As Alan Cooper mentions in his book “The Inmates are Running the Asylum”: *“[...] to create a product that satisfies a broad audience of users [...] you will have far greater success by designing for one single person”* [Coo99].

For the university course “Software Engineering and Knowledge Management” held at Graz University of Technology by Wolfgang Slany in 2010, the Center for Usability Research & Engineering (CURE)[21] created three personas for our purposes. The key facts about these personas are summarized in Table 1, Table 2 and Table 3. Figure 6 shows the sheet with a detailed description of the persona which is always present in our workroom. The use of these personas in our project is particularly important because our target user group comprising, e.g., 11 year old girls, is clearly different from the developers of Catroid.

[21] http://www.cure.at

Name	Tobias Madoff
Age	15
Education	high school for mechanical engineering
Social status	has a girlfriend for a short time
Interests	- his mobile phone - computers - music - his moped - online games
Skills	- versed in technical stuff - likes to understand technical things - composes his own music
Web	- uses Facebook and YouTube on his mobile phone - also member of MySpace
Negative experiences	- one of his music clips got pirated
Wishes	- wants to make animated clips with photos and videos and with his own music - wants be able to upload and share those clips on the web

Table 1. Persona *Tobias* [CUR10].

Name	Angelika Bacher
Age	44
Education	abandoned her studies at law school
Occupation	works at a lawyer's office
Social status	mother of 3 children (1 girl - 14 years old, 2 boys - 10 and 8 years old)
Web	- uses the web for emails when she is at home - uses the web for research during work - her older kids use the web to communicate with their friends
Negative experiences	- because of her work she knows that there is a lot of inappropriate content for her children on the Internet - her daughter got some inappropriate offers in an online community once
Wishes	- wants a program that helps her kids to learn but it should also have some sort of social component so that they can be creative and work together -her husband, who is at home very rarely during daytime, wants to see some creations of his children online so that he can take part in their lives at least over the web

Table 2. Persona *Angelika* [CUR10].

Name	Silvia Eder
Age	10
Education	secondary school
Social status	has two brothers (twins, 5 years old)
Interests	- handball - doing handicrafts - painting - Hannah Montana
Skills	- painting - she's quite creative - doing handicrafts
Web	- communicates with friends over the Internet - emails with an old friend in another town and shares photos with her - member of some social networks - her parents do not allow her to upload a photo of herself
Negative experiences	- gets email spam - lost her homework once due to a computer crash
Wishes	- wants a program to do graphical stuff easily - wants to share her work with friends - wants to create short videos or small games and show them to her friends - wants her drawings (mostly animals) to "get alive"

Table 3. Persona *Silvia* [CUR10].

Figure 6. Personas poster [CUR10].

Figure 7. Chocolate with photos of personas.

3.2.3. Implementation

When selecting tools and technologies for the development of the website we make sure to use only freely available and open source software and concepts. This is important as our software is free and open source too and will not become a commercial project in the future. Currently we are using the following tools and technologies for implementing the website.

(1) Apache HTTP Server[22] to host and run the website.

(2) PHP[23] as server-side language.

(3) Additional server-side libraries like the GD Graphics Library[24] for image processing, cURL[25] for making HTTP requests, XML_Serializer[26] for generating XML files, and Snoopy[27] for interaction with APIs.

[22] http://httpd.apache.org

[23] http://www.php.net

[24] http://www.php.net/manual/en/intro.image.php

[25] http://curl.haxx.se

[26] http://pear.php.net/package/XML_Serializer

[27] http://sourceforge.net/projects/snoopy

(4) PostgreSQL[28] as database management system: the decision to use PostgreSQL instead of MySQL was made because of the unclear future of MySQL as free open source software since it is controlled by Oracle.

(5) On client-side we use HTML5[29], CSS, and JavaScript. As additional JavaScript libraries we use jQuery[30] to simplify AJAX requests, event handling, and DOM manipulation, and Classy[31] to keep the JavaScript code more object oriented, better readable, and more easily maintainable.

(6) The free and open source software packages phpBB[32] for our bulletin board and MediaWiki[33] for the wiki of the community website.

(7) Mercurial[34] as version control management tool for our source code hosted at Google Code[35].

(8) PHPUnit[36] for unit testing the PHP source code.

(9) Selenium GRID[37] together with Selenium RC[38] for functional testing of the website.

(10) Pootle[39], a crowd-sourcing software to allow volunteers to correct and/or translate static language texts of the website and the Catroid software into different languages.

---

[28] http://www.postgresql.org

[29] http://dev.w3.org/html5/spec/Overview.html

[30] http://jquery.com

[31] http://classy.pocoo.org

[32] http://www.phpbb.com

[33] http://www.mediawiki.org/wiki/MediaWiki

[34] http://mercurial.selenic.com

[35] http://code.google.com/p/catroid

[36] https://github.com/sebastianbergmann/phpunit

[37] http://selenium-grid.seleniumhq.org

[38] http://seleniumhq.org/projects/remote-control

(11) Apache ANT[40] to run automated tasks like tests, database updates, deploying to server, generating language XML files, etc.

When starting to implement the community website I decided to reuse a small Model-View-Controller (MVC) framework for PHP[41] which I previously had implemented for another project and which is based on an idea from Joe Stump [Stu05]. This made it possible to start immediately without having to learn a new complex PHP-Framework. Another reason for that decision was that we needed some framework which is flexible and easily extendable because of our principle of making small steps and because we did not know exactly what step would come next.

To compare our framework to other popular PHP frameworks we did some benchmark tests on our test server[42] using the Apache HTTP server benchmarking tool. Similar to the PHPmark project[43] we compared the minimal overhead of our framework and some other popular PHP frameworks. The frameworks in question were Symfony[44], CodeIgniter[45], Yii[46], Zend[47], and CakePHP[48] which is used by the Scratch community website. We measured how many requests can be handled by the web server when calling the framework. Database connections had been switched off and no caching mechanisms were enabled. The framework just had to print out “Hello World”. The test made 10 concurrency requests over a time period of 30 seconds. As a result we got the number of requests per second that could be handled by the web server. Figure 8 shows the results of the

---

[39] http://translate.sourceforge.net/wiki/pootle/index

[40] http://ant.apache.org

[41] http://code.google.com/p/mvc-framework

[42] http://catroidwebtest.ist.tugraz.at

[43] http://code.google.com/p/phpmark

[44] http://www.symfony-project.org

[45] http://codeigniter.com

[46] http://www.yiiframework.com

[47] http://www.zend.com

[48] http://cakephp.org

benchmark test. It can be seen that our framework has a much higher performance than the rest of the frameworks. Even when database connections were switched on for our framework it performed better than Yii, Zend, and CakePHP without database access.

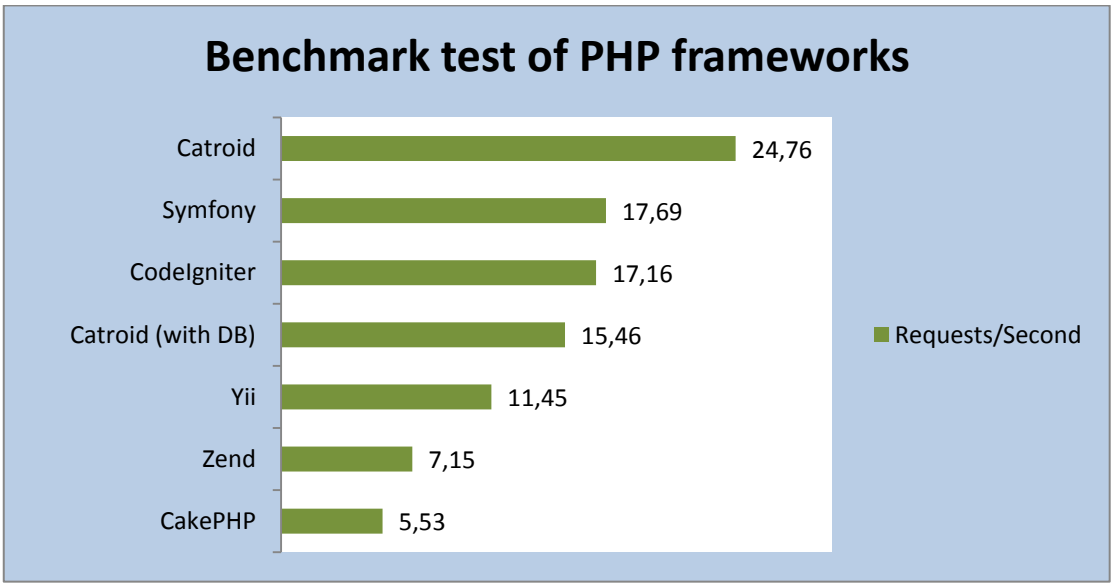

Figure 8. Benchmark test of different PHP frameworks on our test server. Even with database access the Catroid framework outperformed some other frameworks without database access.

3.2.4. Testing

As already mentioned we use Test Driven Development (TDD). As our main website logic is written in PHP we use PHPUnit for unit testing. The tests can be automated with ANT. Additional to unit tests we test the overall behavior and functionality of the website as well as the functionality of our java script functions with Selenium. Selenium tests simulate user behavior and navigate through the website as a normal user would do. During this procedure several conditions are asserted. These tests are rather slow as they restart the browser for every test case and should be executed for all supported browsers. Thus we use Selenium Grid as this allows us to run tests in parallel on different machines.

3.3. Early design and features

3.3.1. First release

The first working release of the website simply provided the possibility of up- and downloading projects. These two functions were considered as the core functionality of the website. In this context two aspects were most important to us: the first was the ability to upload the project directly from the Catroid app without having to visit the website with a browser. This functionality makes sure that children can concentrate on the creation process and do not waste time with searching for ways how to share their creations. The upload process should be as simple as possible. The second important aspect was the ability to immediately open a project of interest with the Catroid app after using the download functionality of the website. There should be no additional step necessary between downloading and trying out the project. Due to the lack of a player on the website to view and try out the projects, this aspect becomes even more important. Such a player is hard to realize because Catroid already uses specific device sensors and will use them even more intensively in the future. The access to all these sensors through a browser is strongly limited at the moment.

