



**University of
Zurich^{UZH}**



Visual Browsing of Mobile Apps

Bachelor's Thesis
supervised by the

Department of Informatics at the University of Zurich

Elaine M. Huang, Ph.D

to obtain the degree of
Bachelor of Science UZH (in Business Information Systems)

Author: Stephane Rufer
Course of Studies: Business Information Systems
Student ID: 08-274-219
Address: Binzmühlestrasse 14
8050 Zürich
E-Mail: stephane.rufer@uzh.ch
Closing date: December 18, 2012
Supervisor: Elaine M. Huang

Abstract

This thesis focuses on an alternate form of app data visualization. Traditionally category centric designs (e.g. top lists) are used to discover new apps. In the following we look at related work in the fields of exploration of digital libraries, music visualizations, mobile communication and to some extent app discovery, to elicit the important features and affordances needed for app discovery.

With this analysis as a backdrop, many forms of visualizations were brainstormed, of which three were prototyped as wireframes and shown to potential users. This informal evaluation enlightened the decision of which visualization to implement. The final implementation (AppDJ) was evaluated in a cognitive walkthrough and then released on the Google Play Store. Following this, a user study of AppDJ was conducted and app analytics data was collected.

The evaluation of both the user questionnaire and the analytics uncovered several interesting aspects. It was found that AppDJ is not only a simple and easy visualization, but it support serendipitous discovery as well as driving installs down the long tail. Furthermore, the data collected on app market clicks gave rise to the notion of an App Adoption Cycle.

Acknowledgments

I would like to thank Dr. Elaine Huang for suggesting this topic and allowing me to shape it according to my interests. It was a great experience to be able to work together with a startup, develop and release my own app.

I would also like to thank the whole ZPAC group for their feedback and support. I extend my gratitude to the guys from 42matters AG (Andrea Girardello, Andreas Budde, Bo Wang and Ivan Delchev) for helping me with various Android related problems that arose during the implementation, as well as the insightful discussions. Additionally, a big thanks to Corneliu Margine for providing some of his analytics data and the many comments and discussions that aided in pushing my thoughts in interesting directions.

Last but not least, I would like to thank my friends and family for their support and especially their assistance in testing the app in the early stages. A special thanks goes to my brother and sister who, as my quality assurance team, found and squashed many bugs before the app was release on the market.

Contents

1	Introduction	1
2	Related Work	3
2.1	Discovery in Digital Libraries	3
2.1.1	Exploration and Immersion	3
2.1.2	Exploration and Discovery in Music Applications	5
2.2	App Discovery	6
2.2.1	Existing Discovery Solutions	6
2.2.2	Important Factors	8
3	Visualization Prototyping	11
3.1	Early Design Ideas	11
3.2	Wireframe Prototypes	15
3.2.1	Mood Agent	15
3.2.2	Tag Cloud	17
3.2.3	App Matcher	19
3.3	Informal User Evaluation	20
3.3.1	Mood Agent	20
3.3.2	Tag Cloud	21
3.3.3	App Matcher	21
4	AppDJ	23
4.1	Design	24
4.2	Technical Details	29
4.3	Recommender System	30
4.3.1	App Traits	31
4.3.2	Variable Dimensions	32
4.3.3	Category Dimensions	33
4.4	Implementation	34
4.4.1	Dimension to Slider Mapping	34
4.4.2	Recommender Algorithm	35
4.4.3	Variable Dimension API Interface	35
4.4.4	Category Dimensions API Interface	35
4.4.5	Recommender System Process	38
4.5	Issues and Problems	39
5	Evaluation	41
5.1	Prerelease Evaluation	41
5.1.1	Cognitive Walkthrough	41

5.1.2	Informal User Feedback	42
5.1.3	Changes and Additions	42
5.2	Deployment and Field Evaluation	43
5.2.1	App Analytics	43
5.2.2	Minor Improvements	50
5.2.3	User Evaluation	51
6	Discussion	53
6.1	User Response	53
6.2	Longtail App Discovery	57
6.3	App Adoption Cycle	58
7	Conclusion	65
8	Future Work	69
9	References	71
Appendix A	Methods of App Discovery	75
Appendix B	Preliminary Visualizations	76
B.1	Step 1: Starting Point	76
B.1.1	Simple Search	76
B.1.2	Advanced Search	76
B.1.3	Collage	76
B.1.4	TreeMap	77
B.1.5	What's New	77
B.2	Step 2: Exploring Apps	78
B.2.1	Simple Graph	78
B.2.2	Social Graph	78
B.2.3	Advanced Social Graph	79
B.2.4	TreeMap Categories	80
B.2.5	Ribbons	81
B.2.6	Flipboard	81
Appendix C	Brainstorming Visualizations	82
C.1	Designs	82
C.1.1	Radar	82
C.1.2	Wheel of Fortune	83
C.1.3	Clouds	83
C.1.4	Clothes Hanger	83

C.1.5 Slot Machine	84
C.1.6 “Spotify Radio”	84
C.1.7 Boxes/Crates	84
C.1.8 Poster Roll	85
C.1.9 Social Nests	85
C.1.10 Bar	86
C.1.11 Photo Album	86
C.1.12 Concert	87
C.2 Excursus into Gamification	87
Appendix D Slider Dimensions	88
Appendix E Cognitive Walkthrough	91
Appendix F Analytics	100
Appendix G User Evaluation	101
G.1 Questions	101
G.2 Responses	102
Appendix H App Adoption Cycle	117

List of Figures

1	Brainstorming for visualization ideas	12
2	Results of brainstorming	13
3	Ideas developed from brainstorming and analysis	14
4	Results of the group brainstorming	15
5	Mood Agent wireframe start screen and screen showed when not logged in	16
6	Mood Agent wireframe screen showed when logged in	16
7	Tag Cloud start screen	17
8	Tag Cloud screen when tags are active	18
9	App Matcher start screen	19
10	App Matcher comparison screen	20
11	The AppDJ wireframe integrating ideas from the Mood Agent visualization	25
12	Android UI wireframe of the AppDJ visualization	26
13	Final UI of the AppDJ Android app running on a tablet	28
14	Final UI of the AppDJ Android app running on a mobile phone	28
15	The process description of transforming the slider profile into requests that fill the list of apps displayed to the user	39
16	The engagement flow of AppDJ	44
17	Distribution of download categories	49
18	Total session duration compared to sessions resulting in market clicks, overlaid with number of market clicks per session range (y axis on the right shows the market clicks as well as session times resulting in market clicks)	50
19	User responses to the three likert scale questions	51
20	AppDJ market clicks compared to AppAware	58
21	The number of apps per Google Play download category	60
22	The adoption cycle of apps	61
23	A possible search visualization	76
24	What a Spotify style start screen could look like	77
25	A simple node link graph visualization	78
26	The social and app graph integrated into one visualization	79
27	Expanded social graph visualization	79
28	A TreeMap visualization focusing on apps	80
29	Visualization using ribbons	81
30	App Matcher comparison screen	82
31	Radar visualization that categories apps	82
32	Wheel of fortune visualization that emphasizes randomness	83

33	Clouds of similar apps floating across the screen that takes a lean back approach	83
34	A similar lean back approach with app hangers visualized as cloths hangers	83
35	A random gameified approach in the form of a slot machine	84
36	A visualization based on a recommender system that allows user tweaking	84
37	A category centeric approach to browsing apps	85
38	Apps are placed on a poster roll in a random collage visualization	85
39	An approach focusing on social groups	85
40	A social grouping approach based on the analogy of a bar	86
41	A collage based visualization integrating category grouping	86
42	A social visualization that integrates social and app graphs	87
43	Possible ideas for games	87
44	Traits rank per category	89
45	Category ranking across dimensions	90
46	Market click distribution across categories	100
47	Distribution of ratings based on app market clicks in AppDJ	101
48	Apps ordered by absolute number of market clicks	101
49	The average number of apps per download category in the popularity and trending list	117
50	The Gartner Hype Cycle	117

List of Tables

1	Slider levels are attributed to the their corresponding category dimension	34
2	An example of the different stages of the App Adoption Cycle	63
3	How users discover apps	75

List of Algorithms

1	Finds the total weight of all categories, giving category intersections more significance	36
2	Set the number of apps to query for per category	37
3	Fills up the list with apps from other categories in the slider selection . .	38

1 Introduction

The real voyage of discovery consists not in seeking new landscapes, but in having new eyes.

Marcel Proust

Over the past five years, the growth of the market for mobile apps has indubitably exploded. In 2009, when the smartphone market was still in its infancy, global app downloads were roughly 7 billion [23]. In September 2012, Google announced that the 25 billion app downloads milestone had been surpassed [16], less than one year after reaching 10 billion downloads [30]. Not only has the number of downloads risen exponentially, but the market has become increasingly crowded as more developers move into the mobile app space. In late 2010 there were more than 100,000 apps on the market, which rose to 250,000 in July 2011 [13], and almost tripled by September 2012 [16]. This illustrates just how congested and multifaceted the app market has become, forging the daunting and challenging task of app discovery, for developers as well as consumers. Finding the right app for a specific task is fast becoming the Holy Grail in the world of apps, as the majority of the time spent on mobile phones is spent using apps [19].

Alongside this spate of apps, the new market of discovery tools has emerged. There are a multitude of different avenues perused by various apps and discovery frameworks, all attempting to provide the perfect solution for both developers to promote their apps and consumers to find the apps that suit their needs. For many of these apps, the focus is on the act of discovery and not on the path that leads to it. The category centric approach to finding apps and variants of this are numerous. These solutions do not necessarily address the issues confronted by non power users and many introduce new discovery problems themselves, as algorithms fail to scale in the face of the app onslaught. This thesis addresses the issues present in many of the currently available tools, as well as propose a new form of visualization that reduces the complexity of app discovery, making it easier for new and non expert users. The ultimate goal of the visualization is to illuminate the path of discovery and change the perspective, helping people see the crowded world of apps in a new light. A lean-back approach focusing on serendipity should assist users in finding previously undiscovered apps, essentially giving people new eyes through which to see the massive, and continually growing market of apps.

To begin, the review of related work looks at discovery in digital libraries and user

behaviors pertaining to smartphones. Additionally, implementations of visualizations in the music industry that aid discovery of new music are examined and the concepts are applied to the domain of apps. To elicit the proper visualization that addresses the core issues and needs of the users, one must first determine the traits and characteristics important for app discovery. In a first step existing discovery tools are briefly investigated. From this the factors that are important to app discovery are obtained, which create the elementary foundation on which design ideas can be brainstormed. After conceptualizing a number of design ideas, three were chosen to be further developed. These visualizations were then presented to a group of potential users that evaluated and commented on each one, which was used to determine the visualization most befitting the needs of the target users. Out of these visualizations the final AppDJ visualization was developed and implemented as an Android app. Preceding the release of the app, a cognitive walkthrough was conducted to ascertain the usability of the app as well as determining the weak design points. The app was publicly released on the Google Play Store and in the space of two months AppDJ was downloaded over 13,000 times, with an average of more than 300 daily active users. Following the deployment, the usage analytics collected by the app were analyzed and evaluated. In order to confirm the success of the visualization, a user study of the active AppDJ users was conducted to decipher the goals of users as well as what they considered as the positive and negative aspects of AppDJ. Following this, the findings from the analytics are discussed in the context of the user responses, showing that the AppDJ visualization affords a novel form of discovery. Finally, the analysis of the data in comparison to more category centric visualizations gave rise to a new concept, dubbed the App Adoption Cycle.

The results demonstrate that the challenges for app discovery are both diverse and complex. AppDJ addresses a new area of discovery, but still leaves many aspects uncovered. While the data does support the notion of an App Adoption Cycle, conclusive application to singular apps is still lacking. AppDJ proposes an alluring concept for an alternate visualization, opening up the field of app discovery with new insights.

2 Related Work

2.1 Discovery in Digital Libraries

With increasing influence of technology in our lives, our reliance and the importance of digital libraries has risen. Most of the content that we consume on a daily basis has been digitalized into libraries available online, of which the mobile app market is just a small, emerging player. As a result, the design of these libraries has become more important. Xie [27] mentions that the design of digital libraries as well as the available, or unavailable physical affordances has a direct effect on how users interact with, and search within the library. She states that one unique aspect of a digital library is its browsing function. It allows the user to explore and engage with the library in a multitude of ways, which are not possible with their analog counterpart. In her study comparing the usage of two different digital libraries, Xie [27] found that the subjects interacted differently based on the features and design of the tools. In one tool users could only browse for subjects within one category, which prompted users to browse through multiple categories to find the desired information. This can be applied to the context of app discovery, in that users have a clear idea of a certain app they would like to install, but currently must search through different categories in order to find the perfect match. Xie [27] also remarks that users interact with libraries in diverse ways, indicating that these should not only support different levels of knowledge and skills, but also varying approaches to discovery. Interestingly the usability of the interface and performance were deemed as more important by participants than the quality of the collection of data in the library.

The context of mobile phones further influences the ways people attempt to access information. Church and Smyth [8] found that when using mobile phones for search queries, user behavior and the way in which they phrase their query is different. For mobile users the context and their surrounding environment plays into the things they would like to find, as well as how they go about finding them.

2.1.1 Exploration and Immersion

The smartphone market has grown tremendously over the last years, with almost half of U.S mobile subscribers owning a smartphone in 2012 compared to 36% in 2011 [20]. A recent Nielsen report [19] found that 64% of time spent on mobile phones is spent in apps. This shows just how important the app ecosystem has become, and as a result the ability to find the right app for a certain need.

The question that remains open is what kinds of apps people install and use, as well as the accompanying indicators that can be used by developers and app marketplaces to predict users preferences for apps in order to provide more detailed recommendations. Falaki et al. [10] look into how people use their smartphones in terms of interactions with the phone, application use, network traffic and energy drain. The interactions and application usage dimensions of specific smartphone usage is particularly interesting in the context of this thesis. The findings create the analytic foundation on which important characteristics for discovery apps can be elucidated and used to build a tool that helps people optimally find the apps that they desire. Falaki et al. [10] found that the users differ significantly, where some use many apps with short interaction cycles, others use few apps but for longer periods of time. Additionally, the authors state that demographics are not a good indicator of smartphone usage. They found no statistically significant behavioral differences between age groups. Although high school students are the primary users of media and communication apps and business people frequently interact with productivity apps, behavioral predictions based on this categorization are unreliable. This shows the high degree of diversity present in the smartphone market. This stands at odds with the app discovery market, where the process of finding apps is relatively standardized and geared towards a “lean-forward” approach. The range of users is such that an approach that only considers the “average” user will not work for most of the population. For all their differences, Falaki et al. [10] found that similarities exists between users, of which the most reliable discriminator for smartphone usage was session time of apps.

Broder [7] created a taxonomy for web searches, segmenting queries into three classes. The intent behind each query is a different one, where navigational queries are executed to reach a particular site, informational ones to gather information and transactional queries intend to perform some mediated activity where subsequent interactions occur. When users are searching for apps, the context can also play a role in the search procedure undertaken. Viewing the search for a mobile app in this way can aid the creation of tools that address how people go about discovering and searching.

North [22] defines insight in terms of six characteristics, of which “relevant” applies to the simplicity and “unexpected” to the serendipity of the visualization. North’s [22] idea that building insight takes time, reinforces the importance of providing affordance to exploration and immersion. Yi et al. state the following about insight: “We also believe that insight is often not an achievable goal of a predefined task or procedure, but more

likely a by-product of exploration without an initial destination” [29, pg 2]. This shows how important exploration is to gain insight and the role of accidental discovery in the process of gaining understanding. Going further, Yi et al. [29] state the importance of the iterative process applied when making sense of data. In terms of app discovery this means that an iterative approach could potentially help users understand and process information better to find the right app. They found that there are four categorizations of procedures that play into gaining insight. “Providing overview” plays a role in helping people understand the global context of the data, and precedes exploration. Following this, people adjust the visible data set through a process called “adjust”, where interaction through grouping and filtering play an important role. Finally, people “detect patterns” through which they gain insight and new knowledge about the information. The process of detecting patterns is less important for users discovering apps, as their ultimate goal is to find a specific app and less the relationships between apps. The last step is “match mental model”, where the linking of the visualization with real world phenomena reduces the cognitive load of understanding the data set. This is particularly pertinent to app discovery, as the wealth of different apps creates an environment of increasing complexity.

2.1.2 Exploration and Discovery in Music Applications

There is a large body of work that looks into the exploration and discovery of content in music applications. Many of the elements of these tools as well as the findings can be applied to the app market.

Many music discovery tools use contextual information to help users navigate their music libraries [12]. The method used often employs fuzzy and non specific criteria to navigate to a final piece of information that conforms to the specifications set by the user. The effort on the side of the user is minimal and the result is relatively accurate, but has an element of serendipity. Another important factor, especially with mobile applications is scalability, as addressed by Dachsel and Frisch [9].

The social element involved in the interaction with libraries is addressed by Holmberg and Torrens [24]. With Musicstrands they provide a way to share the discovery path a user follows. The authors mention the two behaviors people show when exploring digital libraries. Some search for a specific piece of information they would like to find, while others discover without knowing exactly what they are looking for. Holmberg and Torrens [24] focus on providing serendipitous discovery by design, where the user benefits from painlessly being able to explore and find new music. To aid the user in such discovery, they state that visualizations should not only provide an overview, but

also enough detail and context to provide meaningful suggestions and information. Moreover, clean and simple visualization aids users in the exploration process, allowing them concentrate on the essence of discovery [25].

Bainbridge et al. [4] indicate the importance of “recognition” versus “recall” mentioned in other HCI, cognitive science and psychology of memory literature. As a result a “lean-back” approach to app discovery poses as an interesting concept. Moreover, the authors introduce the method of discovery where the user flips through music as they would in a record store. Such an analogy can be leveraged for a digital tool to support and encourage natural discovery.

2.2 App Discovery

App discovery is a ubiquitous problem for all mobile platforms [1]. There are many solutions that attempt to address the various problems that arise with app discovery. In the following section we will look at some of the existing solutions and how they aim to make app discovery easier. The issues and undressed problems of some visualizations are shortly explained.

2.2.1 Existing Discovery Solutions

In principle there are three basic approaches to app discovery:

- **Search:** The user inputs a phrase or search keywords that correspond to the problem or need they want to resolve. Searching is very specific and is only useful for the user if they know exactly what they are looking for. People need to have a search keyword in mind, which means they actively “find” the app instead of “discovering” it.
- **Recommendation:** Recommendation is more “lean-back” and random, in that the user does not actively have to do something to discover apps. The caveat is that these systems recommend similar apps, so the user never comes into contact with other novel apps that might be of value. Additionally, there is a need for a “taste profile”, which means that the quality of the recommendation is only given once the user has interacted with the application for some time.
- **Topic Centric:** The market of apps is segmented into categories. This reduces the burden on the user, because they can browse through apps based on topics. The issue here is that some apps can belong to different categories, which

means that search and browsing requires looking through multiple groups of apps. Moreover, not all apps fall neatly into the predefined categories, which reduces the accuracy of these groupings.

Discover Apps

A simple user interface, where the user searches for a “seed” app and then can explore the app graph of related applications. The graph is a simple connection of nodes (applications) with one type of edge (relationship, sameness). There are two entry points, one of which is searching for a specific app. The other is using a general “top list” that can be used to explore trending apps. There is also the ability to share the discovered app over various social channels. The problem with this form of discovery is the lack of filtering capabilities and the analytical form of the visualization. A graph is a good tool for the visualization of the topography of the environment in which an app resides, but is not necessarily the most natural and intuitive way for users to discover new apps.

Facebook App Center

The Facebook App Center is an app store that integrates the social graph of Facebook together with apps. The App Center shows which apps are used and installed by which friends. There is the ability to explore and find new apps over the Facebook platform using classic top lists and browsing. Once an app is found it can be sent directly to the mobile for installation. The Facebook App Center integrates a wide variety of apps ranging from Facebook, mobile to HTML5 apps. The novelty of the Facebook App Center, is that unlike other app stores, it does not rely on traditional metrics like installs. Instead it shows apps based on the user’s interest and habits.

Chip Online App Guide

This is a simple guide for what apps are recommended by people and the reviews associated with the apps. This allows users to discover apps according to human recommendations and reviews. The users can see what the “test” score of apps was, giving them some sort of quality seal.

Yahoo AppSpot

This service combines app searching, browsing and reviews. The idea is not very novel, with no social integration or app recommendations based on an app “taste profile”. To download an app a QR Code has to be scanned, which leads to a redirect to the app store where the app can be downloaded. Yahoo AppSpot is more of an aggregator that gives a full overview of the apps available on the market.

AppBrain/AppFire

Both markets cater to a special crowd (AppBrain is for Android, AppFire for iOS). They integrate app discovery and installation, so the experience is seamless. Apps are explored by viewing different top lists and categories. The USP (unique selling proposition) of these tools are that they are able to give more precise recommendations based on installs and usage.

StumbleUpon

The random algorithm of StumbleUpon allows people to “stumble” across new apps. The approach is the same as the original StumbleUpon with websites.

The issue with all these approaches is that they don't drive app discovery down the long tail. For an app to gain traction it needs to make it into one of the top lists. Otherwise developing mindshare is difficult if not impossible. The design of app markets has a direct effect on which app marketing tools are the most effective and prevalent throughout the industry. Word of mouth, for example is still one of the largest drivers of app installs. A tool that provides better presence for niche developers is one that flattens the layers of app discovery.

2.2.2 Important Factors

App discovery in an app store is no different than finding things in others stores. The analogy of a store is a good way to describe the experience, people use a directory to find what they want, they window shop, look at the details of a product, compare and then finally make the buying decision. Categories serve as a good way for users to navigate the market, much like a store directory. For window shopping, mannequins and display windows are important, which translates to the app name, icon as well as screenshots in the app world. Finally, the app description as well as pricing, reviews and rating play into the final download decision.

When people search for new apps and decide on installing, they look at certain aspects. Boswell [5] finds that the following factors play a key role in discovery and subsequent install:

- **Localization:** The language of the description in the app store can help to boost discovery. Many users will prefer to read an app description in their native language, even if the app itself is not translated.

- **Rating:** People use this as an indicator for the quality and usefulness of the app. Typically apps with a rating below 3 out of 5 stars will have substantial difficulties with user acquisition.
- **Reviews:** As with ratings, people use this as a litmus test. Reviews allow users to rate the app in a more detailed manner than the limited character of ratings as feedback. As an extension to the in store reviews, blogs such as Lifehacker¹ and sites like Chip Online² weighing in and evaluating an app, can also boost installs and visibility. Bouchard [6] also mentions blogger outreach to be one of the keys to successfully promoting an app and increasing the number of users discovering it. The more positive the reviews the more likely it is that people will install the app. Conversely, just a few negative reviews can be a significant hindrance.
- **Categories:** In the world of apps categories have a similar function to genre in music. People looking for a specific type of app will browse in the corresponding categories.
- **Recommendations:** The social component to app discovery is also important. Both recommendations over digital social networks as well as the traditional word of mouth are decisive in aiding discoverability.
- **Pricing:** Many users filter their results by the price of an app. Pricing that considers the target audience helps those discover the app that have a need for it and would be willing to pay the asking price.

Moreover, Boswell [5] notes the importance of app store optimization, a process described in detail by Graaf [26]. The five key elements that need to be optimized are the app name, app publisher, keywords and description of the app, installs as well as ratings and reviews. Graaf [26] states that installs play a key role in conveying trust, but that not just absolute installs but also install trends are essential to growth and continued success. As such, installs serve as a form of implicit “recommendation”, where many installs show that an app is legitimate and useful. The description of an app is not only used by search algorithms to elicit keywords [26], but also serves as additional information for users to determine if the app suits their needs. Like any marketing message it should be short and to the point, clearly state the unique selling proposition and concentrate on the core features. Finally, the app name is the most prominent feature besides the logo and should both catch the user’s attention as well as allow him/her to decide if the further details are worth looking at. The name and logo are like a first impression, if they are effective, they communicate the function of the app.

¹ <http://lifehacker.com/>

² <http://www.chip.de/>

3 Visualization Prototyping

As more developers push into the mobile development space, the discovery mechanism users adopt, and optimally targeting users in this process becomes increasingly important. The discourse on app discovery is ample, with some seeing discovery as more of a developer problem [14]. While it is primarily an issue for developers, weeding out the good apps and the ones suited to the problem at hand steadily becomes more challenging for users as more apps push into the market. For developers the costs of inefficient app discovery has a direct monetary influence, because their apps are not being installed at a rate that would be warranted by the quality of the app. On the other hand, the cost for the user is implicit and hidden, as they loose in terms of productivity or enjoyment, because the app they found and installed is not the one that optimally suites their current needs. As a result, discovery is not just a marketing problem on the supply side, but an equally salient quandary on the demand side.

Properly addressing user issues with discovery should be given more attention, especially when one considers the failure rate of apps. According to a study of the still nascent app market by Localytics in 2010 [15], over a quarter of all apps downloaded were used just once. How high usage is beyond that is still unknown. Margine [18] found that user retention³ over three months is between 90% and 60%, depending on the category. This is an indication of the increasing importance discovery and app exploration solutions will have in the future. Before beginning with the brainstorming for visualizations of app data, we looked at what aspects are important to users when discovering apps (section Important Factors) and how they go about discovery (Appendix A).

3.1 Early Design Ideas

After analyzing what factors are important to users and how they go about discovery, some initial ideas were contrived and developed. These ideas turned out to be a good starting point, but did not conclusively address the issues and features important to users. These preliminary ideas are described in more detail in Appendix B, considering the fact that they build the foundation on which the subsequent visualizations were conceived. To open the design space and think outside the box, a brainstorming session was conducted. In this phase all ideas that came to mind where posted on a whiteboard, regardless of if they fit into the concept or not. These

³ These are people that still have the app installed after three months. Accordingly, it could be that these people are not actively using the app, although they still have it installed

sketched visualizations were then evaluated based on the criteria in the section Important Factors. The visualizations that seemed plausible as well as interesting for users were elaborated and discussed in the group. From this discussion three visualizations were chosen to be iterated on and further developed into more mature wireframe prototypes. These prototypes were then presented to potential users in an informal setting, in order to determine which would provide the best abstraction for users in respect to the goal of providing a “lean-back” approach to app discovery.

Brainstorming



Figure 1: Brainstorming for visualization ideas

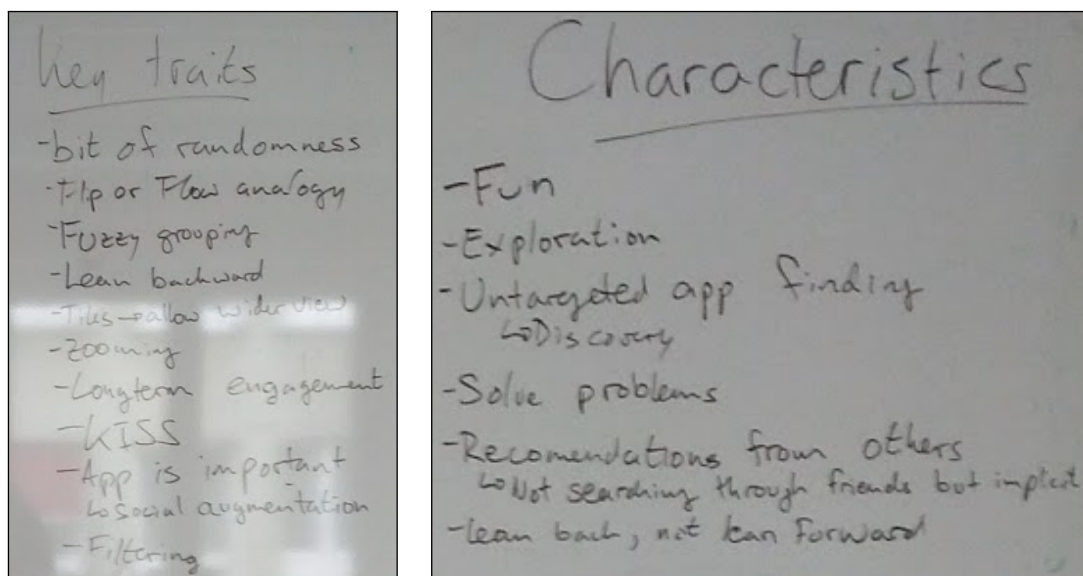
In a first step all ideas were written on Post-its and stuck randomly on a whiteboard. These ideas ranged from games like Wheel of Fortune, to the analogy of a highway with exits or clothes hangers. Additionally, simple design concepts, such as tiles and pin walls were included as ideas for how some of these analogies could be effectively be visualized. The complete set of ideas can be seen in Figure 30 in Appendix C. Following this brainstorming, the Post-its were clustered into groups according to what they represented (see Figure 1). This was done by using affinity diagramming⁴. Concrete ideas were separated from concepts and analogies. Once an overview of all the ideas was achieved, the clusters were organized by the degree of organization

⁴ MIZUNO, S. Management for Quality Improvement: The 7 New QC Tools. *Productivity Press*. 1988.

and categorization that the visualization allows on the x axis. The y axis represents a very rough categorization into ZUI (zoomable GUIs), coverflow (representations with flowing images) and tiles (grid view like representations). This allowed the definition of key characteristics and traits that are the foundations on which the final ideas build, which were then developed into visualizations.

Key Traits and Characteristics

Keeping in mind the factors that are important for users when discovering apps, as detailed in section Important Factors, and the ideas and concepts developed during the brainstorming phase, key traits and characteristics were elicited. These are aspects that are common across many of the themes that sprung out of the early conceptual phase. Figure 2 shows the elements that should be included in the potential visualizations. Ultimately not all of the listed traits or characteristics could be incorporated into one single visualization, but all of them were used to guide the process of creating substantive visualization ideas, which in turn were the basis for the final prototype. In the process of brainstorming visualizations from these ideas, lean-back, fun and exploration as listed in Figure 2b became key elements. Of the traits in Figure 2a, randomness, simplicity and fuzzy grouping represented the focus in most visualizations. In the first iteration social augmentation (Figure 2a) was still a central element. After showing the first wireframes to potential users this facet was dropped. Many people found the social dimension to be less interesting and could not discern a tangible benefit of social integration. The integration of Facebook would have added additional complexity as well as requiring more permissions, which in turn could stave off user adoption.



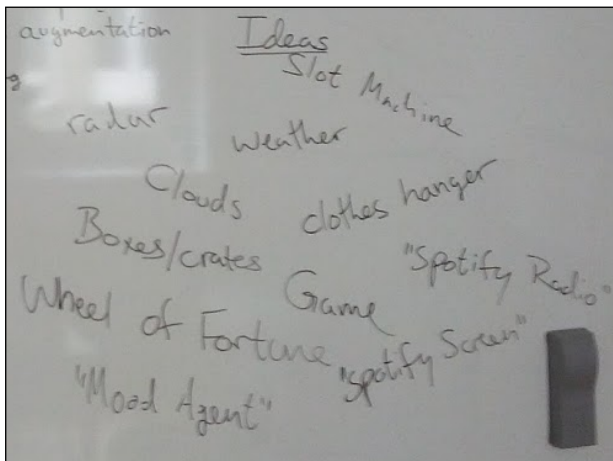
(a) Key traits

(b) Characteristics

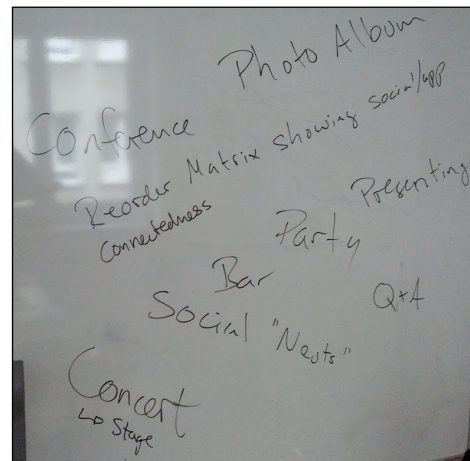
Figure 2: Results of brainstorming

Resulting Ideas

In a final step, all the ideas generated from brainstorming and the resulting syndication of traits and characteristics were combined and formed into an explicit design ideas. Aside from looking into conventional visualizations, the aspect of gamification was explored. The results of this foray into games is exhibited in Figure 43 in Appendix C. The gamification concept was later drop in favor of a more coherent approach that allows immersion and exploration. The basic idea of providing a visualization that conveys a “fun” experience remained. Figure 3a shows some of the first concepts that surfaced directly out of the aforementioned brainstorming. While these basic ideas were further developed with sketches, some other novel approaches arose (Figure 3b). These ideas aimed to expand the design space with more generic analogies. All of these designs, in addition to the ones listed in Figure 3 are described in more detail in Appendix C.



(a) Generated ideas



(b) Additional concepts and analogies

Figure 3: Ideas developed from brainstorming and analysis

Once all these potential visualizations were sketched out, they were presented and iterated on in the group. The results of this group brainstorming can be seen in Figure 4. Out of all these, three equally feasible visualizations were chosen to be wireframed in more detail. These visualizations were the only ones that were presented to potential users. With the help of this informal feedback, one of the design ideas was chosen, from which a working prototype was created.

As can be seen in Figure 4, gamification was a central topic driving the brainstorming sessions. In fact, the visualization that was later chosen as the basis for the implemented prototype (Mood Agent) was conceptualized as a game. Actually, the App Matcher idea was born out of the discussion of how to integrate a social component into a game for discovering apps. This was one of the visualizations that

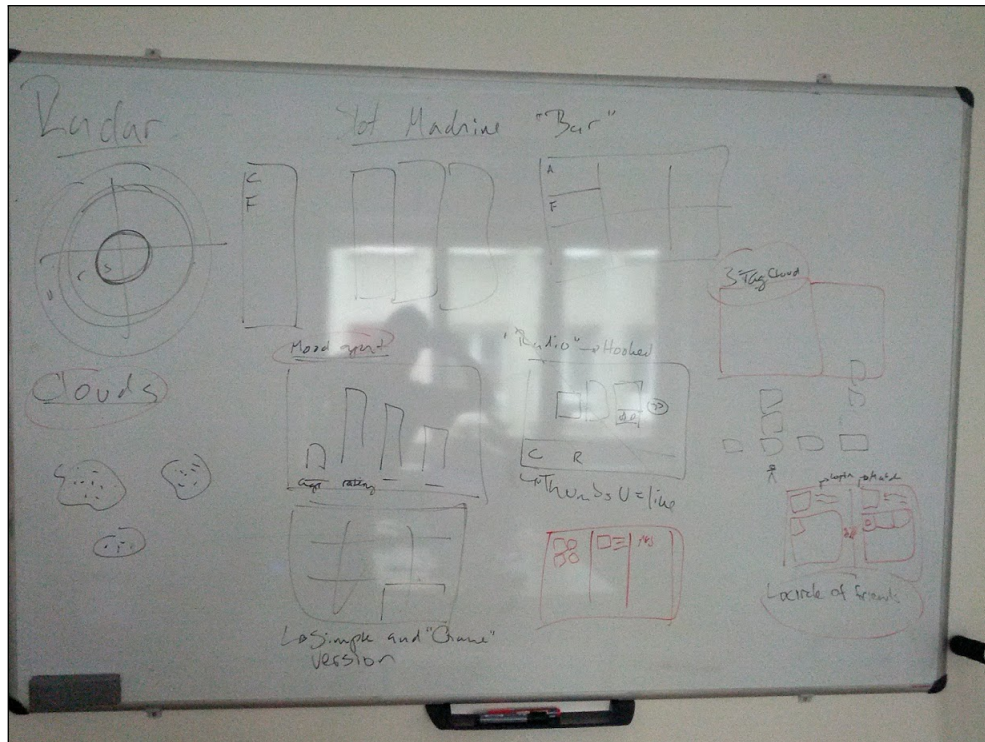


Figure 4: Results of the group brainstorming

was prototyped as a wireframe and shown to users (details in the subsection App Matcher of the section Wireframe Prototypes).

3.2 Wireframe Prototypes

3.2.1 Mood Agent

The main goal behind this visualization is to provide a simple interface that is fun and engaging. The elements that look like a bar chart in Figure 5 can be moved up and down to set their value, much like the standard Android SeekBar⁵. These bars look and behave similar to equalizers, which is why they were called sliders. These sliders represent the different dimensions of an app. The user sets these sliders to the desired level, depending on the kinds of apps he/she would like to find. The apps that correspond to this profile are then displayed below the sliders in a grid. This grid layout has the form of a TreeMap⁶, which indicates how popular a specific app is. The number of public likes in AppAware⁷ are shown if they are available.

⁵ <http://developer.android.com/design/building-blocks/seek-bars.html>

⁶ SHNEIDERMAN, B. Treemaps for space-constrained visualization of hierarchies. <http://www.cs.umd.edu/hcil/treemap-history/>, 1998. Accessed: 2012-11-07.