3.3.2. Refactoring

As our developer team grew, more and more people began to work on different parts of the website. Because of this it became necessary to somehow automate the test process of the code. Unit-tests were added for the framework which had no tests at all before. In addition to the unit-tests we decided to use Selenium to test the functionality of the website. Some other changes “under the hood” were made to the framework like error handling and prepared statements for database queries. All these steps were necessary to reduce the risk of bugs and keep the whole project flexible and easy to extend. These changes led to our so called refactoring release which went online around the end of December 2010 and already contained some demo projects for downloading and sharing.

3.4. Current state

In January 2011 we started to implement the next release and the implementation is still going on by July 2011.

3.4.1. Inappropriate content

Our goal is to bring the website to a state where it is ready for the public and, most important, suitable for children. This brings up one important question. What does “suitable for children” mean? One of our three fictional personas is *Angelika Bacher*. She works in a lawyer’s office and sometimes has to do with Internet crime. For this reason she has some concerns when her children use the Internet. And she is right: 16% of the children in Germany reported that they stumbled upon sites which were – in their own opinion – inappropriate for children. More than half of these sites were inappropriate because of sexual content and about 20% because of violent content [Med10]. The third important concern of parents about their children’s use of mobile phone is that they are viewing inappropriate sites – it comes right after overuse and cost of the bills [GSM10]. While keeping these aspects in mind we find the presence of two features essential:

(1) Users should be able to report any project on the website when they think it has any inappropriate parts, regardless whether these parts are in the title, description, thumbnail, or even inside the project and only become visible during the execution of the project.

(2) Inappropriate content which is detectable automatically should be rejected.

To realize the latter feature we implemented a filter for common inappropriate words. This includes an extendable list of cusswords which every nickname of new users, project title, project description, comment, wiki entry, and forum posting is compared to before it is accepted. This method to detect inappropriate content is far from being perfect, e.g., with respect to false positives or completeness. For example it cannot take into account inappropriate graphics or audio content. Therefore we provide a second

possibility to identify unsuitable content by making use of the power of social communities. Every project's details page contains a button to report inappropriate content within that project. By using this button and then providing a short explanation, each community member can actively take care of the community. If a project is reported as inappropriate it immediately gets hidden and an email is sent to the administrators of the site who can review the project and decide to remove or restore it, and possibly extend the list of inappropriate words.

3.4.2. Current project arrangement and details

As mentioned earlier our first release just provided the functionality to upload and download projects. The website mainly consisted of one page which was just a huge list of all uploaded projects. That was not a problem since only a couple of demo projects were uploaded to the site. To be prepared for the future when a lot of projects will be uploaded by many different users, we changed the behavior of the project list. Several of the most recent projects are initially displayed. At any time only a defined amount of projects is displayed at once to keep the site clearly arranged and to avoid overloading it. To navigate through older projects or back to the recent projects again, buttons are available at the top and the bottom of the list. To display the newest, which means the just uploaded projects, right on the start page can be an essential factor to increase participation [Zuc09]. A search function is also planned for the current release which is pretty important to find a particular project as the community grows. Other methods of arranging the projects will be considered in the future. This will be discussed in Section 6.5. A details page is available for each project where project description and other details like upload date, author, view- and download-counts, and also the inappropriate button mentioned above are presented in a clearly arranged manner. From that details page it is also possible to download the project.

Depending on the size and number of pictures and sound clips used in a project the download can have a size of several megabytes. Without a data

flat rate such downloads are not recommendable. To protect children from large bills for their mobile phones – as already mentioned, this is a major concern of parents – we display the size of the project directly inside the download button. Additionally a QR-Code representing the download link of the project is displayed when the project details are viewed with a desktop browser. This provides a quick and easy way to download the project directly to a mobile phone just by taking a photo of the code.

3.4.3. User registration

Another big part in this release and an important step toward becoming a real community is the introduction of user registration. This provides the possibility to connect projects to an author so that users can identify themselves with their work. To become even more socially connected to our users and to give them a place to communicate with each other, ask questions, and provide feedback we added a discussion board. Also a wiki is available to have a powerful tool to present help pages, frequently asked question, and other supporting documents in a structured and multilingual way. The user management of the discussion board as well as the wiki is connected directly to the user management of our Catroid website. This has the advantage that users have to register and login to our site only once, which further supports the goal to make the community website as easy and intuitive as possible. This clears the way for lots of interesting features to implement in the future. These features will be discussed in Section 6.

3.4.4. Internationalization

It is essential for a successful website to be available in several languages. While English is always a good default it is important to be able to support other languages as well. This becomes even more important when designing an international website for children. On the one hand this clears the way to get a foothold in a wide range of countries and on the other hand this fits to our goal to foster the cooperation among kids as it scales up the number of children that can easily communicate with each other. As we are an international developer team with speakers of English, Mandarin, Cantonese,

Hindi, Arabian, Japanese, Russian, German, Turkish, French, Urdu, Rumanian, and Malaysian this is a great chance to easily translate the community website into many languages.

Due to our development method of taking small steps and incrementally adding features we did not consider internationalization at all when we started the implementation of the community website. This fact makes the internationalization a major refactoring task.

We were provided with a great opportunity when we got selected as a mentoring organization for Google Summer of Code (GSoC) 2011[49]. In this program Google offers money for students who write code for open source projects. Our project got four slots from Google which meant we could hire four students. After reviewing all 180 project proposals we received from students from all over the world we handpicked the four most applicable proposals. As one of those we chose the internationalization of the community website which means that a GSoC student is currently (summer 2011) developing the system together with some of the senior developers of our team.

The internationalization can be divided into the following requirements:

(1) Internationalization of static website text

(2) Internationalization of inappropriate words filter

(3) Translation of user generated text (e.g., project titles, comments, project descriptions)

(4) A template-based system for comments

(5) Integration of crowd-sourcing software Pootle[50]

To translate static website text we store all text strings in XML-files which can be easily imported into Pootle. Pootle is a crowd-sourcing software which allows, e.g., parents or teachers to correct and add translations and new

[49] http://www.google-melange.com/gsoc/homepage/google/gsoc2011

[50] http://translate.sourceforge.net/wiki/pootle/

languages. As soon as at least 30% of the text is available in a new language, it can be written to a new XML-file and added to the website which then supports another language (missing translations are replaced by the English version of the words and phrases). The same methods are used to provide an international version of the cussword filter to detect inappropriate words in project titles, descriptions, comments, forum postings, and wiki pages, in as many different languages as possible. Compared to translating static website text it is quite a challenge to support different languages throughout all user generated text. We use Google Translate API[51] to optionally offer an automatic translation of project titles, project descriptions, and comments to a project. As the Google Translate API has become deprecated as of May 2011 and will be shut down in December 2011 we consider using the Translate Element[52] of Google Web Elements instead. Additionally to that we support children in communicating with their foreign peers by introducing a simple template-based system for comments similar to the system used on Club Penguin[53]. Through this method kids can build comments out of a set of templates of common phrases. These templates are available in each supported language, again editable and extendable by the crowd-sourcing part of our project, and hence the most popular type of comments can potentially be displayed in the users' favored language.

The development of the internationalization system is currently progressing and an international version of the community website supporting several popular languages should be available by the end of August 2011.

4. Measuring success of online communities

To measure the success of a website, social network site or online community it is essential to gather the appropriate statistics. Some of these statistics can be acquired by using powerful and free analyzing tools like

[51] http://code.google.com/apis/language/translate/overview.html

[52] http://www.google.com/webelements/#!/translate

[53] http://www.clubpenguin.com

Google Analytics[54] while others can easily be collected from the usage data of the community. Some statistics about the Scratch community are publicly available on the Scratch website.

The aim of this section is to identify measures for the Catroid community in order to make predictions about its success.

4.1. Identifying measures

By the time our community website goes publicly online we would like to have a system in place to assess whether the website is successful or not. The first challenge is to identify suitable measures for our purpose.

Figure 9 shows three dimensions of community measurement – three kinds of metrics that every community should track – as they are described by Cothrel [Cot00].

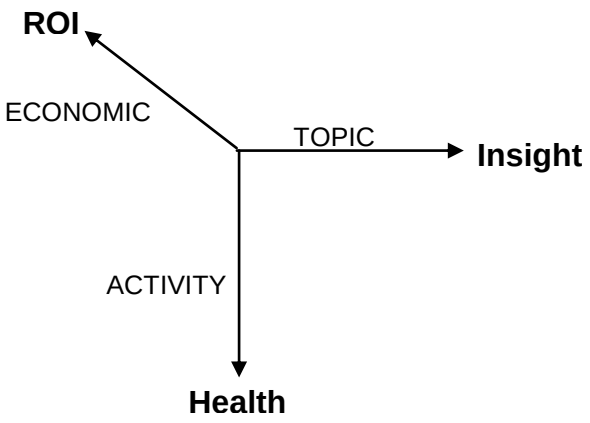

Figure 9: Three dimensions of community measurement [Cot00].

The economic axis illustrates the financial value or Return-on-Investment (ROI) for the community [Cot00]. Upon the fact that Catroid as well as the associated community are not intended to make profit it is clear that there will be no ROI axis needed for our community. But it could make sense to replace this ROI axis with a new one regarding the user satisfaction as could be seen in Figure 10.

[54] http://www.google.com/intl/en/analytics

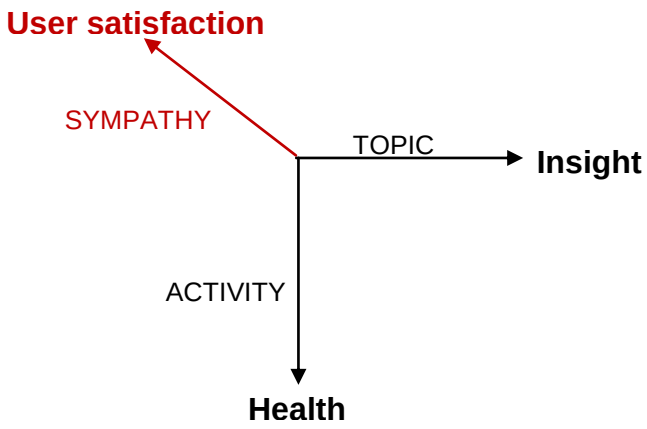

Figure 10: Added axis for user satisfaction.

According to Cothrel, activity measures indicate the general health of the community while topic measures assess the insights that are offered by a community [Cot00]. User satisfaction measures could indicate the overall sympathy with our community.

Cothrel identified a list of most common measures for online communities which all apply to the category of activity measures [Cot00]:

- unique visitors
- page views
- session time
- community click-through
- registered members
- postings per day/week/month
- read-to-post-ratio
- page additions
- page revisions
- peak number of concurrent users
- total number of users
- audience penetration

- repeat visits
- frequent visitors

Not all of the above measures will fit for our purpose. For example from the very beginning of the World Wide Web *thread depth* has been a common measure for interactivity in communities [Raf97]. But this would not make much sense for our purpose. A look at the Scratch community statistics shows that, amongst others, they are using the following measures:

- registered members
- unique project creators
- uploaded projects
- comments
- scripts
- sprites
- remixes

Additionally the number of new users, projects, remixes, and comments can be seen on a monthly base back to the year 2007 when the Scratch community was launched [Lif11].

For the purpose of the Catroid community we adapted these lists to get a suitable set of measures. Once our community website will be launched publicly this data can be used to measure the success of the community or to identify potential problems. The following lists show the recommended activity measures for the Catroid community:

- registered users
- number of participants
- uploaded projects
- project downloads
- remixes

Additionally these measures should also be available on a monthly or even a weekly base as decreasing measures can be an indicator for potential problems or disaffection of users. The number of participants can further be refined into different types of members like content creators, lurkers, question askers, etc. [Pre01]. Two distinct participation types – content creators and social participants – were also identified in the Israeli Scratch community [Zuc09]. Additional to these activity measures the following topic measures should be used to gather information on how complex the uploaded projects are and how intensive different functionalities of Catroid are used by children:

- number of sprites
- number of scripts
- used bricks

Measuring the number of times a specific brick is used in the projects will give us the possibility to identify popular and unpopular bricks. Note that unpopular bricks could indicate that kids do not well understand their usage or that they simply seldom need these bricks.

To measure user satisfaction we can use the following sympathy measures:

- recurring users
- suggestions for improvements

To extend those measures additional statistics regarding the age, language, and location of users can be collected to determine the success of the community in a particular set of users. Optional registration data, logged data, and Google Analytics will be used for the Catroid community website to acquire those additional statistics.

4.2. What does *successful* mean?

Participation and member contribution are essential for an online community to be successful. It is not sufficient to just build an online community or try to attract potential users by using technology innovations [Ted05]. What

success means for an online community strongly depends on the type of community. The goal of an online community created by a business, for example a community for buyers and sellers in an online marketplace to exchange information related to purchases or sales, would be a positive impact on profitability or revenues. Thus success would mean amongst others to get a high return on investment [Cot00]. The same applies to the majority of contemporary business networking platforms like XING[55] or LinkedIn[56] as well as to some social networks like StayFriends[57] which all generate revenue through membership fees for premium features [End08]. This model is not limited to adult communities only. Disney's Club Penguin[58], a community for children, relies on a business model that requires a paid subscription. Most of well known free social network sites, like Facebook[59] or recently launched Google+[60] as well as a lot of free content creation services such as YouTube[61] or Twitter[62] generate revenue through advertisements. These services are highly interested in the so called click-through rate which measures the success of online advertising.

As Catroid and the Catroid community website are free services the project is not profit oriented and no money will be collected through advertising, user fees, or by selling user data. A good way to determine the success of the Catroid community is to compare it with similar initiatives like the Scratch community. A possible evaluation criterion is the rate of growth of the popularity of the site. Ideally, this growth should be exponential (up to a natural limit) as that would ensure that almost every potential user would know about the system in a short period of time. A community grows

[55] http://www.xing.com

[56] http://www.linkedin.com

[57] http://www.stayfriends.at

[58] http://www.clubpenguin.com

[59] http://www.facebook.com

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

[61] http://www.youtube.com

[62] http://twitter.com

exponentially if, e.g., the number of users through new registrations increases by at least a constant multiplicative factor over equal time intervals. Figure 11 shows the growth of the Scratch community since its launch in 2007. The initial growth looks exponential though it may now have slowed down a bit but still increases impressively. We aspire at least to match this growth rate to have a significant impact on the spreading of programming skills and understanding of free open source software among the children of the world.

Figure 11: Growth of the Scratch community.

5. The long tail

Chris Anderson first mentioned the term *The Long Tail* in connection with Web 2.0 in an article in the Wired magazine in 2004. Anderson explains how Amazon.com could push the popularity of rather nameless books through the combination of infinite shelf space and real-time information about actual buying trends and public opinion. A few years earlier these books would never ever have reached that many readers. This is not limited to books only. It rather becomes an entirely new economic model for the media and entertainment industries. This applies to every service where it is possible to

provide a nearly unlimited selection and people can go deep into the catalog to the very end of the list, far away from blockbusters or mainstream. This is what Anderson calls *The Long Tail*. The comparable small amount of bestsellers and mainstream books represent the so called *short head* while the rest is the *long tail* [And04].

5.1. The long tail in today's Internet businesses

The long tail phenomenon can be seen in almost every branch of Internet business, e.g., in Amazon's book shelf, together with the view counts of YouTube videos and also on Wikipedia, where almost every topic has its article. To give another example contact pools on social network sites also follow this mechanism. The short head is represented by the contacts which are accessible via traditional networking. The maximum number of contacts retainable via this method is reached very quickly while many additional contacts representing the long tail are accessible only through the potential of online networking [And04] [End08]. Figure 12 shows some statistics from the Scratch community illustrating the content creation and comment behavior of community members. Short head and long tail can be easily seen in these statistics.

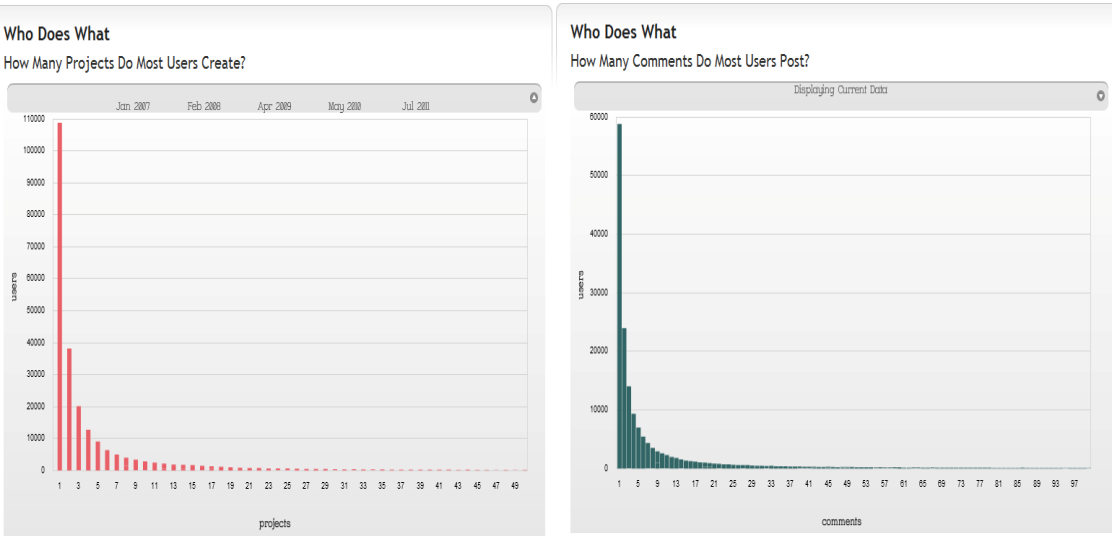

Figure 12: Scratch statistics on "How Many Projects Do Most Users Create" (left) and "How Many Comments Do Most Users Post" (right) [Lif11].

5.2. The long tail in the Catroid community

The Catroid website can be seen as a huge catalog of user contributed projects. As the number of contributors rises the diversity of the Catroid projects will grow as well. When looking at the Scratch community website we see that there are projects targeting almost every topic. We can expect that we will have a similar long tail distribution of projects on our website.