⁷ <http://appaware.com/>

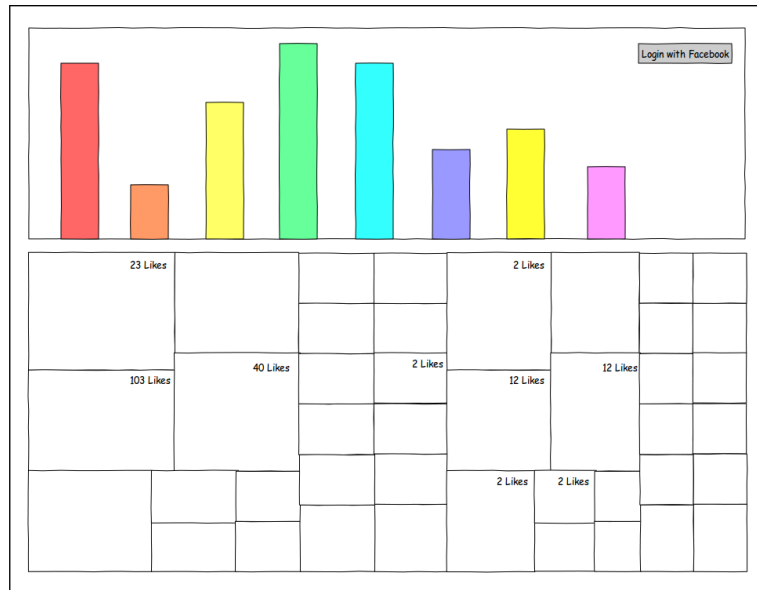


Figure 5: Mood Agent wireframe start screen and screen showed when not logged in

The user can also login with Facebook, which adds a social dimension to the discovery experience. As can be seen in Figure 6, when the user is logged in an additional slider (in gray) is displayed representing the social dimension. By changing the value of this slider the user determines how much of the profile for app recommendation should be influenced by the apps their friends have installed and liked. Accordingly, the likes shown on the apps correspond to the likes of friends, when these are available. The social component looks at the user's friends as one coherent group. There is no way to select specific friends that influence the mix of apps presented. The rest of the user experience remains the same.

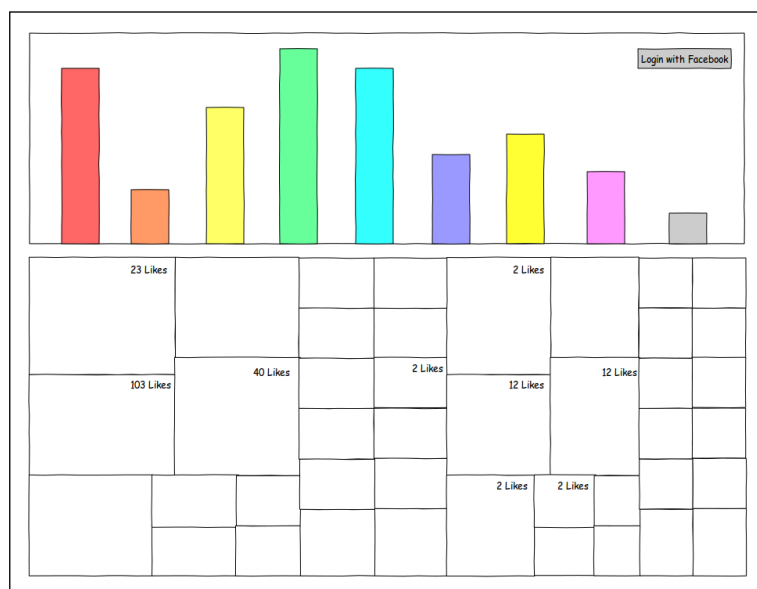


Figure 6: Mood Agent wireframe screen showed when logged in

The design attempts to support a lean-back approach to discovery. The user does not need to actively know what kinds of apps to install. The hope is that this aids users in serendipitous discovery of apps. A drawback of this visualization is that it does not allow detailed exploration, because the basis of the algorithm recommending apps supports a certain amount of randomness, with the aim to surprise the user and elicit engagement and cultivate fun.

3.2.2 Tag Cloud

The Tag Cloud takes a similar approach in displaying apps, but uses the more common notion of tags for exploration. The goal was to make a visualization that builds on a concept that is well known and widely used. The aim was to have a counterpart to the novel Mood Agent approach. The thought was that people would have a better understanding of this visualization and thus would result in greater acceptance. The social component is integrated into the visualization by adding a separate cloud of friend's profiles (Figure 7b).

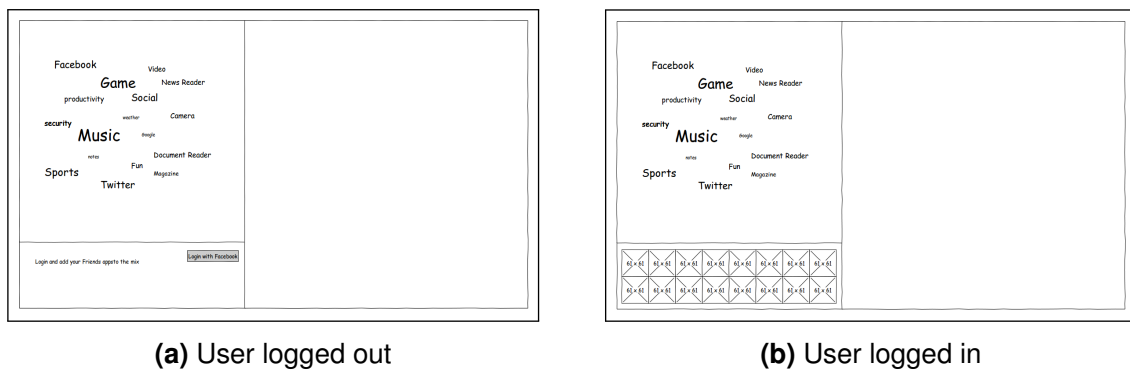


Figure 7: Tag Cloud start screen

By selecting different tags in the tag cloud as shown in Figure 8 the user creates the profile of apps that are displayed on the right hand side. Like in the Mood Agent visualization the apps are displayed in the form of a TreeMap with app likes shown. Figure 8b shows the screen when the user is logged in with Facebook. The user's friends are shown below the tag cloud but none are active, which means the social component of the algorithm showing apps is not active. In Figure 8c one can see that a friend's profile is selected, which means that only this friend's apps are included in the algorithm, in addition to the apps that are shown without the active social component. When multiple friends are selected their profiles are added to the mix. A user can also opt to just select one friend's profile without topic tags, in which case only the apps this friend has installed influence the apps that are recommended.

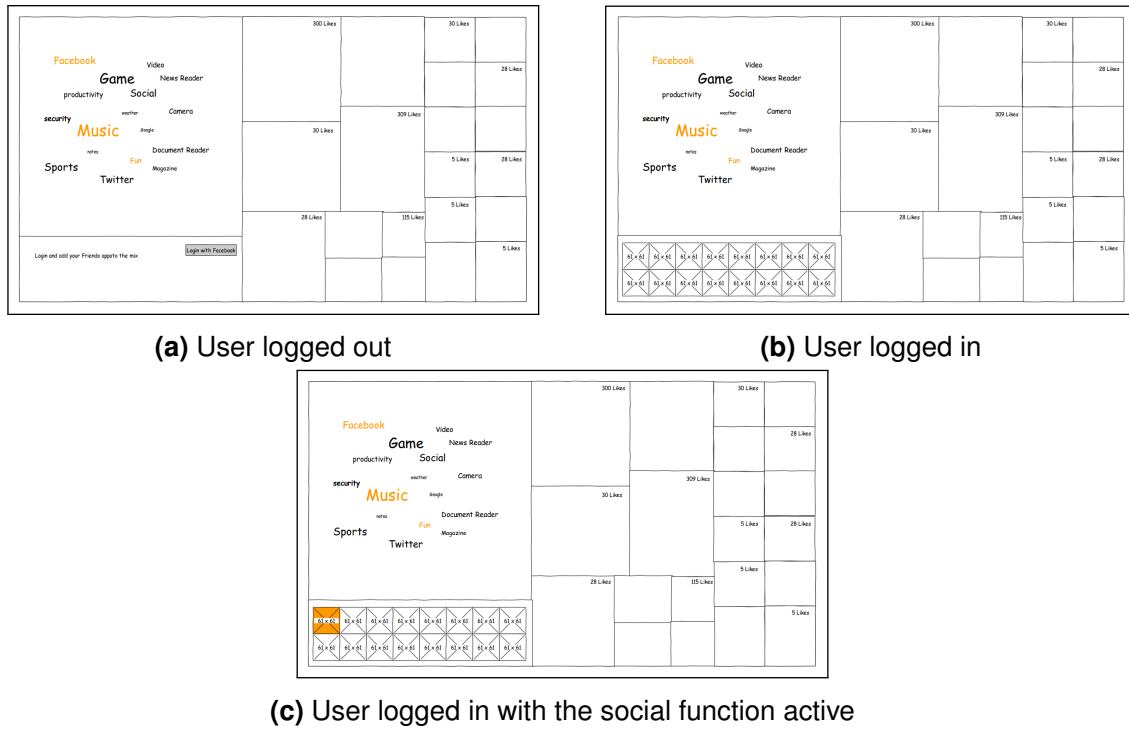


Figure 8: Tag Cloud screen when tags are active

A tag cloud visualization supports a more detailed method of exploration. The user can precisely inspect the apps that he/she would like. Consequently, the visualization is less lean-back in its form, and serendipitous discovery is harder to achieve. The motivation behind this visualization is to allow the user to explore the graph of apps in relatively high detail, without the added overload of a graph visualization as described in Appendix B sections Simple Graph, Social Graph and Advanced Social Graph. This applies not only to the app graph, but also to the social graph. If a user wishes, he/she can opt to only discover new apps based on the apps used within his/her social network. Compared to a graph visualization like the one in Appendix B in the section Advanced Social Graph, which also includes both app and social dimensions, the difference is that this visualization is simple for the user to understand. A user can only look at and go down one path at a time, which reduces the visual complexity of the information displayed on the screen. Additionally, the process of examining apps is simplified. The user selects the desired tags once and can then look through a whole collection of apps. Whereas in the graph visualization a user must walk through the actual physical path in order to have the same collection represented on screen. A tag cloud with different attributes that can be selected to create an app profile and then display all appropriate apps in one grid, so that the user has an immediate overview, is closer to a typical user's mental model than a graph. A graph is appropriate if the user is interested in the relationship between apps and the global environment of specific apps. A tag cloud obfuscates these things, but makes it simpler to explore and

discover apps that fit a certain profile.

3.2.3 App Matcher

The App Matcher visualization was one that was created during the group brainstorming session. Many of the ideas present at the time only had the social element as an afterthought and not as the primary focus. Consequently, we decided to come up with an interface that is social at its core and incorporates game-like aspects. App Matcher is the result of setting the social interaction as the focal point and designing the app discovery around it. As a result, app discovery is secondary to interacting and connecting with people. This form of visualization is social at its core, which allows for social exploration of apps.

Unlike the other visualizations, a user must be logged in to use the app. Figure 9a shows the first screen that users see. After logging in, Figure 9b represents the screen presented to users. On the left the users see their social profile (name, birthday, phone model they use and a short bio) as well as the apps they have installed and the ones they have liked. On the right is a button that when clicked, randomly matches the user to another user in the database.

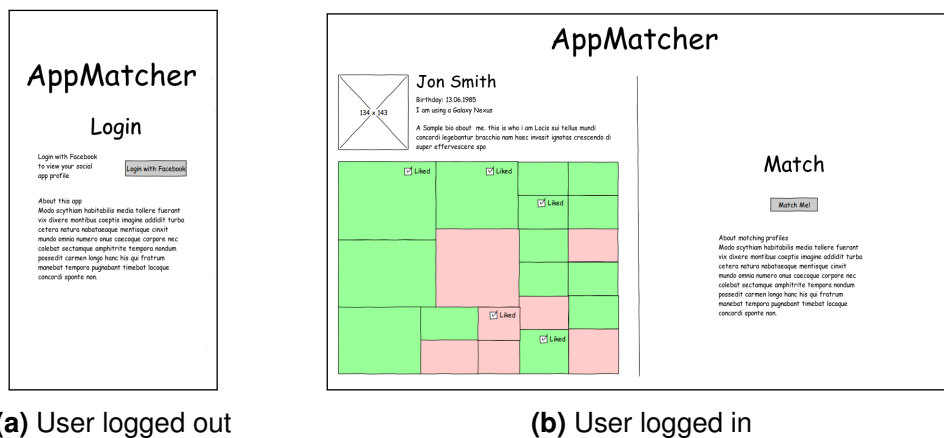


Figure 9: App Matcher start screen

Once the system has chosen another user, certain metrics are calculated which results in a representation as shown in Figure 10. The matched profile is visually the same as the users profile, with the visualized likes corresponding to the likes of the matched user. The user's apps are compared to those of the other person and a compatibility rating is calculated. This is represented by a green bar as well as a value in percent, where 100% means all of the user's apps match. In addition, the apps in both the users' lists are colored in green, indicating they both have them installed and

red being those that differ. The apps colored red in the user's list are the ones that the matched user does not have installed. On the other hand the red apps in the matched user's list are the ones that are not installed by the current user. The goal is that the user then looks at the apps that the matched user has installed and downloads those of interest. As a result, the apps people have installed serve as a indirect form of recommendation to other users.

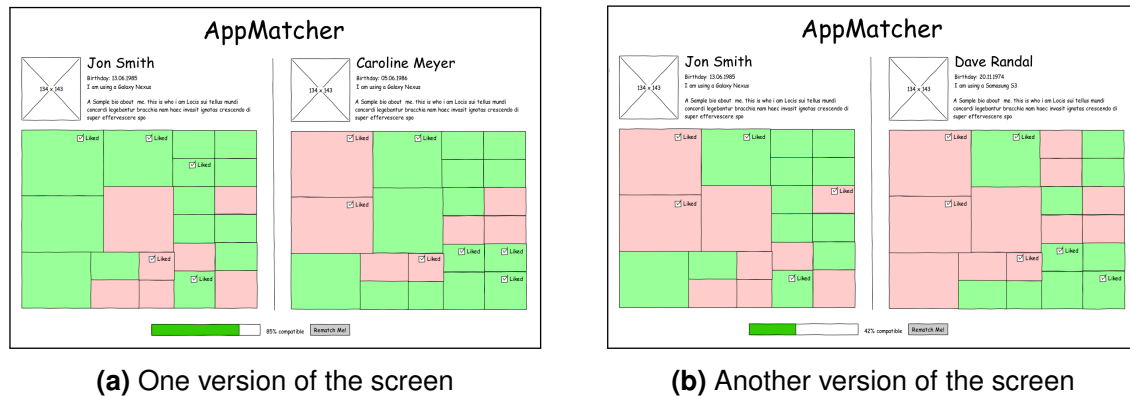


Figure 10: App Matcher comparison screen

3.3 Informal User Evaluation

After the three visualization prototypes were completely developed as wireframe sketches, they were presented to potential users in an informal fashion (around 12 potential users with smartphones).

3.3.1 Mood Agent

This visualization was the favorite for most users. Especially when they were shown the MoodAgent app⁸ for creating music playlists. The idea of the MoodAgent app was applied to apps to create this visualization, so presenting the user with that app made sense. This helped determine the user's understanding of the slider analogy.

Summary of user feedback:

- Bars could be confusing. People might not understand how they work
- Simple way to explore
- Immersive. People get lost in exploring for apps. When showing people the MoodAgent Android app for playlists they immediately started playing with it.

⁸ <https://play.google.com/store/apps/details?id=com.moodagent.android>

- Structured, but not to strict way for finding new app
- Is a coarse and high level, quick way of finding apps
- Very visual
- An approach that catches the eye
- TreeMap representation of apps in the grid conveys little information to the user.
- Why connect with Facebook? If I want to install apps my friends recommend I can do this through other (better) channels.

3.3.2 Tag Cloud

For all users this was the visualization that they grasped the fastest. Most people were familiar with tag clouds, knowledge that could easily be leveraged to navigate through this kind of visualization. The familiarity also has a downside, people can quickly lose interest and it is much harder to grab and retain their attention.

Summery of user feedback:

- Simple and easy for people to understand
- People already know how it works
- Not as much fun after a while
- More structured approach, needs more thought (active thought).
- More detailed way of finding an app
- Not so innovative. Feels boring

3.3.3 App Matcher

The App Matcher was seen as an enticing visualization, but was perceived to be too much of a gimmick. The ability of such a visualization to engage and capture the users attention over the long term is doubtful.

Summery of user feedback:

- Is a bit of a gimmick
- If I want socially recommended apps then I want these from friends and not from random people

- Works as a game, but does not really have any value for exploring apps in the long run
- Social aspect only one dimension of the “app discovering experience”
- Privacy concerns. People do not necessarily want to share their apps with everyone
- Problem with scale. Critical mass needs to be reached for the app to be really engaging and useful for users.

4 AppDJ

There are many apps out in the wild that address the big issue of app discovery on the Android platform. These apps are all on the right track to helping solve the issue of discovery, each with their own innovative approach. When we look at the current market of discovery apps, we can see that there is a wealth of apps that support “lean-forward” discovery, as mentioned in the section App Discovery. The Amazon Appstore allows users to filter and search through an abundance of categories and lists. Just like in a brick and mortar store, this is great for when you know what you’re looking for. Playboard provides the analogy of a shop window, where users are presented with a curated stream of apps, which transforms app discovery from a pull to a push action. All these apps still take a category centric approach, which requires the user to know the type of app they want to find and install. These examples show that app discovery is still a very active task. The fundamental concept behind the AppDJ visualization is the “lean-back” approach to exploration. “Lean-back” means that the user is passively consuming a medium, where the medium takes more control of the experience [21]. It provides a way to explore apps that transcend the rigid lines of categories. Additionally, the visualization is geared to support serendipitous discovery. Andre et al. [3] take a diverse look at serendipity, showing how it can be used in real world applications to induce engagement. Some of these ideas formed the cognitive framework in which the visualization was conceived.

Based on the informal feedback from potential users, the final design dubbed AppDJ was developed. The final visualization is closely related to the Mood Agent prototype, which was well received by many people. In general the same visualization was adapted, with some additional features added and some ideas refined. The concept borrows heavily from existing systems in the area of music discovery, where many tools that simplify the exploration of music have taken hold. Like with music discovery tools that take a seed song or profile, the AppDJ visualization is built on an intuition that sometimes people do not know exactly what they are looking for. By giving users the opportunity to input a minimal amount of information in the form of an “app profile” and then having the system serve up apps that roughly correspond to what the user is interested in, opens the exploration up to serendipitous discovery.

The concept that the visualization should be engaging and fun was taken out of the brainstorming session for the gamification of app discovery. The game ideas that influenced this facet can be found in Appendix C Figure 43. The prioritization of simplicity, which was an essential component of the previous visualizations, was

preserved and became the backdrop for the design of all the UI elements.