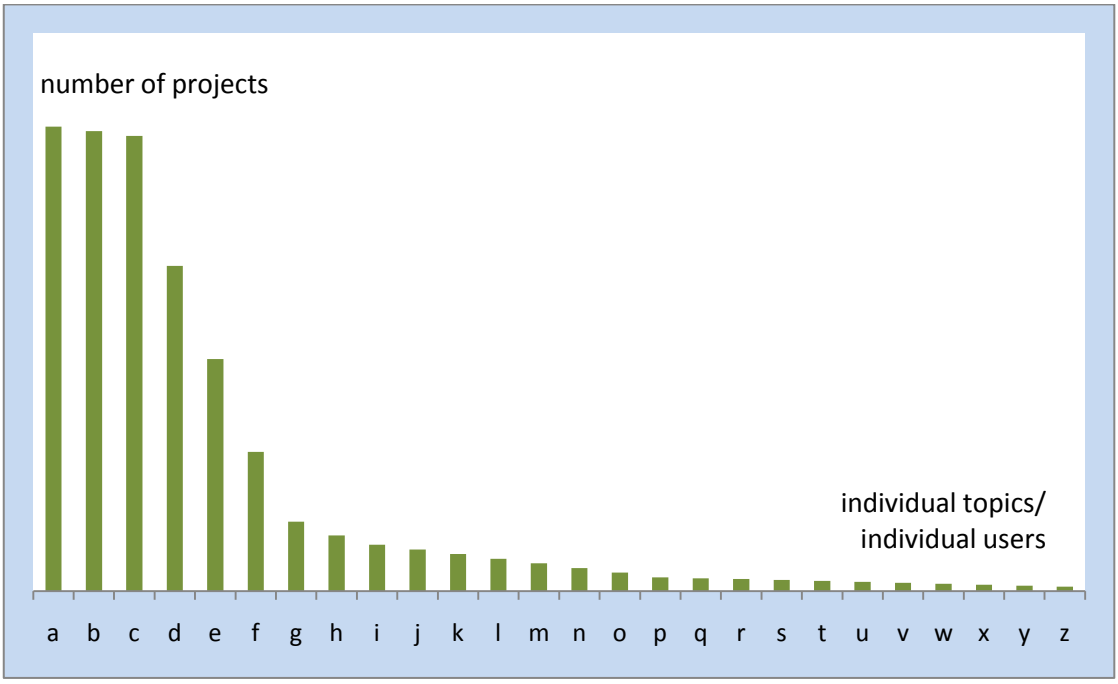

Figure 13: Long tail distribution: lots of projects deal with a few popular topics while many less popular topics often relate to just one project, and a small group of users create a lot of projects while the major part often creates just one project.

Figure 13 shows how the long tail could look like in the Catroid community website. It can be interpreted in different ways. As already mentioned it can be seen from the topics point of view. Many projects will deal with a few popular topics while there will be a lot of less popular topics that often just relate to one single project. Furthermore it shows that comparatively few users create a large number of projects while the major part of users creates just a few projects. However, as the number of contributors raises this long tail represents the major part of the total number of projects. Similar long tail distribution in content contribution can also be seen in other communities dealing with user generated content like Wikipedia or in educational

Wikibooks [Rav08] and are also reported by Zuckerman, Blau and Monroy-Hernández in their analysis of the Israeli Scratch community [Zuc09].

The long tail distribution will lead to the fact that only a small amount of projects will be very popular as the major part will deal with less popular topics which are attracting fewer users. Again, this long tail consisting of individual, less popular projects will represent the major part of the overall projects and may in sum also be much more popular and thus important for the majority of users than the few individual popular projects. This distribution leads to the situation that these popular projects will get a lot of views, downloads, and comments which will make them even more popular. As most of the projects shown on the main page of the Scratch website are chosen from the pool of popular projects, contributors of other projects have no chance to be found through the main page. This is criticized by Scratch users in the Scratch forums[63]. Note that they might still be found by searching for keywords.

Thus one of our challenges would be to focus our users' attention also on projects located in the long tail to increase user satisfaction and as consequence participation. Some methods to make projects located in the long tail easily accessible for children will be discussed in Section 6.5.

6. Future

After the current release (July 2011) as described in Section 3.4 is finished, the community website is in a state which can be called *ready for a public beta*. From that point on, the community will grow and we will have statistical data about how the website is used as well as feedback data coming right from our users. At the point of time this thesis was written this state has not yet been reached. Nevertheless this section gives an overview how the development is planned to evolve in the future.

[63] http://scratch.mit.edu/forums/viewtopic.php?id=43176

6.1. Inspiration sources

Evaluation of the findings of scientific papers and statistical reports regarding the use of online communities by children and topics relative to that lead to some suggestions how the Catroid community website can be successful. As an additional inspiration source the Scratch 2.0 suggestions page[64] can be used. The developers of Scratch give their community the possibility to make suggestions for the upcoming Scratch version 2.0 and to vote for particular suggestions made by others. The suggestions can deal with features for the Scratch software as well as for the community website. For the purpose of this paper we consider the latter. In general these suggestions are public to everyone for reading on the Scratch forums. New suggestions and votes for specific suggestions can only be made by registered users who are active community members. They must have left some constructive comments and have uploaded their own projects in the past. Thus we can be sure that these suggestions are made by Scratch members who really know what they are talking about. This suggestions page can be a good source for us to find out what such a community of young content creators wants and what it does not like.

6.2. “Must have” functionality

There are several features which are kind of “must haves” for online communities and which are planned to be integrated into the Catroid website too.

6.2.1. Comments

It should be possible for children to easily add comments to each project. By doing so kids can commend, criticize, or just discuss the work of others. They can convey what they love about a project or what they would suggest to improve. The communication with peers fosters the ability to learn from each other and to improve one’s skills. A glance at the Scratch statistics shows

[64] http://suggest.scratch.mit.edu

that kids love comments. 300.000 to 400.000 new comments are composed each month in the Scratch community [Lif11].

6.2.2. Votes

One key motivator for children to use Catroid and join the associated community is the ability to entertain their friends and other kids. To foster this motivation and the children's participation to the Catroid community, project ratings given by other members can be supportive. Rather than rating a project based on a defined scale, children should be able to vote for a project which means to recommend it to others. To keep this mechanism simple it should be possible with one single click. The implementation could be some sort of "like" button similar to Facebook or a "love it" button like in the Scratch community. While it is possible and even welcome to write criticizing comments it should not be possible to give a negative vote. Thus there will be no "dislike" or "don't love it" button. Too many negative votes for their project can discourage children from further participation. If someone doesn't like a particular project it is still possible to write a comment and give reasons for one's opinion – ideally together with some suggestions for improvements. In any case it is better to explain criticism in words than just clicking a single button. To clearly trigger the number of users who "love" or "liked" one's project this information could be displayed on each project details page beside the number of views and downloads.

6.2.3. Tags

We also plan to integrate a tagging system into our application. Tags can help to enhance certain functionality like the search function and can lay path for further features like for instance a recommendation system. To acquire tags we can make use of automatic methods, for example automatically identifying keywords in project titles and project descriptions. Additional to that we enable children to add their own tags for their projects right before the upload when they also choose a title and a description.

Further a collaborative tagging system allowing community members to annotate any project with personalized tags would create a much richer network which is often called a *folksonomy*. Such a folksonomy is a network of interrelated users, projects, and tags. However, to solve problems like tag redundancy some sort of data mining technique such as hierarchical tag clustering can be used. The resulting tag clusters can then be used as a basis for a powerful personalized recommendation system [She08] [Mat04].

6.2.4. Friends

Like tags, friends-connections support the creation of a knowledge-base which can then be used to build a complex recommender system as well as to make basic recommendations like “projects uploaded by your friends”. See Section 6.6 for more about recommender systems.

It would be also a benefit for children if they can add other kids as their friends. Children can be more willing to invite their friends if this would expand their personal network in the Catroid community. As described later in Section 6.7, newly invited members can be added as friends automatically.

Another application of friends-relations could be an alert system. Whenever someone uploads a new project one’s friends could be notified automatically (every user should be able to turn this off individually).

6.3. Social networks

Over the last years social networks have become more and more popular not only among adults. Also a rapidly growing number of children discover the opportunities behind these networks. In Germany 43% of the children between the age of 6 and 13 years used an online community at least once a week in 2010. Two years earlier in 2008 this percentage was only 16% [Med10]. This trend is not only seen in Germany. In Japan for example 65% of the children who have a mobile Internet connection use it to read messages on social network sites and on their friends’ profiles. In China this value still reaches 27% [GSM10]. In the United States the situation is similar. 65% of the teenagers between the age of 13 and 17 use social network sites

where Facebook is the most popular with 71% of those teenagers using it. Also 28% of the 8 to 12 year old children who are using social network sites are registered on Facebook [Har10]. According to the traffic analysis website quantcast.com 22% of the Facebook users are aged between 13 and 17 years. For MySpace this percentage is even higher: 27% of their users are in this age range [Qua11].

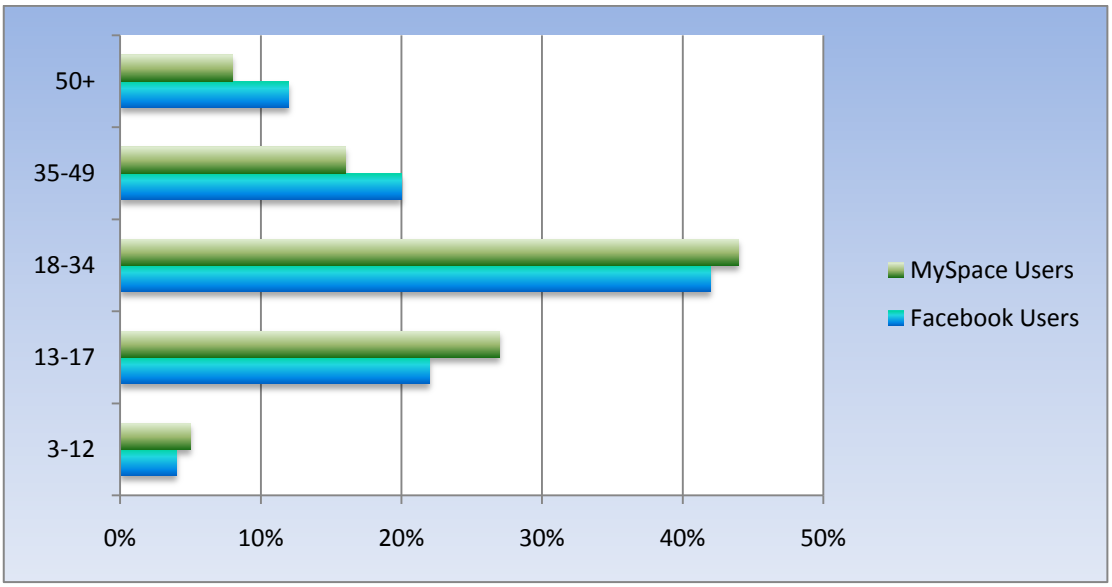

Figure 14. Age of Facebook and MySpace users [Qua11].

The most popular social network site used by children in Germany is SchülerVZ. Two thirds of the children who are members of a social network are using this site. Facebook is used by 22% of the children [Med10]. In Japan Gree[65] and Mixi[66] are the leading social networks which are also used by children [GSM10]. On third place in Japan is MobageTown[67]. It is a mobile-only social network which is more like a virtual gaming world but also offers the ability to chat, communicate, and share. See Figure 15 for an overview of Japan's most popular social network sites.

[65] http://gree.jp

[66] http://mixi.jp

[67] http://www.mbga.jp/.pc

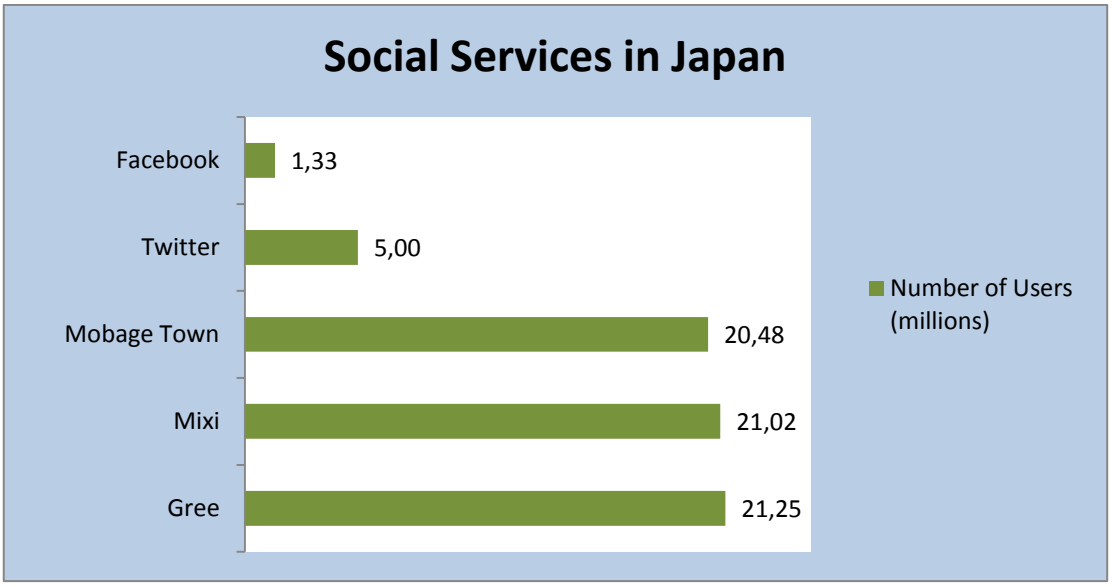

Figure 15. Social services in Japan by number of users [Aki10].

It can be seen through the numbers above that the popularity of a social network site differs a lot from country to country. It seems that almost each country has its own popular social network. This fact makes it very difficult to propose a specific network which should be supported by our community website.

Most of the social network sites mentioned above provide some sort of application programming interface (API) to communicate with them. A possible scenario could be the following: *Tobias* (one of our personas) registers with our Catroid community website. He has a Facebook profile too and connects his Catroid profile with that profile on Facebook. Tobias creates a new project in Catroid – a small multitouch jump 'n' run game with a song composed by him as background music. He uploads the project to the Catroid community website. At the same moment the project appears in the newest projects gallery on the website, a message automatically is generated and posted to Tobias' Facebook wall: "Tobias just uploaded a new project to Catroid! Visit catroid.org to download and run his project." Tobias' Facebook friends notice the message and visit the Catroid website if they are interested. This could lead to a viral marketing effect. The advantage for Tobias is that this will keep up his friends with his activities on Catroid.

Germany	UK	US	Japan
SchülerVZ	Bebo	Facebook	MobageTown
Facebook	Facebook	MySpace	Mixi
			Gree

Table 4. Popular social network sites among children and teenagers.

While this sounds rather good the problem to support different social networks in different countries still remains. Thus the first decision which has to be made in this context is to determine which countries we will concentrate on. If this has been clarified the most popular social network sites of these countries should be supported. An important prerequisite is that the selected social networks provide an API to communicate with them. Most of the sites in Table 4 support OpenSocial, a set of APIs for social network applications. This fact would reduce the effort of implementation by only having to implement for one defined interface. Facebook currently does not support OpenSocial though it provides a proprietary API.

Integrating popular social networks into our community website would create great opportunities for us. Through those social connections our site would get a lot more attention by kids who usually would not be attracted. Also the projects of each user get more attention by being presented and promoted on the children's social network walls. This ensures that the Catroid community website will not become some sort of "closed system".

6.4. Remixing

In connection with our community website, *remixing* describes the possibility to download projects, view the source, edit the source to extend or change the behavior of the project, and upload the changed version again to the community website. This is rather not just a possible feature for the future but it is a *must have.* Being able to remix other projects will give our community members the possibility to build upon and learn from the work of others.

Manovich cites Barb Dybwad writing *"[…] the most interesting aspects of Web 2.0 are new tools that explore the continuum between the personal and the social, and tools that are endowed with a certain flexibility and modularity which enables collaborative remixability – a transformative process in which the information and media we've organized and shared can be recombined and built on to create new forms, concepts, ideas, mashups and services"* [Man05]. Remixing is not a new invention – Hill claims *"remixing is as old as creativity"* [Hil10]. In several areas, for example music or art, remixing is a well known and common concept. Another example to demonstrate the popularity of remixing is the fact that according to filmmaker Kirby Ferguson's blog[68] "*[…] of the ten highest grossing films per year in the last ten years, 74 out of 100 are either sequels or remakes of earlier films or adaptations of comic books, video games, books and so on […]*" [Fer11]. YouTube started to encourage the remixing of videos published under a creative commons license at the beginning of June 2011[69] which shows the growing support of the remixing concept in other major online communities.

The Internet and especially all the aspects of the Web 2.0 with its social component make remixing very easy and therefore available for nearly everyone in real time. The Scratch community website statistics illustrate that children love remixing too. About 28% of the projects are remixes [Lif11]. The principle of remixing was called "deep shareability" in the early stages of Scratch. It allows exporting objects to many types of devices and exchanging them with friends [Mal04]. As Monroy-Hernández mentions, remixing was one of the core ideas behind the Scratch community website [Mon09].

As already mentioned at the beginning of this thesis, Catroid can be understood as some sort of Scratch-like application for mobile devices running Android. Our goal is to encourage children to create their own programs and learn from the projects of others. For that reason the described principle of remixing should become an important part of our community

[68] http://www.everythingisaremix.info

[69] http://youtube-global.blogspot.com/2011/06/youtube-and-creative-commons-raising.htm

website too. Remember one of the wishes of persona *Angelika Bacher* from Table 2: she wants computer programs having some sort of social component to give her kids the possibility to learn from each other. Remixing would be such a component.

When talking about remixing there is an important point: always credit the author of the original project. Otherwise users can get upset because they think someone has stolen their work. In their study, Hill et al. analyzed several projects in the Scratch online community and found that 22% of the original creators who saw remixes of their projects left a direct or indirect complaint of plagiarism [Hil10]. This is an unsustainable situation as Scratch was conceived and designed as a platform for remixing [Mon11]. A first attempt to solve this problem was the introduction of a new system that automatically identifies derivative work and displays the original project and its author under each remix on the Scratch website. But the problem still remained. Monroy-Hernández points out that mentioning the origin project under its remix is not enough. For this reason the people from Scratch introduced a new section to the start page of the Scratch community website – the top remixed projects [Mon09] [Mon10].

The results of a recent study by Monroy-Hernández et al. show that such automatic attribution systems are not effective enough to prevent disaffection in communities like Scratch. Often remixers also gave manual credit to the project originator. This was done in the project description or even inside the project on a special credits screen. Most of the children preferred getting such explicit manual credit from the remixing person rather than automatic attribution. Manual credits were found to be more authentic and important and have greater effect on user behavior as they can only be made by humans. One suggestion the researchers gave is to remind users to give credit by presenting an additional field for credit [Mon11]. In Catroid such a metadata field could be shown before uploading a project which has been identified as remix. In general it should become a common behavior throughout the community to give credit. The authors of above study state

that affordances that help and remind users to do things that a system might do automatically become increasingly important patterns for design interventions in support of community norms for social media communities [Mon11].

Another possibility to prevent children from getting upset when their projects are remixed would be a completely different approach how remixes are assigned and displayed on the community website. On the Scratch community website remixes are always owned by the remixer. We could assign remixes not to the remixer but to the author of the original project. They could be displayed in a special section on the author's profile page called "remixes of my projects". The remixer could have a page with "projects I remixed" respectively. While this approach could help to solve the crediting problem it could lead to other problems. If the remixes of a particular project are remixed again and again this could lead to a project which has nothing to do with the original project anymore but will still be displayed as owned by the original author. This approach deserves further study and can be tested once the Catroid community has an adequate number of users.

6.5. Arrangement of projects

6.5.1. Current state

As illustrated in Figure 16 in our alpha version of the community website (July 2011) there is only one method of project arrangement on the start page: all projects are ordered by their upload time where the newest projects are on the top. It seems to be a suitable method while there are only a few projects available and finding a particular project in that list isn't a real challenge. A search function was added to the current website which allows to search by terms used in the project title and project description.

6.5.2. Best practices

A growing amount of projects will require additional methods of project arrangements to make it easy for kids to find projects of their interest and to focus their attention also on less popular projects located in the long tail. On

the Scratch website you can see newest projects, projects which are most loved, viewed or remixed by the community, and featured projects. To give another example, YouTube shows the most popular video clips and some recommended clips if the user is logged in. Looking at the Scratch 2.0 suggestions[70] by the Scratch community, it seems that the arrangement of projects is an important topic. There are several suggestions dealing with different methods how to order the project lists. The importance of the project placement method is also mentioned by Zuckerman in his study with Israeli learners in the Scratch online community. Zuckerman explains that participation can be increased by certain design features of the community website, e.g., that new projects are displayed on the start page immediately after the upload [Zuc09].