The social aspect of the visualization that previously was addressed, either as an integrated concept or augmentation, was left out in the final visualization. Originally the belief was that people would want to share the apps they find as well as their discovery experience (see Appendix B). The objective was to get people engaged through social interaction. In the offline world, the process of recommendation is a very social activity, which was expected to be able to be leveraged through allowing social connections in the app. While this social augmentation might have marginal applicability and use, many people did not acknowledge it. When asked about it, they did not appreciate the social integration without some concrete benefit. This leads to the conclusion that, specifically in the domain of digital app discovery, social integration for its own sake makes little sense. Also Falaki et al. [10] state that demographic information is not a reliable indicator for smartphone usage, so knowing someone's social circle is of little value for, and has a marginal effect on the precision of an app recommender algorithm. Numbers from the online shopping world also indicate that the influence of digital social networks is less powerful than other forces [17]. This could also apply to app discovery, as it is similar to shopping. If the social component is able to provide value to the user on its own in an integrated fashion, then it could make sense to implement such a feature. For example, social integration in a discovery app like Playboard makes sense, as it is natural to the experience and provides a tangible benefit to the user. In its current state, no way was found to meaningfully integrate the social experience, therefore the idea was dropped. In future iterations when the visualization is integrated into a fully fledged app discovery service, revisiting the idea of social integration could be beneficial.

4.1 Design

The Mood Agent design was refined so that the number of sliders nicely fit a standard tablet screen (1024x720), but also scaled down to support smaller mobile screen sizes⁹. The final number of five sliders was determined after analyzing all 30 categories of the Google Play Store and determining the traits. One slider corresponds to one dimension of categories, which in turn is made up of two traits. The details of the logic behind sliders and what they semantically represent is illustrated in the section Recommender System. Conforming to what was mentioned previously on the topic of social integration, the “Login with Facebook” and the associated social slider was removed.

⁹ http://developer.android.com/guide/practices/screens_support.html

As can be seen in Figure 11, the TreeMap visualization was removed in favor of a simple GridView¹⁰. Part of the reason was that Android has out of the box support for this feature and is backwards compatible. More importantly, the evaluation of the Mood Agent visualization which includes this TreeMap form of displaying apps, showed that many people did not innately understand the information such a representation conveys. All but the most analytical users did not understand that the size of an app was a visual cue of its popularity. The final consideration which lead to the decision of dropping this form of visual cue, was that the scaling down to smaller screens would be difficult. Because of the restricted screen real estate, such a visualization could even prove to be less usable and informative than a simple grid.

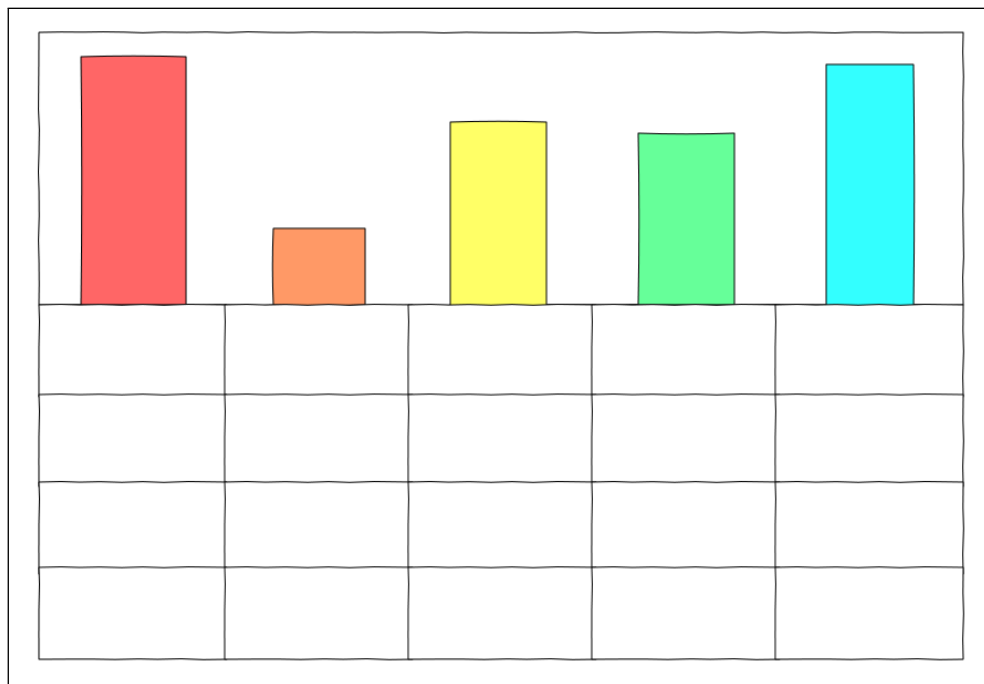


Figure 11: The AppDJ wireframe integrating ideas from the Mood Agent visualization

After the UI was sketched out as a wireframe, the home screen was prototyped using the actual Android UI elements that would be used in the final implementation (Figures 12a and 12b). The previous sketch was extended to include a screen that displays the details of an app, shown in Figure 12c. This popup appears when a user clicks on the feature graphic¹¹ of an app within the grid of apps displayed below the

¹⁰ <http://developer.android.com/guide/topics/ui/layout/gridview.html>

¹¹ A feature graphic is a promotional banner that is more detailed and larger than the app's icon. It is an optional graphic that can be added to an app and is generally more descriptive than the icon. On the Google Play website the feature graphic is shown on the top of the page to the right of the app name and icon.

sliders. In the course of developing the UI wireframe of the final visualization the importance of visuals, something that was also mentioned by potential users during the evaluation of the prototype visualizations, resurfaced. When the first iteration of the wireframe was shown to a couple people, they mentioned the importance that screenshots play when they look for apps. Out of this insight, the app details visualization in Figure 12c emerged. The screenshots for an app are shown first, while the description remains hidden and is shown only if the user wishes to read it.

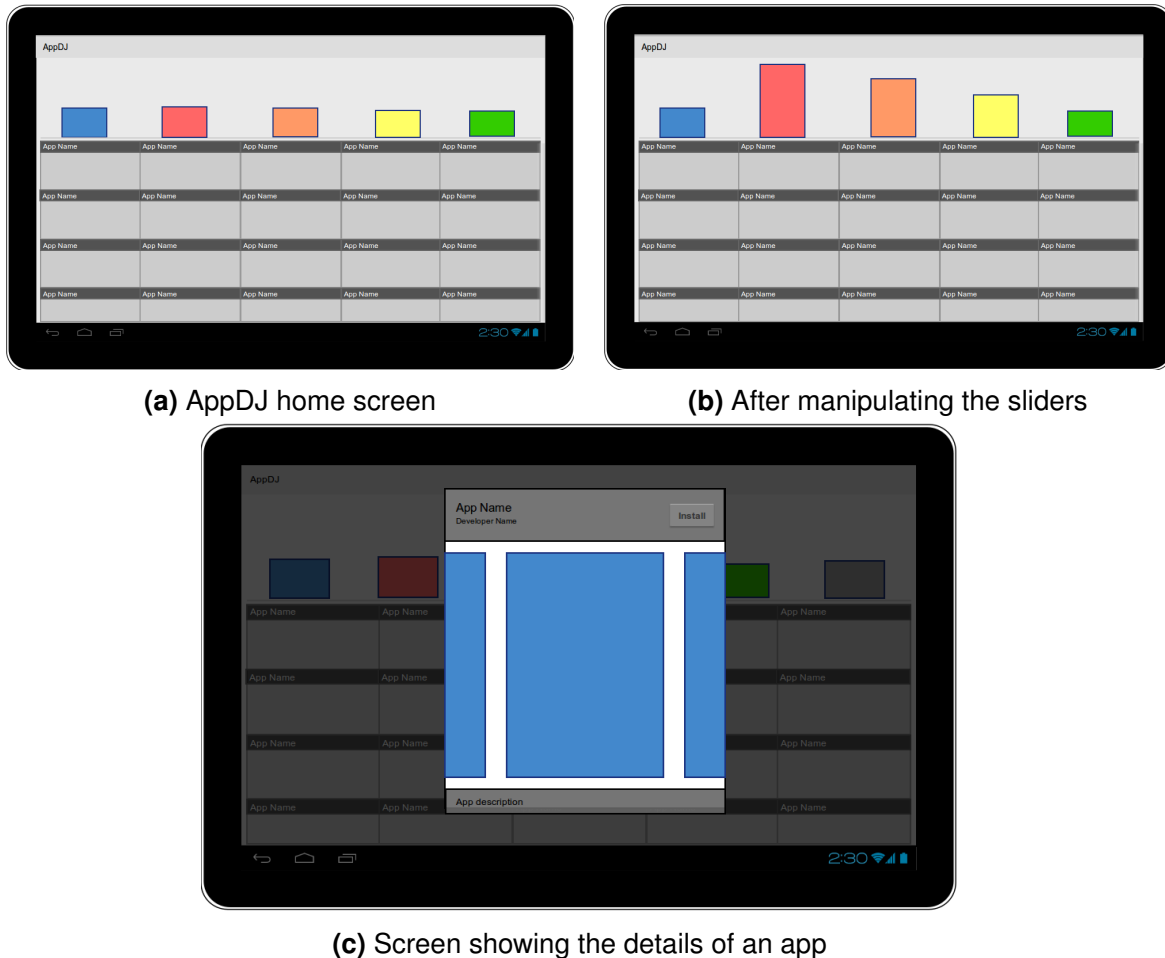


Figure 12: Android UI wireframe of the AppDJ visualization

Figure 13 and 14 are screenshots of the final version of the implemented Android app in the tablet and mobile environment respectively. Figure 13a is the screen that is shown when the user first opens up the app. As the user interacts with the sliders and the apps in the grid, the screen stays the same, only adjusting the visualization of the sliders and the apps displayed below. Figures 13b and 13c show what is displayed when a user clicks on a feature graphic in the grid of apps. The screenshots that are shown are within an Android UI container called a ViewPager¹². This allows one

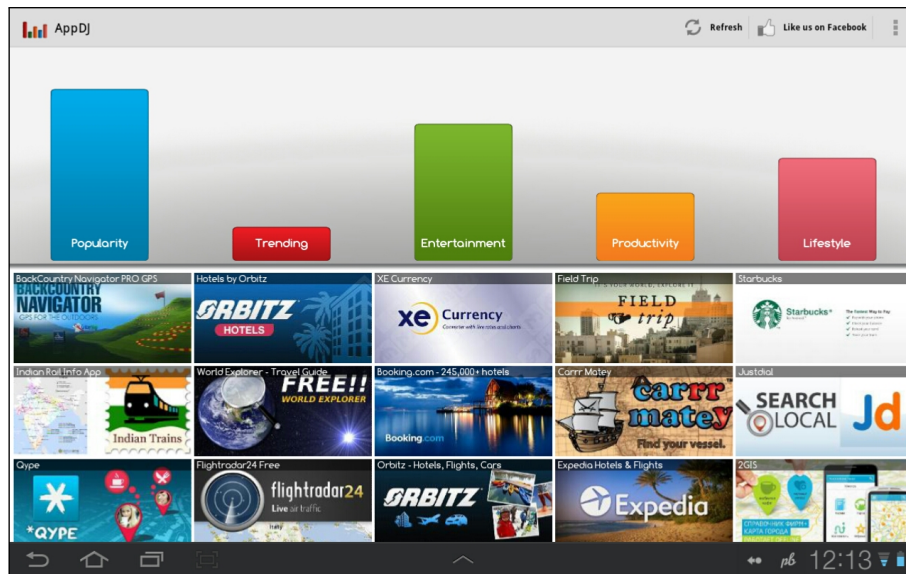
¹² <http://developer.android.com/reference/android/support/v4/view/ViewPager.html>

screenshot to be displayed with a small portion of the ones to the right and left showing. This shows the user that there are additional screenshots and provides a cue on how to interact with the UI element. With a left or right swipe the next screenshot is placed into the center of the screen, forming a carousel like visualization. The app description is hidden and shown using a `SlidingDrawer`¹³ which opens on click or when dragged upwards, similar to the Android system bar.

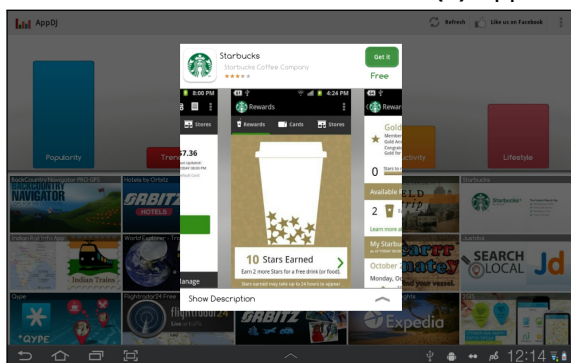
Comparing Figure 12b and 13a one can see that most of the elements from the wireframe were implemented as designed, with the addition of some minor subtle visual enhancements that were integrated after user and expert feedback. The most overt is the state that the app starts in when it is opened. Originally, all sliders were set to zero as sketched in Figure 12a. In the final implementation the sliders in the start screen are initialized with a random value between zero (inactive) and five (fully extended). Additionally, some features were added to the Action Bar. These include a refresh button, a Facebook like button as well as a send feedback and help menu. The reasons for adding these enhancements are discussed in the section Evaluation. For the screen displaying the details of a selected app as illustrated by Figure 13b, there were some changes in the header showing the general information of an app. The price and rating as well as the icon were added, because the UI dimensions allowed the display of more information and these were elements that users cited to be important.

These two sets of screenshots show that the endeavor of keeping the UI as uniform as possible across different screen sizes was successful. The only small and subtle differences between the tablet and mobile version are that the sliders are a bit smaller and more compact and the grid of apps is not as wide. All the essential elements to the experience could be preserved, including the analogy of sliders as well as the manner in which app details are shown. This demonstrates that the visualization is equally adequate for a wide array of screen sizes, making it a generic visualization.

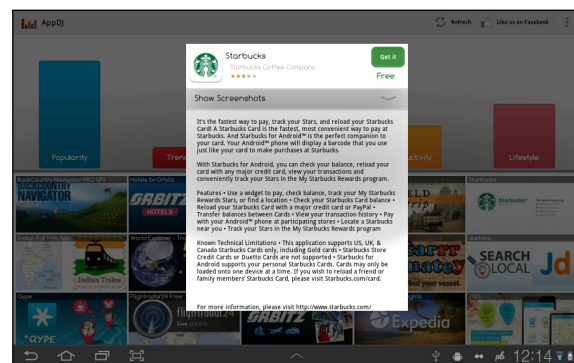
¹³ <http://developer.android.com/reference/android/widget/SlidingDrawer.html>



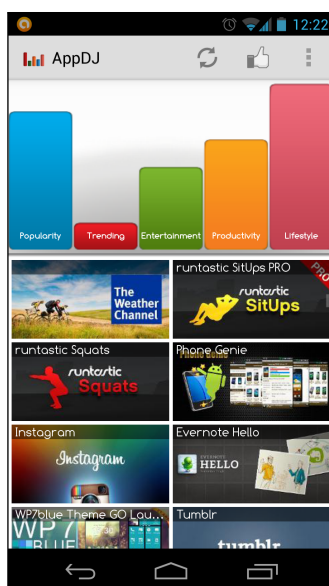
(a) AppDJ home screen



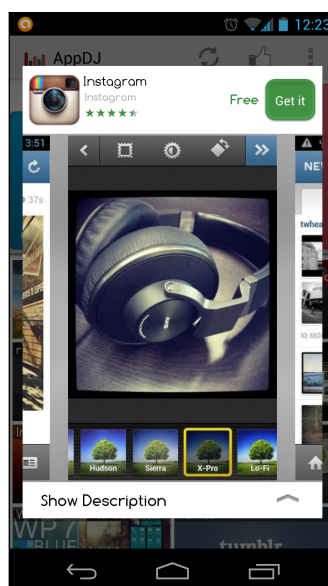
(b) App details screen



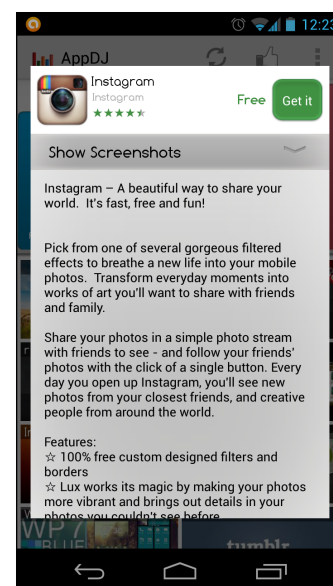
(c) Description of an app

Figure 13: Final UI of the AppDJ Android app running on a tablet

(a) AppDJ home screen



(b) App details screen



(c) Description of an app

Figure 14: Final UI of the AppDJ Android app running on a mobile phone

4.2 Technical Details

The final visualization is implemented as an app on the Android mobile platform. The app is available for download through the Google Play Store¹⁴. In addition to the Android SDK, other libraries were used to support specific features. The app was written and designed primarily for tablets with the target build being the current Android version at the time (version 4.1 Jelly Bean). The app is compatible with mobile devices of all screen sizes down to version 2.2 (Froyo) with the help of the Android Support Library¹⁵. To simplify and streamline HTTP requests the Android Asynchronous Http Client¹⁶ was used. For certain specific data constructs the Guava¹⁷ library was utilized to streamline performance and coding effort. The category mappings for the sliders were statically coded in CSV to allow them to be easily changed and updated. To simplify the reading of these files the OpenCSV¹⁸ library was used. Finally, the analytics and tracking of user events in the app was implemented with the help of Google Analytics¹⁹.

For the GUI implementation the standard Android containers and widgets were used for the most part. A notable exception is ActionBarSherlock²⁰ which was used to support the ActionBar²¹ widget in Android versions below 3.0 (Honeycomb). For this implementation of AppDJ two custom widgets were implemented to provide unique functionality. The SliderBar widget was fully designed and implemented from the ground up, whereas the StarBar was adapted from code received from one of the AppAware developers Bo Wang. One additional custom widget is a simple wrapper around a standard TextView²² to allow custom typeface definition in the Android configuration files.

SliderBar Widget

To provide the slider functionality, a SliderBar widget was created. This widget represents one simple slider which can have a variable amount of discrete levels. Additionally, it can be visualized by any 9-patch drawable²³ and can have a human readable label, whose orientation and position can be defined when the widget is

¹⁴ <https://play.google.com/store/apps/details?id=appdj.android>

¹⁵ <http://developer.android.com/tools/extras/support-library.html>

¹⁶ <http://loopj.com/android-async-http/>

¹⁷ <https://code.google.com/p/guava-libraries/>

¹⁸ <http://opencsv.sourceforge.net/>

¹⁹ <http://www.google.com/analytics/>

²⁰ <http://actionbarsherlock.com/>

²¹ <http://developer.android.com/guide/topics/ui/actionbar.html>

²² <http://developer.android.com/reference/android/widget/TextView.html>

²³ <http://developer.android.com/guide/topics/graphics/2d-graphics.html#nine-patch>

drawn. Each slider knows its current position and has a set minimum and maximum value. Each slider controls its own position independently as well as its state. The value of the slider can be changed either by clicking in the clickable area of the slider or dragging it, which has an affect on how the slider is redrawn when the screen refresh function is called. When the state of a slider changes, this triggers a call in a custom listener.