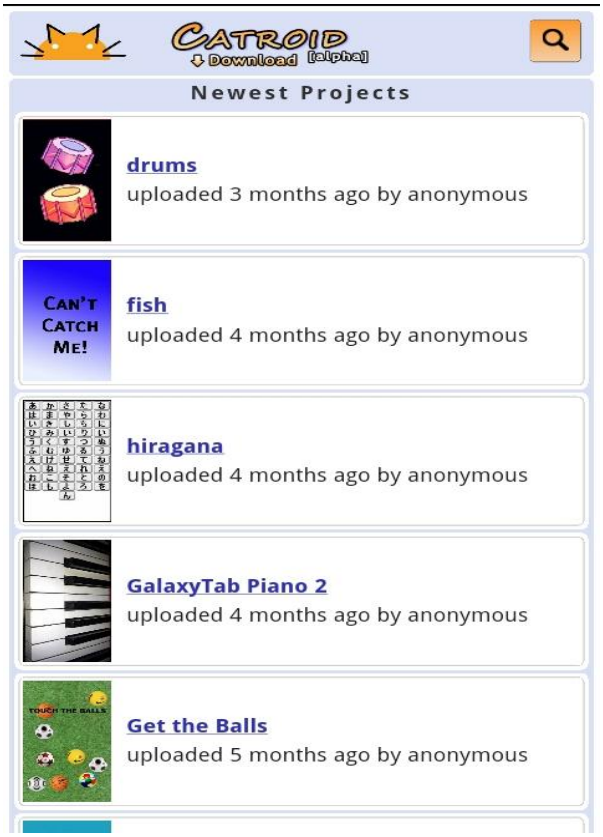

Figure 16. Project arrangement in the current (July 2011) alpha version of the community website.

[70] http://suggest.scratch.mit.edu

The Scratch community website has a lot of good features already – much more than the current version of the Catroid website. Nevertheless Scratch users are still making some interesting proposals. The top voted suggestion concerning the arrangement of projects is to make the “What the community is loving” – , “What the community is viewing” – and “What the community is remixing” – lists more variable. At the moment just the top three projects of each of these categories are shown on the start page. It can happen that they stay the same for a long time as they have a higher chance to be viewed, loved, and remixed again. The idea is that the three projects for these lists should be selected randomly from a greater amount of projects of each category. That would help more authors to get their project presented on the start page which would lead to more views, “I love it”s, and remixes for a broader range of projects. It would also lead to more satisfied authors and indirectly motivate more intensive and active participation.

6.5.3. Geographically related projects

A project not clearly showing up on the “newest projects” page in one’s browser right after one has uploaded a project can discourage children to participate further. A reason why a project disappears too quickly could be simply because there are too many uploads at the same time. This problem can be solved by only showing a set of projects which are geographically related to the website visitor and by ensuring that one’s own latest project is included in one’s view of the webpage, for a certain time and until one accesses the page for the third time. Thus a child in Korea will not see projects from Russia when viewing the “newest projects” page and a girl in Shanghai will see her newly uploaded animation at the top of other new projects from her region in the first few minutes after she uploaded it even though other projects in her region might be newer. Due to the location-awareness of modern smartphones not much additional effort would be necessary to implement such behavior. In 2010 18% of German children between the age of 8 and 13 had access to a phone with GPS sensor [Med10]. Thus geographical information can be sent together with the project during the upload process. If for some reason this information is not available

there is also the possibility to request the approximate location through the IP address of the uploading device. There exist several services which provide the mapping of specific IP address to a geographic location. Some of them are also available for free up to a defined limit of requests. One example would be HostIP.info[71] which also provides an API to easily request IP based location data. MaxMind GeoLite City[72] comes as a binary and provides the possibility to store the geo location data in a database on one's server. The data can be updated monthly.

This method could in the future be extended to showing this project also preferentially for a longer period of time to other children who are geographically related to the uploading person or who viewed projects of this user before. This will increase the chances that local as well as remote friends of a contributor will be able to see the project. Alerting users of new projects from their friends would also support this. Alerts can be geographically related too, e.g., notifying children when someone in the neighborhood uploads a new project. Sure the quality of this functionality would depend on how accurate the particular geo location data is.

6.5.4. Additional methods

To focus the user's attention on projects in the long tail a simple method would be a special "unknown projects" section where some randomly selected, less popular projects are presented for a certain period of time.

To enhance the quality of the search function and to provide a knowledge source for the recommender system which will be described in the upcoming section, a tagging system is planned. It should be possible to add tags before uploading a new project as well as for any project directly on the website.

[71] http://www.hostip.info

[72] http://www.maxmind.com/app/geolitecity

6.6. Recommender system

A highly-voted suggestion on the Scratch suggestion page concerns recommender systems. Similar to YouTube, recommended projects are presented to a particular user.

Recommender systems are best known from Amzon.com[73]. If signed in, the system makes recommendations what products might be interesting for a particular customer. What looks like an easy exercise from outside is based on sophisticated algorithms, mathematics, and a solid knowledge base in the background.

Types of recommendation systems can be distinguished by the source of the required knowledge. Collaborative systems use information about the rating behavior of a user; content-based systems generate recommendations based on particular features associated with products and the user's ratings for those products; demographic recommender systems base upon a demographic profile of the user and knowledge-based systems make suggestions based on a user's needs and preferences [Bur02] [Bur07].

For the purpose of the Catroid website there exist several possible scenarios. An example for a very basic system which could already be implemented right now (July 2011) would be to recommend other projects of the same author on a project's details page. Once there is the possibility to add other members as friends (see Section 6.2.4) another simple method would be to recommend their projects.

A simple to realize version of a demographic recommender system would be to suggest projects which are geographically related either to the provided home country or city of the particular user or to the current location. Information about country and/or city a user lives in is gathered during the registration process either as optional data on the registration page or automatically when the Catroid app is used for the registration. Alternatively

[73] http://www.amazon.com

the user's age and gender can be used to recommend projects created by users of similar age and same gender. An advantage of this method is that we could use information we already have without the need of acquiring additional data. A disadvantage could be that particular users – so called "grey-sheep" – could get recommendations of poor quality because they fall on a border between existing cliques of users [Bur02].

Once our system features a rating or voting system (see Section 6.2.2) the recorded data can be used to make collaborative recommendations. This type of system identifies peers who "loved" similar projects and recommends projects which are also "loved" by these users. The advantage of such a system is that no additional data about the projects is required. The disadvantage is the so called "cold-start" problem. New users who have not given any ratings so far will not receive any recommendations and newly uploaded projects could not be recommended to anyone until someone has voted for them [Bur07].

A more complex recommender system would be based on a tagging system and a folksonomy as described in Section 6.2.3 or a knowledge-based recommender. These knowledge bases and folksonomies require a lot of effort to be built. Thus this would only be an option for a later release. Identifying and acquiring tags for projects could be done though as this could be done more effortless and has advantages for other parts of the system like the search function as well.

To benefit from the advantages and to eliminate the disadvantages of different methods described above a combination of multiple recommendation techniques can be implemented. This combination is called a *hybrid recommender system*. There are several possibilities how to combine multiple techniques. The simplest design may be to weight the scores of each component for a particular project and combine them. This is called a *weighted hybrid system* [Bur07].

Recommended projects can be displayed at the bottom of a project's details page. Thereby the connection between the actual and the recommended project can also be shown, to explain the reason for the recommendation to the user. Another possibility is to add a special page where recommended projects for a particular user are shown in a list similar to the newest projects page.

6.7. Invitation system

One of our goals and a crucial success factor for our community is to quickly and continuously increase the number of members. Ideally this is achieved through viral growth without going to the time and effort of hard user acquisition.

In order to obtain viral growth it is necessary to optimize the recommendation and invitation process and create incentives for users who invite new members. Additionally the engagement of people with a widespread network of contacts – so called hubs – is a benefit during the acquisition process [End08].

In our case schools can act as hubs to quickly acquire an adequate number of users. Furthermore we will need various intuitive invitation tools on the community website as well as in the Catroid software itself. Such invitation systems are integrated in nearly every social software system and social network site, enabling children to send customized invitations to their friends and their peers. The integration of social networks described in Section 6.3 would make it easy to send Catroid invitation messages to friends in those networks. Another possibility would be to send invitations via email or SMS either from the community website or directly out of the Catroid software.

Users tend to recommend a service more likely if they clearly understand the benefits from doing so [End08]. Those can be the growth of one's personal network by automatically adding invited users as friends, certain rewards such as special ranks within the Catroid community, or special goodies and downloads. Recent research supports this approach. One motive for regular

content contributors to invite friends to join the Catroid community could be to extend the number of people to share their projects with.

Users who browse an online community but do not actually participate actively – so called *lurkers* – often make up the majority of members in an online community [Ted05] [Non01]. Regarding invitations, these users can be very useful too. Lurkers are always seeking opportunities to broaden their contacts [Ted05]. Thus they may be highly interested in acquiring new members.

6.8. HTML5 player and screen recorder

In the future Catroid will provide access to a wide range of phone sensors. It will, e.g., also be able to control a LEGO® robot or a Quadcopter using Catroid bricks. These facts make it difficult to develop a project player to be embedded in the website similar to the player on the Scratch community website. Nevertheless we plan to integrate some sort of HTML5 player for Catroid projects. At the moment it is not possible to support all of the phone sensors or to connect external devices with such a player but new features and techniques of HTML5 allow to support at least some basic functionality. Users do then have the possibility to view and try out projects directly on the website and do not have to download them first.

Another – less interactive – way to present a project is a YouTube video. While running the project on the phone it should be possible to record the screen and save it as a video. This video can be uploaded to YouTube and then displayed as embedded video on the details page of the project. Both possibilities – the HTML5 player and the YouTube video – are fancier and more attractive ways to display the project on the website than just showing a static screenshot as preview. Further it would be possible to embed the video or even the HTML5 player on other websites too. That could be on the personal homepage of a user or some sort of Facebook-App where members can watch the projects of their friends.