The global functionality of all sliders are achieved by this custom listener. All sliders in the layout share the same listener, which coordinates all actions. When the listener deems it right to relay a certain action, the aggregated result of all actions in the individual sliders are passed on. In the specific implementation of the listener in AppDJ, a slider in the “dragging” state blocks all requests to the backend. A request action is sent only once the slider has been released. Furthermore, there is a delay of 2 seconds between the last recorded action and the sending of a request. This is so that a user can change multiple sliders at once without having to wait for a response. This also cuts down the amount of requests and the backend load.

StarBar Widget

The StarBar widget is used to display the rating on the screen showing the details of an app. The widget draws ASCII stars based on the inputted rating. The only change to the code made, was that the color could be defined in an Android configuration file.

Backend

The recommender system supplying the data is powered by the AppAware API²⁴. The recommender system is called when the listener for all the SliderBars determines that a request for new apps is necessary. Based on the settings of the sliders, the recommender system determines the mix of apps to display. The backend then queries the AppAware API to received the proper app information. A more detailed description is provided in the following section. The whole system is supported by an LRU memory cache that locally stores the information of apps that have already been viewed by the user.

4.3 Recommender System

AppDJ includes a recommender service that determines which apps to show given a set of slider values. As described in the section Design, each slider represents a

²⁴ <http://dev.appaware.com/>

dimension. This means that each app needs to be described in terms of these dimensions. An initial attempt was to cluster apps with the help of LDA (Latent Dirichlet Allocation)²⁵. The resulting document topics could then be associated to the five slider dimensions, indirectly “rating” the apps in terms of these dimensions. The problem with this approach was that associating the most basic model of 300 topics to 5 sliders would constitute a great deal of work, either manually or programmatically. This effort would go way beyond the scope of this thesis, so it was decided that the prototype would be based on a static mapping of not apps to sliders, but categories to sliders. The basic model is the same, just the level of detail is lower which means specific apps cannot be compared to one another.

4.3.1 App Traits

To work around this problem six traits for apps were elicited and the 30 categories were ranked along these measures. The traits were determined by analyzing a sample of apps in each category and brainstorming “meta categories” that could be associated to all the categories to some degree. To begin, categories were clustered into groups to see what similarities and differences existed. Out of this the final traits were obtained. Additionally, the aspects people find important when looking for new apps were considered. Appendix D Figure 44 shows how the categories were ranked for each trait.

Work

An important criteria when searching are the task oriented features of the app. Categories that have apps in them that are geared towards the work environment such as “Finance” or “Business”, are categories that this trait strongly applies to.

Entertainment

Analogous to this, entertainment is the counterweight. In many cases, the fun factor of an app plays an important role. Entertainment apps are not necessarily only games, but also “Music&Video”, and to a smaller degree “News&Magazine” apps.

Games

For many people who have smartphones, games are a key feature. Facilitating the discovery of games can aid users who are trying to find a new game that fits their

²⁵ For more on LDA read *Latent Dirichlet Allocation* by Blei, Ng and Jordan, 2003: <http://www.cs.princeton.edu/blei/papers/BleiNgJordan2003.pdf> Accessed: 2012-11-27

needs. The game trait is fairly restrictive, applying only to games. The reason that games have their own trait, is that a large portion of the apps on the market are games and there are six categories that are game related.

Productivity

This trait is closely related to the work trait. Most apps that are more work oriented are also productivity apps. Unlike work apps, productivity apps can also be classified as entertainment apps, etc. For example a “Health&Fitness” app can be viewed as both a productivity and to some extent an entertainment app.

Utilities

Tools and personalization apps are ones that fulfill some task where the ultimate goal is not contained within the app itself. It is a medium through which the goal is achieved. These types of apps are classified as utility apps.

Lifestyle

On the other end of the spectrum are lifestyle apps, where the app is an end in itself. These are primarily apps that are to some extent an extension of the users personality and make a fashion statement. “Social” apps, as well as to some extent “Health&Fitness” apps can be classified as lifestyle apps.

4.3.2 Variable Dimensions

As can be discerned from the descriptions of the traits above, many seem to be two sides of the same coin. This relationship between traits is by design, so that they can be combined to form a dimension which can then be associated with a slider. The mapping of these dimensions can be seen in Appendix D Figure 45. In total there are three dimensions that are based on static category mappings. The other two dimensions represent the dynamic component, where time changes the way in which the dimension is applied to the recommendation. This means that the apps attributed to each level of the dimensions can change over time, giving the visualization the character of evolving data. These variable dimensions are popularity and trending, which both have lists that are calculated based on user activity on the AppAware backend, which is accessed via the AppAware API.

Popularity

This dimension does not include any app traits, but instead is an indicator for the degree to which the popularity of an app should influence the recommendation. The higher

the setting of the slider the more rigorous the requirements for popularity are. The AppAware API has different top lists for the different degrees of popularity which are used by this dimension. The lowest popularity setting bases queries off of the hourly popularity list, while the highest takes the results out of the monthly popularity list. The rational behind this is that apps on the monthly popularity list have been in the focus of consumer interest for a longer period of time and are thus more popular.

Trending

Much like popularity, the trending dimension is also not based off of category centric traits. Trending is an indication of how fast an app is being adopted. Generally, people looking for trending apps are interested in ones that are on the cusp of becoming popular. As such, the trending dimension is the irregular and more “up to the minute” pendant to popularity. Like popularity, this dimension is also based on a top list in the AppAware API. These lists are generated based on time, with the value of the slider determining which time frame is used. In contrast to the popularity dimension, the more a user selects trending as a characteristic the shorter the time frame of the queried lists. This means that the hourly trending list is used when trending is at its maximum setting, whereas monthly trending when it is at its lowest. This is based off the notion that an app’s degree of trendiness is closely linked to time, which means more recent trending apps have a higher degree of trendiness. Following this logic, apps that are trending for a longer period of time eventually become popular, and from then on appear in the popularity dimension.

4.3.3 Category Dimensions

In contrast to the variable dimensions popularity and trending, category dimensions are static. A category dimension is made up of two traits on either side of the spectrum. These two traits are then the minimum and maximum value of each dimension, which translates into a slider. The name of the slider and the dimension is given by adopting the name of the trait that represents the maximum value.

Entertainment

The category dimension that combines the traits entertainment and work is referred to as the entertainment dimension. All categories are mapped depending on how close they are to the trait work or entertainment. The nature of the apps in each category determines where they stand in this dimension. Apps geared towards entertainment, such as apps in the “Music&Video” category are higher along this dimension, where as apps in the “Finance” category are lower. In Appendix D, Figure 45a shows the mapping of all 30 categories along this dimension.

Productivity

The productivity dimension brings together the productivity trait as the maximum value with the game trait as the minimum value. Categories are mapped to either side, depending on the nature of the apps contained in the category. The more productivity oriented a category is, the higher it is along this dimension. Games are at the bottom of the productivity spectrum, because they represent the opposite side. “Music&Video” apps could be placed somewhere in the middle, because they incorporate certain game aspects, while still supporting a user in the completion of some specific task. The detailed mappings of this dimension can be seen in Appendix D Figure 45b.

Lifestyle

Combining lifestyle and utilities, two traits focusing on the usage of the app, constitutes the lifestyle dimension. Apps in categories such as “Tools” or “Communication” are classified as utilities, whereas “Social” or “Health&Fitness” apps are attributed the lifestyle trait. Utilities are seen as apps through which a certain goal is achieved with the help of the app. Contrarily to lifestyle apps, the goal is the usage of the app itself, making the app somewhat like a statement of personality or character. The mapping of this dimension can be seen in Appendix D Figure 45c.

4.4 Implementation

4.4.1 Dimension to Slider Mapping

To translate the dimension into values that can be controlled by the sliders, which have six discrete levels whereby one indicates that the slider is not active, the dimensions have to be split up. One dimension has ten steps which need to be mapped to five slider levels. Each of the three category dimensions has a custom dimension to slider mapping, statically defined in the recommender system. This was done by looking at the dimension mapping as seen in Appendix D Figure 45 and determining a good mix, such that there are never too many categories in one level.

The three dimensions are mapped to their three sliders as follows:

Entertainment		Productivity		Lifestyle	
Level	Dimension	Level	Dimension	Level	Dimension
1	1, 2	1	1	1	1
2	3, 4	2	2, 3, 4	2	2, 3, 4
3	5, 6, 7	3	5, 6	3	5, 6
4	8, 9	4	7, 8	4	7, 8
5	10	5	9, 10	5	9, 10

Table 1: Slider levels are attributed to the their corresponding category dimension

4.4.2 Recommender Algorithm

To get the list of apps to display to the user, the levels of each slider need to be transformed into AppAware API queries. The API allows queries to two types or lists, trending and popularity. This makes the translation of the popularity and trending sliders fairly easy, requiring only a weighting factor to get the proper amalgamation of apps. The task becomes more difficult with category dimensions, as these do not have a direct counterpart in the API. The API can limit the query to a specific category, which is leveraged in the hack that transposes the dimensions into a list of apps that reflects the slider profile set by the user.

4.4.3 Variable Dimension API Interface

The two sliders popularity and trending represent the variable element in the query. They map directly to the type of event in the AppAware API²⁶ and are the basis for the queries of the category dimensions. This denotes that the proportion of popularity and trending form the baseline, which the category dimensions refine. When no slider representing a category dimension is active, the list shown reflects only the values of the popularity and trending dimensions. In terms of the API request, this connotes that the query applies to all categories, i.e. the value “All Applications” for the category parameter is selected.

The mapping to a query is fairly trivial for the variable dimension. The value of the slider is translated into the “time frame” parameter of the query as described in the section Variable Dimensions. Since this is the basis for the category dimension queries, the variable dimension is determined after these. Based on the ratio of trending to popularity, each category dimension query is partitioned into two queries where the number of apps selected reflects the ratio of the variable dimension. Consequently, if all sliders in the category dimension are inactive, the only influence on the recommendation is the ratio of the two variable dimensions.

4.4.4 Category Dimensions API Interface

To query the API for the proper apps based on the slider profile, the recommender algorithm uses the mappings in Table 1 together with a specialized algorithm shown in Algorithms 1, 2 and 3. How the recommender service is called and how it interacts in

²⁶ The parameter 't' described in the documentation at <http://dev.appaware.com/1/doc/api/app/top.php> represents the type of list to query. In this case the differentiators used are 'trending' and 'popularity'

the full context of the app is described in the section Recommender System Process. The slider profile needs to be translated into its underlying category composition, because the AppAware API only supports category centric queries. While the category dimensions could have a certain amount of variability in some cases, they are mostly deterministic.

Algorithm 1 Finds the total weight of all categories, giving category intersections more significance

Require: $intersectionFactor = 2 \vee categoryIntersections = 3$

Ensure: $totalCategoryWeight \geq 1 \vee numberOfCategories \geq 0$

```

for all Sorted categories do
  if Number of category intersections is 1 then
    Increment  $totalCategoryWeight$  by one
    Exit loop
  end if
  if New category then
    if Number of categories processed <  $categoryIntersections$  then
       $weight \leftarrow$  number of category intersections *  $intersectionFactor$ 
      Append  $weight$  to  $totalCategoryWeight$ 
      Increment the number of categories processed in  $numberOfCategories$ 
    else
      Append number of category intersections to  $totalCategoryWeight$ 
      Increment the number of categories processed in  $numberOfCategories$ 
    end if
  end if
end for

```

In a first step the recommender system needs to find the category intersections between the different sliders. All categories are mapped along the dimension that each slider represents, so there could be an intersection containing three of the same categories. Algorithm 1 shows how the system goes about finding possible intersections and assigning category weight. When the categories selected by sliders intersect, then these categories are given more weight in comparison to categories that only appear once. The factor that is applied is defined as a static variable represented by the variable *intersectionFactor* in Algorithm 1. The total weight of all categories changes depending on the configuration of categories in the selection. The amount of intersections is capped at three, so that the selection of different categories and subsequently apps is not too diluted. The global variable *categoryIntersections* represents this value in Algorithm 1.

After finding the intersections of categories and the total weight of the categories based on the selection, the system needs to compute the number of apps to query per

Algorithm 2 Set the number of apps to query for per category

Require: $intersectionFactor = 2 \vee totalApps = 100 \vee totalCategoryWeight \geq 1$ **Ensure:** $roundedNumberOfApps \geq 1$

```

for all Sorted categories do
  if Number of category intersections is 1 then
    Exit loop
  end if
  if New category then
     $numApps \leftarrow totalApps / totalCategoryWeight * intersectionFactor$ 
    Add rounded  $numApps$  to  $roundedNumberOfApps$ 
    Request  $numApps$  from API
    Remove category from categories not queried
  end if
end for

```

category. The reason for this is, as mentioned above, that the AppAware API's queries are category centric. The weight of each category in relation to the total category weight determines how many apps are queried in that specific category. Algorithm 2 shows how the weight of a category is translated into the number of apps needed, out of a defined number of apps. The size of the resulting list is a constant expressed by $totalApps$ in Algorithm 2. After finding the number of apps for a category being processed, a request is built and sent to the AppAware API. The request is asynchronous, so the response is handled separately once it is received, meaning that the next category can be processed immediately. A more detailed description of this process is given in section Recommender System Process.

After determining the categories that intersect and the associated number of apps, there could be situations in which the full number of apps allowed in a list has not been exhausted. In this case, the non intersecting categories in the selection based on the current slider setting are used to fill up the remaining space. These categories are chosen at random, which gives the system a measure of indeterminism. Additionally, since these categories are not as relevant as the ones that intersect, there is a multiplier that reduces the impact of the category on the total selection. Algorithm 3 demonstrates how this is done using the constant $randomMultiplier$ to determine how many additional categories to query based on how many intersections have been found. Like in Algorithm 2, the API is queried as soon as the number of apps for the given category has been determined. Since all these numbers, both in Algorithm 2 and 3 are rounded, there could potentially be rounding errors. To fix this problem, the total number of apps queried is factored into the last query and adapted, such that the number of apps in a list is always equal to $totalApps$ as exhibited in Algorithm 3.

Algorithm 3 Fills up the list with apps from other categories in the slider selection

Require: $categoryIntersections = 3 \vee randomMultiplier = 2 \vee totalApps = 100 \vee totalCategoryWeight \geq 1 \vee numberOfCategories \geq 0$

Ensure: $roundedNumberOfApps = totalApps$

```

if  $numberOfCategories < categoryIntersections$  then
     $additionalCategories \leftarrow (categoryIntersections - numberOfCategories) * randomMultiplier$ 
end if
Cap  $additionalCategories$  at total number of categories not queried
for all  $additionalCategories$  do
     $numAdditionalApps \leftarrow totalApps / totalCategoryWeight / additionalCategories$ 
    Add rounded  $numAdditionalApps$  to  $roundedNumberOfApps$ 
    if  $totalApps < roundedNumberOfApps$  then
        Request  $numAdditionalApps + (totalApps - rounded numAdditionalApps)$ 
    from API
    else
        Request  $numAdditionalApps$  from API
    end if
    Remove category from categories not queried
end for
if  $totalApps > roundedNumberOfApps$  then
    Request  $totalApps - roundedNumberOfApps$  from API
end if

```

4.4.5 Recommender System Process

The recommender system is integrated with the UI and is called when a change to the sliders is signaled by their listener. The system then reads the values of the sliders and calculates the queries needed to fill a list of apps that corresponds to this profile. Figure 15 shows how this interaction occurs.

Once the user has changed the sliders, the listener associated with these triggers the process, which gets the recommender service that is responsible for calculating the needed queries and handling the interaction with the AppAware API. Once the service has received all requests it triggers a callback to the caller, who then can process the list of apps. This detaches the UI process that calls the service, so that it can still respond to user input. Once the list of recommended apps has been generated, the UI process (the caller) is notified and can display the list in the proper form.

The recommender system queries the API in an asynchronous manner as detailed in Algorithms 1, 2 and 3 in section Category Dimensions API Interface. The recommender system keeps track of the queries that have been received and as

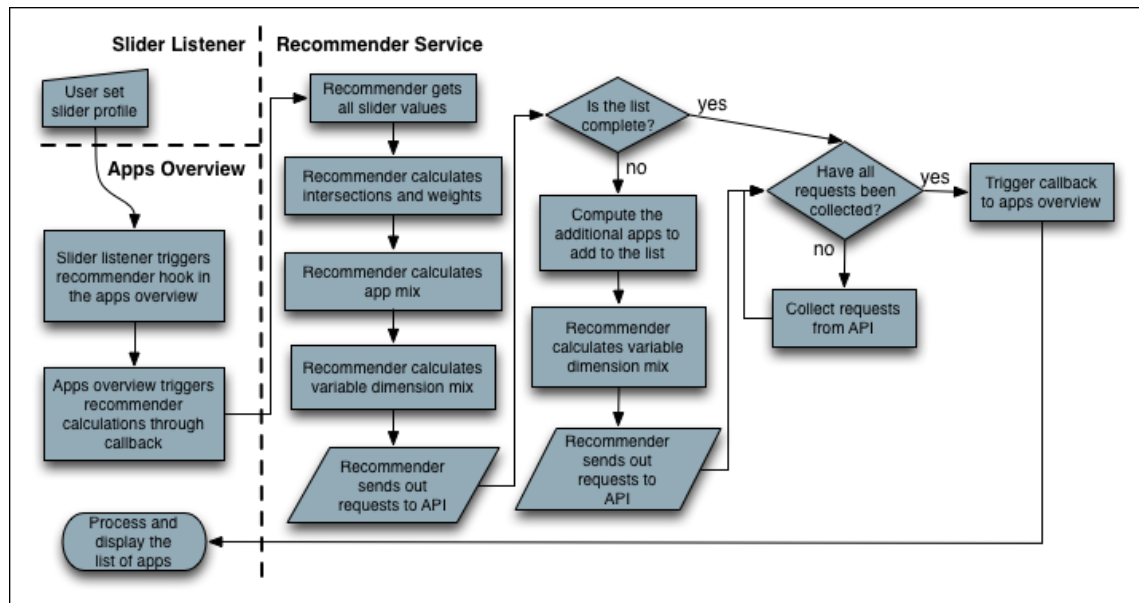


Figure 15: The process description of transforming the slider profile into requests that fill the list of apps displayed to the user

shown in Figure 15, it waits for all of these to be finished. Once the full list has been created the recommend service calls the callback function that was registered when the service was last called.

4.5 Issues and Problems

During the implementation of the visualization, some issues pertaining to the details of the visualization arose. These had to do with the specificities of the Android system, and were resolved either with the help of code examples or changes to the architecture of the app.

Slider Widgets

A separate custom widget had to be implemented for the slider visualization. Since this was a widget that was built from the ground up, functionality issues appeared. In the beginning, the sliding up and down between the steps of the sliders was error prone. Many of these issues arose only when showing the visualization to users, as these usually interacted with the sliders in a different way than the standard testing process. Installing a debug version of the app on the phone of a tester helped to find and solve bugs that arose out of the natural usage of the app.

Image Loading

Some apps do not have feature graphics, in which case the default loading image is displayed. Since this loading image is generic, the visualization loses some of its expressiveness. The image loading process was updated with a function that takes a callable as a parameter, so that each caller can define what the default process should be when image loading fails. This allows the image loader for the app feature graphics in the GridView of apps to load the icon image if no feature graphic is found. In this case, the default loading image will only be shown if there is a network error.

Recommender System

The asynchronous handling of the recommender service in a separate process posed some issues, especially pertaining to the Android activity lifecycle²⁷.

When the user changes the orientation of the device the activity is restarted. When this happens all references to threads spawned from the activity before are lost. Determining the queries and querying the API is a relatively expensive operation, so recalculating things that have been previously calculated is not an option. The responsiveness of the app is paramount, so the state of the process should be saved. For it to be possible to continue the calculation normally, what happens upstream needs to be transparent. To solve this dilemma, a mechanism was created that unhooks the reference to the old caller when it is destroyed and rehooks the new instance of the caller when it reappears, with the rest of the process continuing normally. In the case that the recommender system has completed the process of finding apps, the system calls the new caller's callback as soon as it is registered. The diagram in Figure 15 shows the context of this process in a more detailed fashion.

²⁷ <http://developer.android.com/training/basics/activity-lifecycle/index.html>

5 Evaluation

The evaluation of the AppDJ app was separated into two parts. Before the initial release in the Google Play Store, an expert evaluation was done by the members of the ZPAC group²⁸ at the University of Zurich. Additionally, some more informal user feedback was collected by demonstrating the working app to users. The final deployment of the app contained code to track user inputs via the Google Analytics Framework, which allows evaluation of the app in the field. The resulting data lead to minor adjustments of some of the app's components. After the app had been published on the market for a month, an update was released which included minor changes to the UI, as well as a link to a user survey. The goal of the survey was to elicit organic user feedback and determine if the goals set for the visualization concur with user perception.

5.1 Prerelease Evaluation

Before releasing the AppDJ app to the general public, a prerelease evaluation was conducted. This was done in two stages, the first one being a expert evaluation and the second consisting of obtaining user feedback in an informal manner. User feedback was continually gathered on major iterations, whereby the last iteration after the expert evaluation was not evaluated by users before the release.

5.1.1 Cognitive Walkthrough

In order to determine if the design goals had been reached, a cognitive walkthrough was conducted with the members of the ZPAC group. The goal was to find any design related problems, as well as get general feedback pertaining to the visualization. This additionally allowed certain conceptions as well as processes, that might have been taken for granted, to be evaluated and judged by a third party. The detailed answers given by the participants of the walkthrough are in Appendix E. The input from this session, which took about 45 minutes, was the basis on which many of the changes described in the section Changes and Additions were made.

As with the informal feedback from users, the concept was understood for the most part, however the sliders did present a considerable hurdle to correctly interpreting the visualization (Stephanie: "It is not completely obvious that the bars are interactive, it will likely be an unfamiliar type of interaction", Gunlen: "bars don't have visual

²⁸ www.zpac.ch

indication (affordance) of being interactive sliders”). The fact that the sliders fail to make use of an existing mental model is also the main drawback of this visualization. Nevertheless, it was found that once the concept was understood, the usability was immediately apparent.

5.1.2 Informal User Feedback

Through observation, discussion and comments, the remaining problems and difficulties users had were identified and subsequently fixed. This was done by presenting a couple potential users with a working version of the app and having them play with it (10-12 people). People’s reaction to, and comprehension of the concept of sliders was closely observed. This was seen as one of the critical issues with the visualization, as many people had previously cited having difficulties with understanding how the sliders work in the Mood Agent prototype.

5.1.3 Changes and Additions

The issues arising from this evaluation, were addressed in the released version. The changes and additions that were made to the app following this evaluation are listed below.

- **Added background to sliders:** Following a comment from a user in an early build of the app, a background gradient was added to make the sliders “pop out” and become more distinct from the rest of the visualization.
- **Randomize slider start position:** One user had difficulties noticing that the sliders could be dragged or clicked to change the profile. As a result, an implementation with random initial values was created to determine if this would help users better understand the slider concept. The subsequent evaluations by users showed that while not eradicating the problem, it substantially improved people’s understanding of the slider concept.
- **Back button cancels API request:** A user mentioned that it would be nice if a request could be canceled, for example if the connection was really slow. This was implemented by allowing the loading dialog to be dismissed and the request canceled with a press on the system’s back button.
- **Added refresh functionality:** In the real world environment, network connectivity is a major issue. While showing the visualization to users, it became clear that there is a need for a refresh button, so that the visualization can easily be reloaded after a network timeout.

- **Changed spacing of apps in the overview list:** The cognitive walkthrough showed that it was not clear how the sliders influenced the overview. It was mentioned that the sliders could be viewed as only influencing the column of apps below them (Alice: “Do the column items belong together?”). The fact that the sliders lined up with the columns of the overview was an unintentional artifact of the visualization when being viewed on a tablet in landscape mode. This problem was alleviated in part by changing the spacing between the apps to make it more uniform, as well as distinct from the slider elements.
- **Added scrollbar to list of apps:** An additional problem that was found with the app overview, was that it was not immediately clear that it was a scrollable list (Bob in response to if a user would be able to notice if the list is scrollable: “Maybe, if the screen always looks like this. But a scrollbar would make it more obvious”). To make it obvious to the user that it is a list which is scrollable, a scrollbar that fades after five seconds was added to the right side of the list of apps.
- **Visualization divider:** During the cognitive walkthrough many participants had issues with separating the sliders from the overview of apps. It was noted that the user could be confused by the fact that the slider lined up with the columns of apps below them. Additionally, it was mentioned that a user might not notice that the app list is scrollable. Alice later suggested the this could be resolved by adding a “visualization divider” (a shadow underneath the slider background, separating it from the app overview).

5.2 Deployment and Field Evaluation

The final version of the app included tracking of user actions over Google Analytics. These analytics not only provide information on user behaviors, but also lends itself to determining how the system should react and at what speed. The review of these analytics uncovered some areas where minor fine tuning was required. Additionally, a version of the app was released with a link to a questionnaire, with the aim of getting user feedback and determining if the goals set for the visualization concur with the user perception of the app.

5.2.1 App Analytics

Various user interactions with the app were recorded using Google Analytics Events. Events such as market clicks²⁹, application views, scrolling and slider changes were tracked. The purpose was to elucidate the impact of the visualization and how people

²⁹ A click on the “get it” button, which redirects the user to the Google Play Store

interacted with the app. These analytics proved to be less apt to perform exact fine grained calculations, as the system seems to have a high margin of error. Nevertheless, global trends could be identified and are substantially supported by the data.

Engagement Flow

The engagement flow is a standard visualization created by the Google Analytics web interface. It shows the aggregate paths that users take when navigating through the application and where they drop off. Figure 16 shows how users navigate through AppDJ. It should be noted that the screen counting in this diagram does not take into account multiple app views within the same iteration. This means that if a user looks a two or more apps without changing the sliders, this will not result in two screen views in the engagement flow.

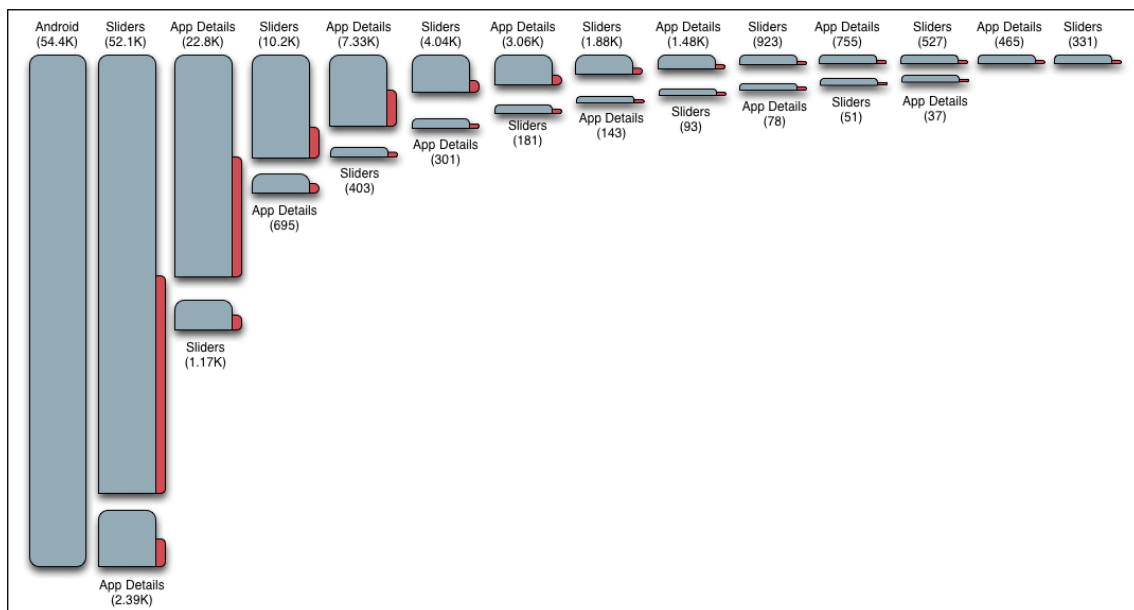


Figure 16: The engagement flow of AppDJ

The engagement flow in Figure 16 clearly shows a logarithmic scale of interactions down to a maximum of 13 steps. This visualization shows the two paths that are possible, either first changing the sliders or looking at an app. Apparently, most people prefer changing the sliders before looking at apps, than selecting an app from the recommendations generated with the random initial values of the sliders. We can also see that more than half of all sessions only changed the sliders once, while a fifth of the sessions last more than three steps. This is inline with the finding that the average number of screens per session is 3.8. Another interesting observation is that the drop

off rate intermittently drops to 27.8% from 55.4% in the second step only to jump back up to 44.9% in the fourth. From then on the drop off rate stabilizes at around 20-30%. In the case that the user first selected an app instead of moving the sliders (4.38% of sessions), the drop off rates are marginally lower in the first three steps until they move towards equality with the drop off rates of the more popular path. The lower drop off rate to the second step, can be explained in that these users have not yet used the sliders, so they will be more likely to change them, if the app they looked at does not suit them. The fact that even the next two steps have significantly lower drop off rates in this path, cannot be interpreted conclusively. One explanation could be that these users get “hooked” more by this variant of moving through the visualization. It could be that when looking at an app and then setting the profile, is more effective in engaging the user, thus resulting in more steps and screen views.

UI Interaction

The analytics also exposed some curious results for how people use certain elements of the app, which were not expected based on the informal user studies done on previous beta versions of the app.

The most surprising finding was in the app details screen (Figure 12c in the section Design). The bar which can be clicked or dragged to display the description of an app was used differently than by users that were shown the app and observed in person. The data shows that over 85% of the users drag the bar instead of clicking on it, which is roughly the inverse of what was observed in person. This might be because dragging is an established action for Android users, because the system bar has been draggable far longer than in iOS.

The data also shows that most people understand the concept of sliders and how they work. Only 11% of the changes to sliders consist of click actions, all others are changes that were done by dragging the slider up or down. Moreover, out of all slider actions in the beginning of the app deployment, around 13% were clicks, one month later this number was reduced to around 9%. This shows that a majority of the userbase not only grasped the concept quickly, but the mental model behind the sliders seems to be developing.

Finally, the analysis of how people explore the apps in the list of recommended apps revealed people’s behavior in terms of how far down the list they search before

changing the criteria (changing the sliders). An overwhelming amount (62%) of lists were only explored within the first fifth, and just over half of the app feature graphics in 90% of the requested lists were viewed. In 9% of the requests, people scrolled through 80% of the list before recalibrating the profile. In very rare cases (0.0027%), users scrolled all the way down to the bottom of the list. This indicates that when the profile was set to the liking of the user, most found an appropriate app within the top half of the list. If they did not, most did not have the patience to sift through the entire list. For tablets the behavior was similar, except that those users never scrolled all the way to the bottom of the list.

App Market Clicks

Actual installs of apps could not be tracked, however redirects to the Google Play Store i.e. when the user clicks on the “get it” button, could be monitored. These actions are called market clicks and, while not representing exact installs, they can give an indication of a general trend of interest. Nevertheless, these numbers should not be taken at face value, as Margine [18] found that there could be significant differences between market clicks and the actual installs.

Market clicks were tracked by two separate events, one that compared market clicks to the views of apps and the other to the number of times the sliders were changed. Both of these indicators diverged by 25.7% and 24.9% for total market clicks and tablets respectively. This is an indication that the loss of data is a systemic issue that is rooted in the functionality of Google Analytics. Unfortunately, this means that reasonable assumptions about user behavior can only be made by comparing the same metric, but not across different metrics. Despite this drawback, there are some intriguing findings pertaining to market clicks, views and slider changes of apps across the different categories.

The app was first designed specifically with tablets in mind and only then ported to smaller screens. Consequently, the data was also analyzed with this in mind. Figure 46 in Appendix F shows market clicks in all categories for tablets in comparison with the total market clicks. Figure 46a shows the conversion rate of market clicks, or the percentage of market clicks in terms of views. Figure 46b compares the percentage of total market clicks within each category, or the percentage of total market clicks. A one way analysis of variance (ANOVA) showed that global behavior for both user populations is the same ($p=0.0505$ for conversion rate, $p=0.9999$ for share in total). Nevertheless, some categories such as “Fitness&Health” and

“Music&Audio” have higher conversion rates for tablets.

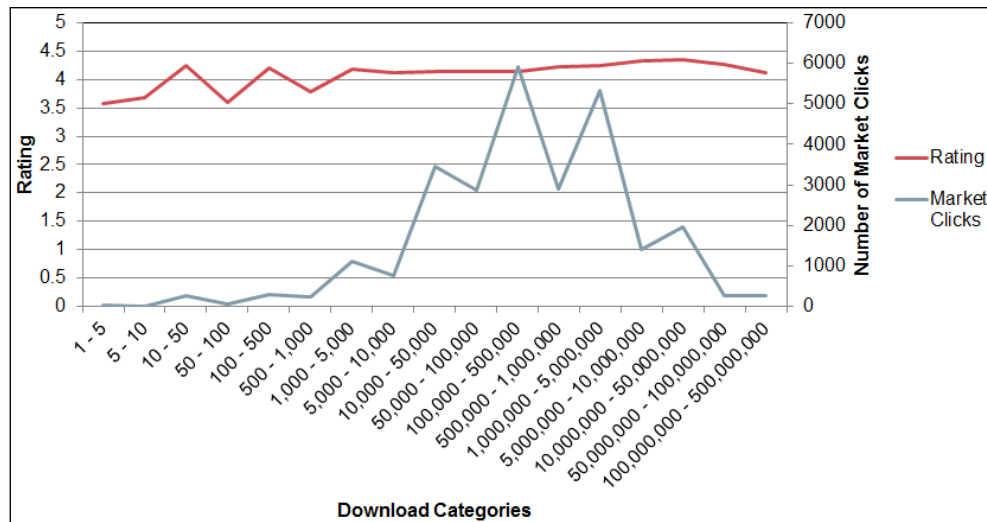
On average 2.7 apps are viewed before there is a market click for an app, with a very low standard deviation (0.25) across all categories. For tablet users the finding is similar, the average being 2.59 apps viewed per market click with a standard deviation of 0.37. Interestingly the metric tracking the number of times the sliders were changed before a market click ensued was even closer. Tablet users changed sliders an average of 1.31 times (standard deviation=0.31), while the global user population changed them 1.30 times (standard deviation=0.17) before clicking the “get it” button.

App Popularity

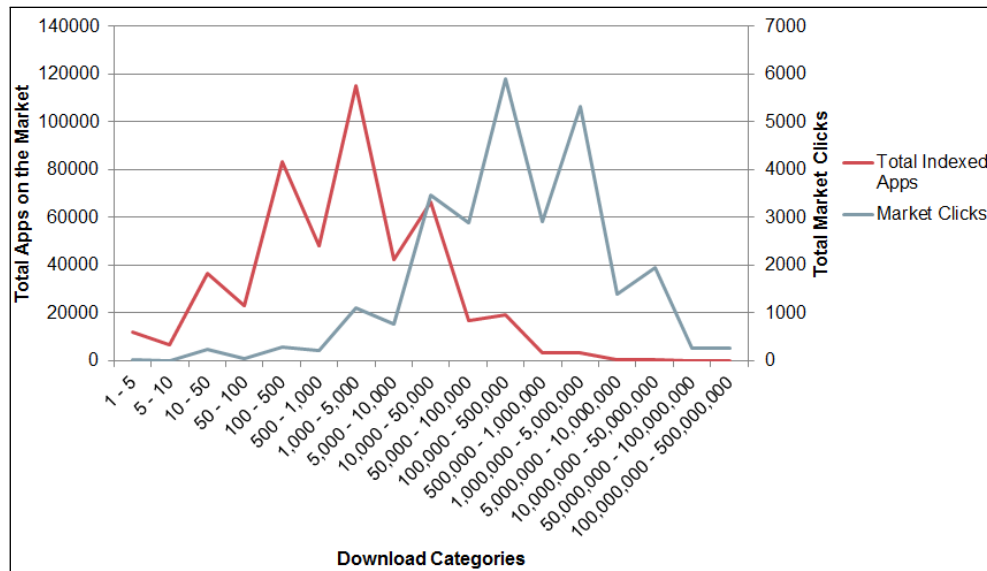
Another metric that was examined was the popularity of the apps that resulted in market clicks. The Google Play Store does not publish the actual number of downloads but instead defines certain ranges of downloads. These “download categories” were used as a measure of popularity of a specific app, assuming that the retention rate is the same. This means that an app in the group of 100,000-500,000 downloads is seen as more popular than an app in the 1,000-5,000 group, because *ceteris paribus* the userbase of the former app is larger. Although Yan and Chen [28] mention the caveats of using download numbers to gauge if a user likes an app, this does not diminish its importance as a metric of popularity. In this case, a popular app is defined as one with widespread adoption, regardless if users like it or not. For certain apps popularity and what users like coincide more closely than with others. For example, there is no alternative to Facebook making it popular but not liked, whereas music fingerprinting services are easily interchangeable, resulting in popular apps also being liked by users. This could explain the disparity between the number of downloads and the rating of an app. Some apps did not have a recorded category (265 apps) due to a system error, as a result these apps were not considered in the evaluation and discarded. Google has changed their download grouping over time, which resulted in some apps belonging to a group that no longer exists. These three groups (<50, 50,000-250,000 and >250,000) were associated with the new groups 10-50, 50,000-100,000 and 100,000-500,000 respectively. These download categories did not include a significant amount of apps, so they did not influence global picture.