These advanced methods of presenting the projects could also support “viral marketing” as it lowers the bar for others to experience one’s games and animations.

6.9. Skinnability

Skinnability in the context of the Catroid community website means that children are able to customize parts of the site as they desire.

Personalization of information technology devices affects users cognitively, socially, and emotionally [Blo03]. While once the measure for the value of computer technology was its usefulness for solving problems, nowadays desirability, visual appeal, and aesthetics are important parts of design. Skins allow users to change the appearance of applications or websites. While the functionality still remains the same it may change the usability [Tra05].

Skinnability has become a common feature in recent computer related applications [Fis06]. Gmail[74] allows to choose from a large number of designs or to create an individual design for the Gmail account. Firefox Personas[75] enables users of the Firefox browser to customize the look and feel of the browser. Skinnability can also be programmable. WordPress[76] themes allow site owners to change the global styles of the site [Yee08].

Looking again at Scratch’s suggestion page shows clearly that kids love skinnability. The top voted suggestion regarding the Scratch community website deals with customizable userpages. The idea is to enable children to choose colors, pictures and projects to be displayed on their profile page. It is also thinkable to allow changing fonts, rearranging icons or including customized lists of projects. Another idea is to make the front page

[74] http://mail.google.com

[75] http://www.getpersonas.com

[76] http://wordpress.com

customizable similar to iGoogle[77] to let the children choose what they see on the start page.

To be able to compete the Catroid community website should somehow integrate support for customizing too. It is possible to implement minor features like a customizable avatar and some changeable areas on the profile page first and add extended features like customized project lists and advanced skin management later on.

6.10.Usability

Although Catroid should attract users from all ages it is especially intended for children from the age of 8. Thus it is important to consider that children's and teenagers' requirements for design and usability differ from those for adults.

Recent research by Jakob Nielsen and the Nielsen Norman Group targets usability issues when designing websites for kids. The studies involved children from 3 to 12 years and teenagers from 13 to 17 years. One important finding is that beside different requirements for adults and children they are also different for children and teenagers and between various age groups of children [Nie05] [Nie10].

Some examples of different browsing behavior between adults, teenagers and children are [Nie05] [Nie10]:

- The primary goal of children is entertainment while adults mostly want to get things done. Thus adults are more willing to wait for a limited amount of time for the website to respond while children want instant reaction and gratification.
- Children get discouraged by too much text, which is particularly true for younger kids with less experienced reading skills.

[77] http://www.google.at/ig

- Age-targeted design is crucial for children but it is important to make distinctions between age groups, especially between younger kids and teenagers. While children like animations and sound effects they could disaffect teenagers.

- The popularity of elements like back button, scrolling or font size can vary depending on the age group. The younger the users are the less back buttons and scrolling bars are used.

In general teenagers should never be addressed as kids. They react negatively when they have the feeling of being compared with children. Nielsen recommends having separate sections for children and teenagers on a website for young people, as teenagers do not want content which appears to be childish [Nie05].

Interactive features like games, the ability to share pictures or stories, message boards and forums, and content creation features are claimed to work well for teenagers. Teenagers want to "*[…] make their mark on the Internet and express themselves in various ways […]"* [Nie05].

These findings are good news for our purpose as the Catroid community website is intended to offer a wide range of those features and to allow children and teenagers to express themselves. Nevertheless success or failure also will depend on appropriate usability and design for the intended target age. As we also ask for the age of users during registration, we could also show the site in different ways to different age groups.

It is obvious that further research regarding the appropriate usability of the Catroid community website has to be done. Especially extensive usability tests and interviews with children and teenagers representing our target group will be crucial to identify potential weak points and problems with the design of the website.

6.11. Support for online collaboration

One of the primary aims of the Catroid project is to foster the collaboration of children and support them to cooperate with their peers in order to get inspired and learn from each other and to understand that collaboration and building on other people's work can lead to better results. To fulfill this goal we have to enable kids to collaborate at any place and any time through the Catroid online community. Internationalization and remixing are first steps to let the community website come up to this goal even though it is clear that further effort is necessary.

On the existing version of the Scratch community website it is possible to create so called "galleries" which were originally intended to group projects with similar topics. It turned out that children used these galleries very often to form collaborative groups calling them "companies". These companies consist of kids from diverse ages, cultures, and geographical locations and some of them are successfully developing complex projects [Mon09]. Nevertheless it appears that there is still a need for better support of online collaboration in the Scratch community. One possible improvement Scratch users voted for on the Scratch suggestion page is the possibility to make joint userpages for a team.

The Catroid community website will have to provide possibilities for children to form teams and to work together on one project regardless of where the team members come from and what language they speak.

7. Conclusions

The Catroid community website as well as the Catroid software is still in an early stage of development. Thus it is not yet ready to be tested extensively with participants representing our target group. Such testing would not make much sense as functionality, behavior, usability, and look may change at any time.

However, some basic functionality is already available. With the Catroid software it is possible to create low complex linear animations like music videos and comics as well as slightly interactive projects with simple user interaction using the conditional “if touched” brick or the “broadcast” mechanism.

The community website is ready to host projects created on and uploaded directly from the mobile device. With abuse-preventing mechanisms like cusswords-filters and “report as inappropriate” buttons it is ready to be released to a public mainly composed of children and teenagers. User management, search functionality, and the recently added internationalization bring the website closer to beta development stage. Further basic community features like user comments, votes and a tag- and friendship-system are currently under development or planned for the ongoing release.

It is difficult to draw conclusions on how the community website is being accepted before concrete tests have been run with real kids. This thesis surveyed relevant literature and analyzed best practices of related systems and projects similar to Catroid to identify general requirements our community website must meet in order to be successful.

- ➔ Minimize barriers and make the design most suitable for children.

  Children suitable design and usability is crucial. That includes an age-targeted design of the website as well as fluent borders between the Catroid software and the community website. Unnecessary barriers for joining and using the community as well as boring and unattractive design will disaffect children. One part of a solution is to make the website skinnable. This enables children to style parts of the site as they want.

- ➔ Make the website available in the children’s native languages.

Depending on the age and experience most children understand foreign languages poorly or not at all. Thus multi language support is extremely important for success. Every additional language will increase the number of potential users. Beside translations of static texts a template system for user contributed comments would lower international and intercultural barriers and foster collaboration.

➔ Clearly trigger the reactions of peers to the contributor.

Project contributors have to get feedback to stay motivated and to guarantee further participation. Votes, comments, and number of views, downloads, and remixes provide such feedback. This information must be clearly visible and easily accessible.

➔ Avoid parental concerns.

If parents have concerns about a website they prevent children from using it and from joining the community. Common parental concerns regarding the use of mobile Internet are phone overuse, high costs, and inappropriate content. Parents should be able to see that there are measures avoiding these concerns.

➔ Avoid disaffection caused by remixing.

The explanation of remixing and open source concepts must be easy to understand for children. Receiving manual credit from remixers is preferred by authors of the original project. Automatic attribution of related content is not sufficient to prevent complaints about plagiarism. Thus the design of the website and the software should slightly enforce manual crediting.

➔ Make the discovery of projects of one's interests as easy and intuitive as possible.

It is important to enable children to walk along the long tail in order to quickly find projects they are interested in. Powerful tools like a

recommender system and a collaborative tagging system can help to achieve this goal. A simpler method would be to show a list of projects which are geographically related to the website visitor.

➔ Include certain features to achieve viral growth.

Online communities are extremely vulnerable for cold start problems. Especially in the beginning it is important to include intuitive tools to allow children to send invitations. Further, so called hubs, e.g., schools, which have a widespread network of contacts, must be found to acquire an adequate number of users in the start phase. Integration of popular social network sites, the possibility to embed projects on other websites using a HTML5 player, and allowing uploading project videos to YouTube can help to achieve viral growth of the community.

The Catroid project is a long term project. Thus the development process will be evolutionary where new functionality will be added and unimportant functionality will be removed continuously. Similar to Scratch, Catroid can be seen as a platform to research new ideas and visions. Once a public beta version is published and the community has reached a significant number of members, additional topics for research will emerge. Possible future topics could be measuring the acceptance of existing or upcoming functionality, usability tests with children and teenagers, or methods to enhance participation.

List of figures

List of tables

References

[1UP09] 1UP games. (2009). *1UP.com*. Retrieved July 06, 2011, from http://www.1up.com/news/warioware-games

[ACM11] AC&M Consulting. (2011). *Prime-Tass*. Retrieved June 24, 2011, from http://www.bit.prime-tass.ru/news/show.asp?id=79606&ct=news

[Aki10] Akimoto, A. (2010, August 14). *asiajin.com*. Retrieved June 6, 2011, from http://asiajin.com/blog/2010/08/14/what-are-japans-top-5-social-services-heres-a-quick-overview

[And04] Anderson, C. (2004). *Wired*. Retrieved May 18, 2011, from http://www.wired.com/wired/archive/12.10/tail.html

[And10] Anderson, D. J. (2010). *Kanban.* Blue Hole Press.

[Ara10] Arabian Business Publishing Ltd. (2010). *arabianBusiness*. Retrieved June 24, 2011, from http://www.arabianbusiness.com/saudi-arabia-sees-mobile-broadband-users-increase-291245.html

[Bet10] Betazeta Networks S.A. (2010). *FayerWayer*. Retrieved June 24, 2011, from http://www.fayerwayer.com/2010/01/argentina-hay-mas-celulares-que-habitantes/

[Blo03] Bloom, J. O., & Monk, A. F. (2003, September). Theory of personalization of appearance: why users personalize their pcs and mobile phones. *Human-Computer Interaction , 18* (3), pp. 193-228.

[Bur02] Burke, R. (2002, November). Hybrid Recommender Systems: Survey and Experiments. *User Modeling and User-Adapted Interaction , 12* (4), pp. 331-370.