In Figure 17a we can see that the rating of apps within the different download categories are constant. This could be an indication that ratings are not a good indicator for popularity. Further analysis of the ratings of apps downloaded confirmed that the rating of an app may indicate success, consistent with the findings of Margine

[18], but not necessarily popularity. Yan and Chen [28] also find that there is a large discrepancy in terms of the popularity of an app and the number of user ratings. In Appendix F in Figure 47 we can see that the ratings in terms of absolute downloads follow a normal distribution with a mean that is consistent with the average ratings in Figure 17a. The absolute number of market clicks in Figure 17a follow a normal distribution, except for a distinct drop within the group 500,000-1,000,000 downloads. Figure 17b plots the same numbers, but compared to the total number of apps in the index within that specific download category. This paints an interesting picture where most apps are below the threshold of 10,000-50,000 downloads, but most market clicks are above this point, where the number of apps in the index declines dramatically. This shows that the majority of mindshare is spread over a small number of apps.



(a) Market clicks and average rating



(b) Market clicks and total apps in index

Figure 17: Distribution of download categories

Session Duration

The final indicator that was investigated was session duration, as this gives an indication of how engaging the app is. For AppDJ the average session length is 2 minutes and 33 seconds, over 58,808 sessions. When disregarding the sessions under 10 seconds this average moves up to 3 minutes and 32 seconds. This session time is comparable to AppAware, where session time is marginally lower. Additionally, the average time of sessions resulting in market clicks were analyzed, demonstrating that longer sessions are an indicator for the amount of market clicks. The average time for these sessions was 6 minutes and 13 seconds, almost double the global session average.

Figure 18 compares the duration of total sessions to sessions with market clicks, plotted on the left y axis. The right y axis shows the number of market clicks per range in session duration. Here we can see that the sessions under 10 seconds skew the data, which implies that the average session duration without this range is more descriptive of actual user behavior. This is additionally supported by the form of the curve showing sessions leading to market clicks.

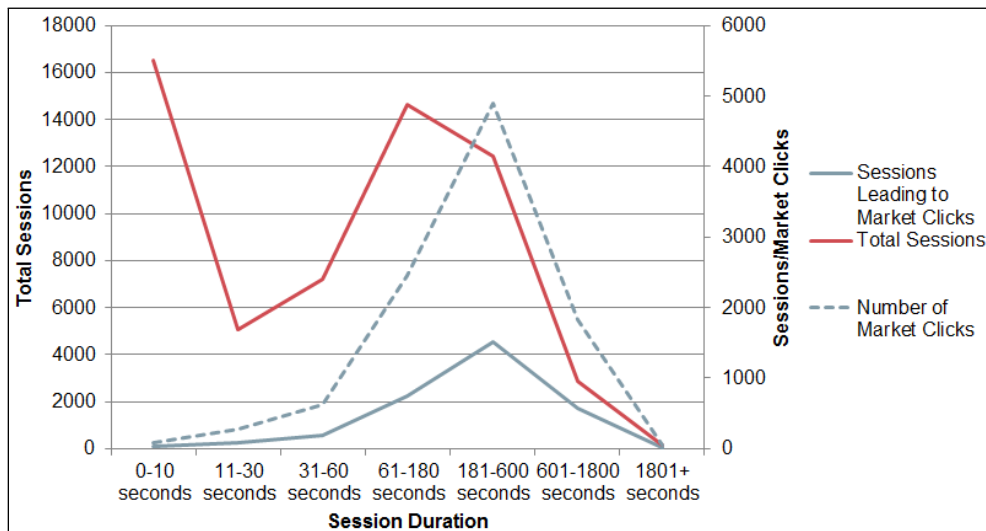


Figure 18: Total session duration compared to sessions resulting in market clicks, overlaid with number of market clicks per session range (y axis on the right shows the market clicks as well as session times resulting in market clicks)

5.2.2 Minor Improvements

Some of the data that was gathered and analyzed showed how people use specific elements of the app. This resulted in direct changes to the UI, the details of which are described below.

- **Slider timeout:** So that multiple sliders can be changed, there is a timeout between the slider action and the request to the backend. In the first release this was 2.5 seconds, which was reduced to 2 seconds. This was done because the analytics showed that on average the time between changing sliders was 1.16 seconds. 2 seconds is sufficient enough for people to change multiple sliders, while not hindering the feeling of responsiveness and usability of the application.
- **Send feedback:** In the previous version, the Action Bar included an icon to send feedback via email. Many people clicked on this icon and sent an email, but did not add any content. Most people probably did not know how to get back to the previous screen so they just sent the email. The next update of AppDJ changed

the “send feedback” button to a “like us on Facebook” button. The idea was to use people’s apparent inclination to play and click around in the app to gain more presence on social media. It is a less obtrusive way to nudge people into sharing the AppDJ app with friends on Facebook.

- **Recommender filters out installed apps:** The first version of the recommender system was very rudimentary, showing the full list received from the backend inquiry without filtering out the apps that were already installed on the phone. The update fixed this missing feature after blogs reviewing the app stated this as desirable. Any apps that are installed on the phone are filtered and new apps within the same category are requested from the backend to replace them.

5.2.3 User Evaluation

For the user evaluation a Google Docs form was used to create a questionnaire and a link was implemented in the revision of AppDJ. The questionnaire data was gathered within the space of three weeks, in which 130 users gave their feedback. The questions are listed in Appendix G.

The five closed ended questions were evaluated and grouped into categories based on the responses. Figure 19 shows the aggregated responses of the first three questions using a likert scale. For the first question there were 127 responses, for the second 126 and for the third 128. The two yes/no questions were meant to determine if the participants used the app in the way that was intended. Most people (70.77%) stated that they did not use AppDJ to find specific apps, while almost all (96.15%) said they used the app to discover new apps. For these questions the number of respondents was 127.

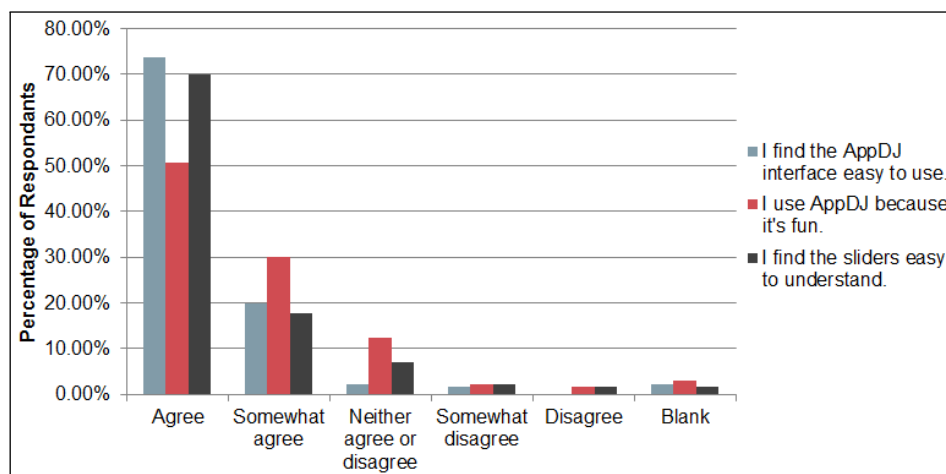


Figure 19: User responses to the three likert scale questions

For the multiple choice question, where users could select the type of apps they installed, the responses were aggregated by category. The deviation in the rank of categories that people reported they installed, and what the analytics picked up in terms of market clicks was 6.88. This large discrepancy could be explained by the difference in market clicks and actual installs that Margine [18] observed.

The responses to the four open ended questions were each clustered according to the nature of the response. For the first question the themes that emerged were comments on discovering and finding apps, the user interface, the simplicity of the app as well as its fun factor and how it gives affordance to discovering less known apps. The next two questions asked the user what they liked or disliked about the current implementation of AppDJ. What people particularly liked was the mixing of categories, the selection and the serendipity of the discovery, the sliders as well as the simplicity of the interface and game like aspects mentioned in the first question. There was significantly less things that people disliked. Some people had difficulties understanding the concept of the sliders, while others had issues with the user interface and the functionality provided by the app. In the final question, the users could give suggestions for other features they would like to see in a future release of AppDJ. Many wanted more sliders and ancillary features such as filters and a search function, others said they would like to keep it as simple as it is now.

6 Discussion

In the course of analyzing the app analytics data as well as reviewing the user evaluation, many intriguing aspects were discovered. The data that was collected showed interesting traits when compared with a baseline condition, which in this case consisted of the AppAware app.³⁰ Moreover, the user evaluation data reinforced many of the ideas and concepts that influence the creation of the AppDJ visualization.

6.1 User Response

The user response to AppDJ was not only ample but also considerably positive. Many blogs featured the app shortly after it was released³¹, commenting on the innovative nature of discovery. This was also reflected in the user survey conducted to evaluate user reception. In the following section the key topics that emerged, as well as some comments are looked at in more detail. Appendix G shows the detailed results of the questionnaire for all 130 participants.

Users stated discovering apps that they would not have searched for and finding non mainstream apps to be the primary motivation to use AppDJ. They also appreciated the novelty of the visualization as well as the minimum effort needed to find new apps. This is nicely portrayed by this user's statement (timestamp: 11/26/2012 17:26):

The sliding catagories[sic] let me search for new apps without the trouble of putting the right keywords todehther[sic]. Iv[sic] managed to find far more aweso.e[sic] apps that suit my needs with appdj than just typing [keywords] into google or some such app market.

This is especially the case for users that are new to the Android platform, who do not yet have a clear idea of the offerings of the app market (timestamp: 11/16/2012 14:16:12, "I am not limited by my lack of knowledge or insight"). Another facet people valued was the non category centric approach. Mixing categories helped people find things they would usually not think of looking for, supporting serendipitous discovery.

³⁰ <https://play.google.com/store/apps/details?id=com.appaware>

³¹ **Lifehacker:** <http://lifehacker.com/5951749/appdj-for-android-asks-for-your-interests-guides-you-to-new-and-useful-apps>,
Addictive Tips: <http://www.addictivetips.com/android/appdj-suggests-android-apps-by-your-interest-level-for-each-category/>,
Make Use Of: <http://www.makeuseof.com/dir/appdj-discovering-apps-innovative-android-22/>,
OMG Droid: <http://omgdroid.com/app-review-appdj-a-simpler-way-to-discover-new-apps/>,
kedDroid: <https://www.youtube.com/watch?v=WraFWe4bGtA>

Additionally, the element of randomness was seen as improving the experience and making the visualization more enticing.

The fundamental concept on which the app was built, was the lean-back approach to visualization. The idea that the system takes the initiative and guides a user through the discovery process was an essential element considered in the design process. The evaluation of the user feedback shows that almost all of the users employed the app for discovery purposes, while a majority utilized the app in the way that it was conceived, i.e. for lean-back discovery and not targeted searching for a specific app. Nevertheless, a more lean-forward feature, such as search was identified by many users to be a desirable feature that would further enhance the usability and utility of AppDJ.

Ease of Use, Engagement and Fun

A common theme among many of the responses was the ease of use and the simplicity of the visualization. From the beginning, a key goal was to create a simple way for users to engage in app discovery. One participant directly confirmed the intuition that for some people and situations an alternative visualization of a top list can be beneficial. This participant (timestamp: 11/27/2012 22:37) stated that “It’s much simpler than browsing through lots of categories and lists.” Out of the ideas of gamification came the conviction that the app should be fun to use, so that the discovery experience becomes engaging. The user study showed that many people appreciate this element of engagement, as “fun” was mentioned in lockstep with “easy” and “simple.” The negative comments lead to the assumption that these elements of the visualization were well solved, as there was only one comment which suggested that the interface was not adequate. This user simply found that “the UI is ugly” (timestamp: 11/12/2012 7:36:50). Another went so far as saying that the interface was too simple for his/her use as a power user (timestamp: 11/12/2012 6:05:15). This underscores the fulfillment of the goal of creating an app discovery tool for users with less experience and interest in the app community.

Slider Concept

One issue that was identified both during the cognitive walkthrough and by showing potential users a beta version of the app was that the concept of sliders might not have a suitable mental model, making it the key hurdle to adoption and understanding the visualization in an intuitive manner. While both the analytics as well as the user

feedback confirm this issue, both also dissuade the impact of a missing mental model and the potential misunderstandings that could arise from the slider visualization. The analytics data indicates that for the most part people understand how the sliders work and that over time the idea becomes engrained within the user population. Numerous participants in the user evaluation expressed their enjoyment with engaging with the slider visualization, even stating that it is the element which makes the discovery experience fun (timestamp: 11/19/2012 13:46:09, “[I] Like the way u[sic] move the sliders it’s easy & fun”)

Of course there were users that had difficulties understanding the sliders and asked for more assistance on how to use them (timestamp: 12/1/2012 0:07, “I am not sure how the sliders work... Some extra explanation would be appreciated”). There was even one user which sent an email requesting help, but these incidents remained exceptions that did not undermine the validity of the slider analogy.

Session Duration

Falaki et al. [10] found that games were most popular with users that had high session lengths. This could indicate that AppDJ has game like qualities, as the average session length is higher than that of more category centric visualizations. Moreover, many users commented that the interface was engaging, fun and easy to use.

The graph (Figure 18 in the section App Analytics) clearly shows that there are many people that opened the app and then immediately closed it, resulting in few or no installs. The rest of the curve of total sessions is normally distributed, peaking in the range of 61-180 seconds. The range where the number of sessions resulting in market clicks is highest, is one step removed, showing that sessions where users went to the Google Play Store to possibly install the app are longer. As a result, the session duration could be used as an indicator for app installs, much like Falaki et al. found that it is a good indicator for app usage.

Visualization Caveats

One major issue people observed was the absence of a search function and additional filtering options. These are also both elements that were determined to be important to people discovering new apps. There were mentions of filtering for paid or free apps and filtering by county. Another problem was the granularity of the search criteria.

While some (timestamp: 11/22/2012 11:15:47, “Ability to search for a specific app and to get recommended apps based on that.”) wanted a way to search for specific apps, others (timestamp: 11/29/2012 4:17, “More sliders for categories of apps.”) were more keen on additional meta categories, in keeping with the pure lean-back approach. Some users suggested additions to the app, such as tracking installs for better recommendations (timestamp: 11/22/2012 20:19:16) or using the Android notifications system (timestamp: 11/20/2012 1:46:45). Finally, there were some issues with the quality of the apps presented, which has less to do with the visualization itself and more with the AppAware API (timestamp: 11/10/2012 7:40:12, “It’s not really the app itself but more a problem with the play store that has many random and low quality apps which turn up in app dj”).

Serendipity in Design

Creating a visualization that enhances serendipitous discovery was a key design goal. To the question of how participants use AppDJ, one user replied: “To find new apps that other wise[sic] I wouldnt[sic] have searched for” (timestamp:11/21/2012 17:17:31). This supports the statement that AppDJ affords serendipity in the discovery process. This is more of an application of the process of supporting people in gaining insight, as described by Yi et al. [29]. In this case serendipity is not a random chance encounter, but a clear cut process that leads users to a set of information that previously required burdensome and time consuming search and browsing to uncover.

In the conceptualization phase, the idea of randomness, as implemented by StumbleUpon, had an important influence in the design. The impact of this influence is shown by the following quote from a participant, that AppDJ is “essentially Stumble Upon for the play store” (timestamp: 11/21/2012 0:07:57). The application of randomness is based on a more “traditional” understanding of designing for serendipity by focusing on the cause, as described by Andre et al. [3]. On the one hand, AppDJ supports random discovery, but also helps people with sagacious insight, an area that Andre et al. [3] found to be inadequately explored.

The user evaluation confirmed that many of the goals set for the visualization as well as the traits and characteristics, line up with user perception. While some drawbacks could be identified, general reception was not only positive, but also reinforced the usability and the legitimacy of the visualization. The issues with the visualization that had been identified before the release were proven to be minor and only have a bearing on fringe groups. As a whole, one can say that AppDJ was well received and

addresses a need within the area of app discovery. Even after two months on the market the active userbase has remained stable, indicating that the app also has longer term appeal.

6.2 Longtail App Discovery

Since it was introduced as a fundamental concept for digital media, the long tail has become a hallmark for all digital services. What the long tail exactly connotes is best described by Anderson himself:

It started like any other demand curve, ranked by popularity. A few hits were downloaded a huge number of times at the head of the curve, and then it fell off steeply with less popular tracks. But the interesting thing was that it never fell to zero. I'd go to the 100,000th track, zoom in, and the downloads per month were still in the thousands. And the curve just kept going: 200,000, 300,000, 400,000 tracks ... In statistics, curves like that are called "long-tailed distributions," because the tail of the curve is very long relative to the head [2, pg 9-10].

Anderson [2] observed this trend first in online retail, but elaborates on the concept, claiming that long tails can be found in virtually all markets. He traces the core elements of the long tail all the way back to the late nineteenth century, where large central warehouses, mass production and the proliferation of the railway system fostered the emergence of a new economic system. Many of the app discovery tools available today concentrate on the top 20% of apps, which Anderson calls the "head" of the long tail. A design goal for AppDJ was driving installs down the long tail, giving more visibility and mindshare to the niche apps and ones on the cusp of mass acclaim. Figure 48 in Appendix F plots the apps from the most popular to the least popular based on the number of market clicks in AppDJ. The x axis shows the rank of the app from 1 to 6847, meaning that out of the total of 21736 market views, 6847 are unique. This graph shows that in the case of AppDJ, the head of the long tail is fairly small, indicating that it is a visualization the serves niche markets.