[Bur07] Burke, R. (2007). Hybrid web recommender systems. In P. Brusilovsky, A. Kobsa, & W. Nejdl (Eds.), *The adaptive web* (pp. 377-408). Berlin, Heidelberg: Springer-Verlag.

[Coo99] Cooper, A. (1999). *The Inmates are Running the Asylum.* Indianapolis, IN, USA: Macmillan Publishing Co., Inc.

[Cot00] Cothrel, J. P. (2000). Measuring the success of an online community. *Strategy and Leadership , 28* (3), pp. 17-21.

[CTI10] CTIA – The Wireless Association, International Association for the wireless telecommunication Industry. (2010). *CTIA Consumer Info*. Retrieved June 24, 2011, from http://www.ctia.org/consumer_info/index.cfm/AID/10323

[CUR10] CURE – Center for Usability Research & Engineering. (2010, April). M3 Scratch Personas (Unpublished document).

[DIT11] DIT Technologies. (2011). *D'Technology weblog*. Retrieved July 06, 2011, from http://www.ditii.com/2011/03/03/kodu-with-community-game-sharing-released/

[End08] Enders, A., Hungenberg, H., Denker, H.-P., & Mauch, S. (2008). The long tail of social networking: Revenue models of social networking sites. *European Management Journal , 26* (3), pp. 199-211.

[Far07] Farkas, M. G. (2007). *Social Software in Libraries: Building Collaboration, Communication, and Community Online.* Medford, NJ, USA: Information Today, Inc.

[Fer11] Ferguson, K. (2011). *Everything is a Remix*. Retrieved June 27, 2011, from http://www.everythingisaremix.info

[Fis06] Fishwick, P. A. (2006). *Aesthetic Computing.* The MIT Press.

[GSM10] GSM Association and Mobile Society Research Institute within NTT DOCOMO Inc. (2010). *Children's Use of Mobile Phones and Personal Relationships – An International Comparison 2010.* Japan: GSM Association.

[Har10] Harris Interactive Inc. Youth & Education Research. (2010, November). YouthPulse 2010. *Trends&Tudes , 9* (2).

[Hat11] Hatena Co., Ltd. (2011). *Flipnote Hatena*. Retrieved July 06, 2011, from http://flipnote.hatena.com/help

[Hil10] Hill, B. M., Monroy-Hernández, A., & Olson, K. R. (2010). Responses to Remixing on a Social Media Sharing Website. *Proc. ICWSM 2010* , pp. 74-81.

[Kol11] Koller, N. (2011). Achieving High Quality Software for Interactive Mobile Applications.

[Laz03] Lazar, J., & Preece, J. (2003). Social Considerations in Online Communities: Usability, Sociability, and Success Factors. In H. Van Oostendorp (Ed.), *Cognition in a Digital World* (pp. 127-151). London: Lawrence Erlbaum Associates.

[Lif11] Lifelong Kindergarten Group at the MIT Media Lab. (2011). *Scratch Stats*. Retrieved July 04, 2011, from http://stats.scratch.mit.edu/community/

[Mac11] MacLaurin, M. B. (2011). The design of kodu: a tiny visual programming language for children on the Xbox 360. *Proceedings of the 38th annual ACM SIGPLAN-SIGACT symposium on Principles of programming language* , pp. 241-246.

[Mac09] MacLaurin, M. (2009). Kodu: end-user programming and design for games. *Proceedings of the 4th International Conference on Foundations of Digital Games* .

[Mal04] Maloney, J., Burd, L., Kafai, Y., Rusk, N., Silverman, B., & Resnick, M. (2004). Scratch: A Sneak Preview. *Proceedings of the Second International Conference on Creating, Connecting, and Collaborating through Computing (January 2004)* , pp. 104-109.

[Man05] Manovich, L. (2005). *Remixing and Remixability.* Retrieved April 15, 2011, from http://imlportfolio.usc.edu/ctcs505/ManovichRemixModular.pdf

[Mar87] Markus, M. L. (1987, October). Toward a “Critical Mass” Theory of Interactive Media: Universal Access, Interdependence and Diffusion. *Communication Research , 14* (5), pp. 491-511.

[Mat04] Mathes, A. (2004). *Folksonomies - cooperative classification and communication through shared metadata.*

[Med10] Medienpädagogischer Forschungsverbund Südwest. (2010). *KIM-Studie 2010 Kinder + Medien + Computer + Internet.* Stuttgart: Medienpädagogischer Forschungsverbund Südwest.

[Mic10] Microsoft FUSE Labs. (2010). *KODU Game Lab*. Retrieved July 06, 2011, from http://fuse.microsoft.com/project/kodu.aspx

[Mon09] Monroy-Hernández, A. (2009). Designing a website for creative learning. *WebSci'09: Society On-Line* .

[Mon10] Monroy-Hernández, A., & Hill, B. M. (2010). Cooperation and attribution in an online community of young creators. *Poster in ACM CSCW'10* .

[Mon11] Monroy-Hernández, A., Hill, B. M., Gonzalez-Rivero, J., & Boyd, D. (2011). Computers can't Give Credit: How Automatic Attribution Falls Short in an Online Remixing Community. *Proceedings of the 2011 annual conference on Human factors in computing systems* .

[Nie10] Nielsen, J. (2010, September). *Children's Websites: Usability Issues in Designing for Kids*. Retrieved July 12, 2011, from useit.com Alertbox: http://www.useit.com/alertbox/children.html

[Nie05] Nielsen, J. (2005, January). *Usability of Websites for Teenagers*. Retrieved July 12, 2011, from useit.com Alertbox: http://www.useit.com/alertbox/teenagers.html

[Nin10] Nintendo. (2010). *WarioWare D.I.Y.* Retrieved July 06, 2011, from http://www.wariowarediy.com/do-it-yourself/

[Non01] Nonnecke, B., & Preece, J. (2001). Why lurkers lurk. *Proceeding of Seventh Americas Conference on Information Systems* .

[Ove04] Overmars, M. (2004, April). Teaching computer science through game design. *Computer , 37* (4), pp. 81-83.

[Pre00] Preece, J. (2000). *Online Communities: Designing Usability, Supporting Sociability.* Chichester, England: John Wiley & Sons.

[Pre01] Preece, J. (2001). Sociability and usability in online communities: Determining and measuring success. *Behaviour and Information Technology , 20* (5), pp. 347-356.

[Pre03] Preece, J., & Maloney-Krichmar, D. (2003). Online Communities: Focusing on sociability and usability. In J. Jacko, & A. A. Sears (Eds.), *Handbook of Human-Computer Interaction* (pp. 596-620). Mahwah, NJ: Lawrence Erlbaum Associates.

[Qua11] Quantcast Cooperation. (2011). *quantcast.com*. Retrieved June 6, 2011, from http://www.quantcast.com

[Raf97] Rafaeli, S., & Sudweeks, F. (1997). Networked interactivity. *Journal of Computer-Mediated Communication , 2* (4).

[Rav08] Ravid, G., Kalman, Y. M., & Rafaeli, S. (2008). Wikibooks in higher education: Empowerment through online distributed collaboration. *Computers in Human Behavior , 24* (5).

[Res96] Resnick, M., & Rusk, N. (1996). The computer clubhouse: preparing for life in a digital world. *IBM Systems Journal , 35* (3.4), pp. 431-439.

[Res05] Resnick, M., & Silverman, B. (2005). Some reflections on designing construction kits for kids. *Proceedings of the 2005 Conference on Interaction Design and Children* , pp. 117-122.

[Rhe94] Rheingold, H. (1994). A Slice of Life in My Virtual Community. In L. M. Harasim (Ed.), *Global Networks: Computers and International Communication* (pp. 57-80). Cambridge, MA: MIT Press.

[Rhe93] Rheingold, H. (1993). *The Virtual Community.* Massachusetts: Addison-Wesley.

[She08] Shepitsen, A., Gemmell, J., Mobasher, B., & Burke, R. (2008). Personalized recommendation in social tagging systems using hierarchical clustering. *Proceedings of the 2008 ACM conference on Recommender systems* , pp. 259-266.

[Stu05] Stump, J. (2005). *Understanding MVC in PHP*. Retrieved May 18, 2011, from http://oreilly.com/pub/a/php/archive/mvc-intro.html

[Tec09] Technology Merketing Corporation. (2009). *TMCnet*. Retrieved June 24, 2011, from http://www.tmcnet.com/usubmit/2009/06/12/4222607.htm

[Ted05] Tedjamulia, S. J., Olsen, D. R., Dean, D. L., & Albrecht, C. C. (2005). Motivating Content Contributions to Online Communities: Toward a More Comprehensive Theory. *Proceedings of the 38th Hawaii International Conference on System Sciences* .

[Tho11] Thomson Reuters. (2011). *Reuters*. Retrieved June 24, 2011, from http://www.reuters.com/article/2011/06/20/china-mobile-idusl4e7gk0bo20110620

[Tra05] Tractinsky, N. (2005). Does Aesthetics Matter in Human-Computer Interaction? In C. Stary (Ed.), *Mensch & Computer 2005: Kunst und Wissenschaft – Grenzüberschreitungen der interaktiven ART* (pp. 29-42). München: Oldenbourg Verlag.

[Yam10] Yamakami, T. (2010, August). A stage model of mobile video services: Implications from emerging data services in Japan. *6th International Conference on Digital Content, Multimedia Technology and its Applications (IDC)* , pp. 208-212.

[Yee08] Yee, R. (2008). *Pro Web 2.0 Mashups: Remixing Data and Web Services.* APress.

[YoY11] YoYo Games Ltd. (2011). *YOYOGames*. Retrieved July 06, 2011, from http://www.yoyogames.com/help/about

[Zuc09] Zuckerman, O., Blau, I., & Monroy-Hernández, A. (2009). Children's Participation Patterns in Online Communities: An Analysis of Israeli Learners in the Scratch Online Community. *Interdisciplinary Journal of E-Learning and Learning Objects , 5*, pp. 263-274.