To conclusively determine if AppDJ does indeed drive installs down the long tail, the market clicks originating from AppAware were compared to those initiated in AppDJ. The market click numbers for AppDJ were normalized in the same way as in Figure 17 in the section App Analytics. The AppAware data was taken directly from the AppAware Analytics Framework³² and normalized in the same way (the category <50 (451 apps) was merged with the category 10-50). The share of market clicks within

³² The numbers were provided by Corneliu Margine

each of the Google Play Store download categories were compared and graphed, the results of which can be viewed in Figure 20. The visualization shows that in all the

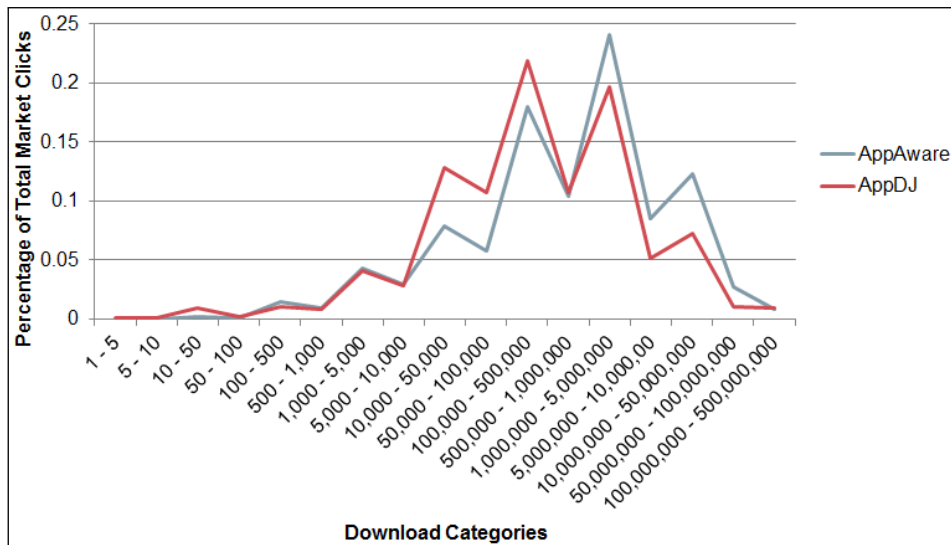


Figure 20: AppDJ market clicks compared to AppAware

download categories up to 5,000-10,000, the path of both AppAware and AppDJ are almost identical. From then on they diverge significantly from one another, only matching once in between 500,000-1,000,000, with a marginal difference of 0.31%. Before this convergence AppDJ has a significantly higher rate of market clicks, whereas above this point AppAware is markedly better at getting users to redirect to the Google Play Store. The clear dichotomy around this one central point shows that AppAware is a visualization that promotes installs of more popular apps, while AppDJ accommodates better visibility for niche apps on the market. As both AppAware and AppDJ pull the recommended apps from the same pool of data, it is the visualization, not the raw data that influences user's behaviors in terms of market clicks. Comments from users also confirm that AppDJ lead them to discover apps off the beaten track, that they would not have found otherwise (timestamp: 11/24/2012 15:13:48, "Much easier to find out about new non-mainstream apps." timestamp: 11/10/2012 15:22:42, "This app lets you find random apps that you would have never found with other apps"). Reflecting upon this data one can state that AppDJ drives app installs down the long tail, making it a possible visualization which could aid the apps of niche developers in gaining more mindshare.

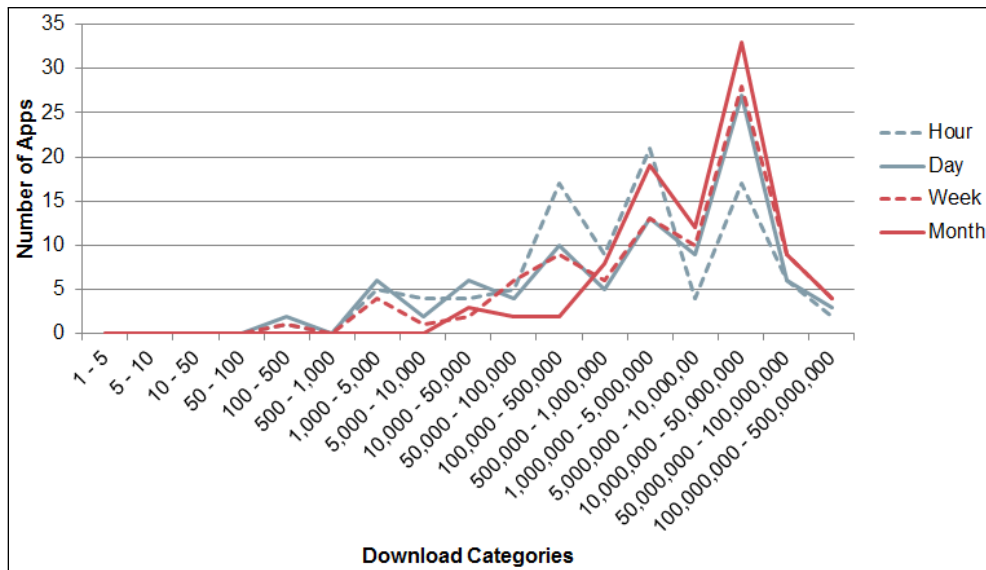
6.3 App Adoption Cycle

The analysis of the analytics data on market clicks in the various download categories threw up some anomalies that could not be explained. Since the shape of the curve is

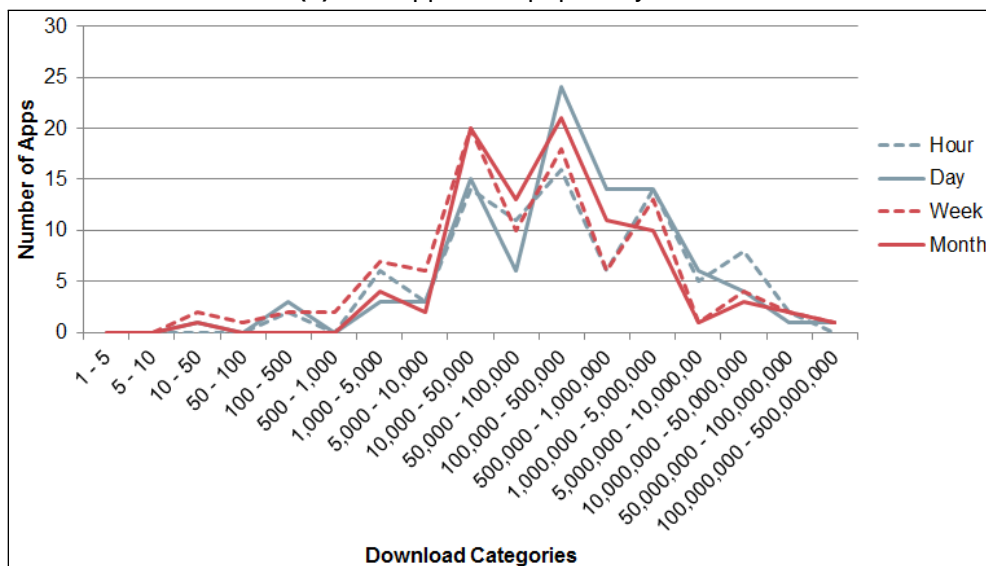
strikingly similar for both AppAware and AppDJ, the conjecture that these are systemic oddities stands to reason. Accordingly, further investigation into the data was conducted.

Examining the market clicks per download category, as shown in Figures 17 and 20, one can see that in every other download category there is a slump in market clicks compared to the previous range. A possible explanation for this dip is that the scale of the download categories is not perfectly logarithmic, but oscillates between orders of magnitudes of 5 and 7.5. This consideration could possibly flatten the curve but does not adequately explain the observed pattern. In an attempt to find a cause of this fluctuation, the popularity and trending list computed by the AppAware backend were analyzed based on the number of apps in the lists for each download category. Figure 21 shows the popularity (Figure 21a) and trending (Figure 21b) lists in the AppAware API as they appeared on the 3 of December 2012. Upon closer inspection one can see that both top lists display peaks and troughs similar to the ones in AppAware and AppDJ market clicks (Figure 20). The peaks in the popularity lists drive the peaks in the upper half of the popularity spectrum i.e. above the range of 500,000-1,000,000 downloads. Correspondingly, the trending lists exert their influence on the vacillations on the lower end. The area below 10,000 downloads is shaped by both types of lists to the same degree, which could explain why both AppAware and AppDJ essentially follow the same pattern in this area.

These two graphs clearly illustrate the difference between popular and trending apps, as well as the aberrations within the lists based on the queried time span. We can observe that lower download rates heavily characterize the trending list, while the apps with more downloads are on the popularity list. This further confirms that the download categories are a good indicator of “app popularity”, something that was discussed in the section App Analytics. It is also apparent that the shape of the graph of market clicks for both discovery tools (Figure 20) are driven by the shape of the popularity and trending lists, best illustrated by Figure 49 in Appendix H. The influence of the trending list on driving installs down the long tail is clearly discernible in the range of 5,000 to 500,000 downloads. Furthermore, overlaying the trending and popularity lists, demonstrates the convergence of the trending and popularity characteristics around the area of 500,000-1,000,000 downloads (Appendix H Figure 49). In the download category 500,000-1,000,000 both lists are weak in espying the appropriate apps to display, as this seems to be the interface between what is defined as trending and what is popular.



(a) The AppAware popularity lists



(b) The AppAware trending lists

Figure 21: The number of apps per Google Play download category

In Figure 17b the total number of apps in each download category is superimposed over the number of market clicks in AppDJ. This not only demonstrates that the larger part of the market is on the lower end of the download spectrum, but also that the significant dip around 500,000 to 1,000,000 downloads cannot be explained by the number of apps on the market. In fact, there are significantly less apps in this category than in the previous one and the crowdedness of the market does not drop dramatically after this point. Nevertheless, this is the only range that departs from the trend line of a standard normal distribution, moreover the ranges around 500,000-1,000,00 are both considerably higher. This points to the fact that there is some other influencer that must be systemic, as both AppAware and AppDJ experience the same pattern. This intuition is amplified by the finding that the rate of

market clicks for both visualizations is nearly the same.

Out of this insight the App Adoption Cycle was born. This cycle is based off the idea behind the Gartner Hype Cycle pictured in Figure 50 in Appendix H. The Gartner Hype Cycle is a research methodology that is used to organize new technologies based on their visibility and maturity [11]. The Hype Cycle is split into five phases, each with special characteristics that define the technology in the specific stage. Comparatively, the App Adoption Cycle is segmented into different phases, albeit seven instead of five. Figure 22 shows the phases through which all apps pass on their path to widespread adoption, or eventual deprecation and discontinuance. Like the Gartner Hype Cycle, the App Adoption Cycle has a peak and a gap, after which the visibility of the app rises anew, to then plateau. As opposed to the Hype Cycle, the visibility of the app stagnates as the app moves towards widespread adoption.

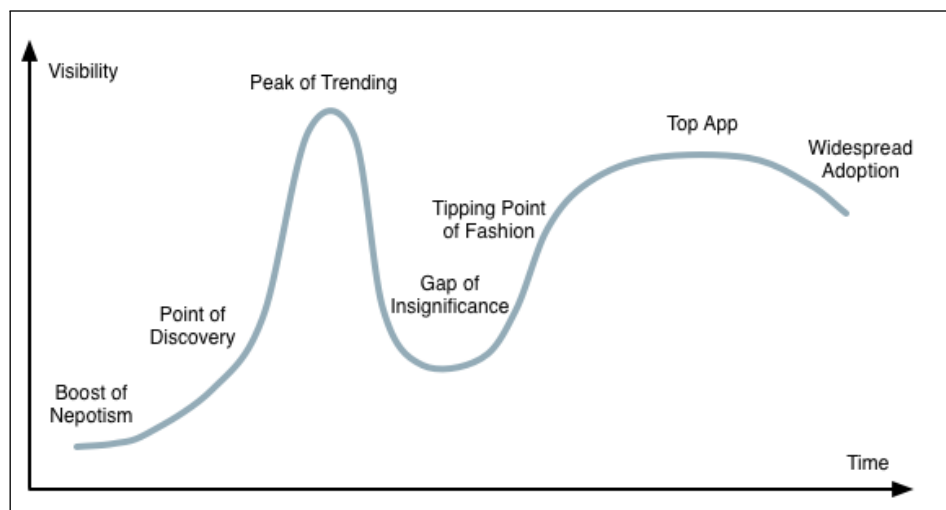


Figure 22: The adoption cycle of apps

Boost of Nepotism

In the beginning of the life of an app, the installs, referrals and recommendations of the people in the developer's immediate network are important. Often not only the first ratings and reviews are from the acquaintances of the developer, but also the downloads. They make up the initial foundation that gives the app a first boost, pushing the app onto the market. In the case that the developer uses paid advertising to promote the app, these early adopters can be seen in the same way.

Point of Discovery

At some point the app is discovered organically by other users. Blogs pick up on the app and review it, introducing it to a wider audience. This is the point where the speed of installs picks up and growth becomes exponential. The app starts to appear in trending lists, further promoting the app.

Peak of Trending

The rate of installs rises up until a certain point at which the continued growth cannot be further sustained. This peak is the pivot point of the “hype” of the app. In the subsequent period installs drop rapidly as the app falls out of trending lists and becomes less compelling. New users increasingly have difficulties discovering the app.

Gap of Insignificance

The rate of installs sinks and discovery becomes exceedingly difficult, until the bottom of the trough is reached. This is the period in which the visibility of the app is at its lowest, making discovery arduous for users.

Tipping Point of Fashion

Once the gulch has been successfully navigated, the app may regain its footing as the install rate rises once again. The visibility of the app increases as the app emerges in popularity lists and discovery become ever more facile for users.

Top App

The app has successfully established itself as a popular app. Through prominent featuring in popularity rankings, the app becomes visible to the mainstream app user. As more people install the app, the growth rate stabilizes and reaches a second, level peak. Retention of the user base becomes increasingly important for the success of the app.

Widespread Adoption

As the market for the app becomes more saturated, the install rate starts to plateau as widespread adoption is reached. In contrast to a top app more people have it on their phone, but the room for additional growth is minimal. In this phase the retention of users becomes important.

Example

Table 2 shows an example of different apps in the different phases of the App Adoption Cycle. The number of downloads in the example is what the general trend of the data shows, meaning that most apps will pass through is phase at that time. Of course this could be slightly different for individual apps.

Phase	Number of Downloads	Example App
Boost of Nepotism	1,000 - 5,000	White Risk ³³
Point of Discovery	5,000 - 10,000	AppDJ ³⁴
Peak of Trending	100,000 - 500,000	LINE cafe ³⁵
Gap of Insignificance	500,000 - 1,000,000	Pocket ³⁶
Tipping Point of Fashion	1,000,000 - 5,000,000	Flipboard ³⁷
Top App	10,000,000 - 50,000,000	Spotify ³⁸
Widespread Adoption	50,000,000 - 100,000,000	Instagram ³⁹

Table 2: An example of the different stages of the App Adoption Cycle

³³ <https://play.google.com/store/apps/details?id=ch.slf.whiteriskmobile>

³⁴ <https://play.google.com/store/apps/details?id=appdj.android>

³⁵ <https://play.google.com/store/apps/details?id=jp.naver.cafe>

³⁶ <https://play.google.com/store/apps/details?id=com.ideashower.readitlater.pro>

³⁷ <https://play.google.com/store/apps/details?id=flipboard.app>

³⁸ <https://play.google.com/store/apps/details?id=com.spotify.mobile.android.ui>

³⁹ <https://play.google.com/store/apps/details?id=com.instagram.android>

7 Conclusion

Clearly there is room in the space of app discovery for visualizations that take a wide array of approaches, without a clear solution that works for everyone. App discovery is a complex topic that requires solutions that consider and embrace this complexity, harnessing it to aid both developers in promoting and users in find apps.

In the beginning a wide range of visualizations were brainstormed. Many of these ideas were outlandish, but they helped focus on the problem in different terms, resulting in the novel approach that characterizes AppDJ. Presenting users with a choice of different designs not only aided the decision making process, but also helped bring to light some of the more subtle factors and less obvious elements that are important to users in the process of discovering new apps. Such an iterative design process assisted in gaining the perspective needed to devise a visualization that is not only distinct from, but also addresses the issues people have with current top lists and category oriented designs. The aspects that most influenced the creation of the visualization were simplicity, easy of use, serendipitous discovery as well as driving app discovery down the long tail. Gamification of the exploring and discovery experience also played into this mix.

The final design that was implemented included features from all these facets. The interface included only two screens, making the navigation within the app drastically simpler than many other discovery apps. Although the sliders were seen as a hurdle to understanding, the novelty and eye catching visualization promoted people to play around. The majority of the users either understood or discovered the functionality behind the sliders with minimal effort. The comments made by users during the evaluation showed that not only was the app simple, easy and fun to use, but also aided people in finding apps that they usually would not have looked for. This strongly supports the notion that AppDJ support serendipitous discovery, by giving people insight into new kinds of apps. Even though the visualization was not explicitly conceived as a game, the intent was to incite engagement and to assist immersive discovery. The fact that users found the interface to be fun to interact with, indicates that game like elements were successfully integrated into the visualization.

Due to the nature of the visualization, some issues arose during the implementation. The sliders are an innovative concept for which no standard widgets existed. As a result a custom widget was built, which is always prone to errors. Additionally the adoption of a backend to power the sliders was needed, seeing that the available data

could only be queries in a category centric way. Although this backend statically maps categories to dimensions and does not allow for detailed recommendations on the level of an app, it is sufficient to create meaningful recommendations. In a future implementation, the backend should be adapted so that the apps map to slider profiles, allowing recommendations to be made at the level of an app. Judging by the user response, the caveats of this type of recommendation are minor at the most, and the mapping is sufficient for powering the visualization thus enabling a truly different form of discovery.

From the analytics data collected during the deployment of the app, a number of interesting findings were made. The market click data showed that in comparison to a category centric visualization like AppAware, AppDJ promotes views of apps in lower “download categories”. As the data that is used as a basis is the same for both apps, this is an indication that the slider visualization of AppDJ drives views, and subsequently installs down the long tail. Additionally, the analysis of the data collected and the comparison with the corresponding data from AppAware brought to light an interesting feature that is constant through both datasets. It was found that not only the form of both market click curves were the same, but also the dramatic drop off in the “download category” 500,000-1,000,000. This gave rise to the idea of an App Adoption Cycle, based on the Gartner Hype Cycle. The App Adoption Cycle is an attempt to explain not only the shape of the curve observed in Figure 20, but also the simultaneous drop off of both AppAware and AppDJ market redirects around the area of 500,000-1,000,000 installs. Apps that are in this realm are in the “Gap of Insignificance”, where neither the popularity nor the trending algorithms can successfully provide for proper visibility. The market click data provides the insight to the possible path that apps go through, and builds the basis where future work can contribute to improving app visibility and the efficiency of recommender algorithms.

The slider visualization has proven to be a simple, engaging and fun way to explore apps. The multi-method evaluation of the interface has made evident the value and usability of this form of app discovery. Moreover, the success of AppDJ on the commercial market and wide acclaim in the blogosphere as well as comparison to established discovery tools⁴⁰, connotes that AppDJ is not just a viable academic prototype, but also a tool with great commercial potential. With over 300 average daily active users and 13,000 downloads, AppDJ has had an impact on users.

⁴⁰ <http://www.makeuseof.com/tag/discover-new-android-apps-with-other-android-apps/>

AppDJ has been successfully in:

- proving a lean-back approach to app discovery
- enabling users to explore the long tail of apps
- mixing categories, thus allowing easier browsing of multiple categories

However, it has become clear that not only the visualization, but also the basic algorithms powering the recommendations play an important role in supporting optimal discovery of apps. This is especially poignant for apps that are in the center of the trending/popularity spectrum.

8 Future Work

In this paper the notion of an App Adoption Cycle was introduced. While the data does clearly indicate that this is a general trend, the validity of this proposition needs to be confirmed on the basis of other datasets. Moreover, the paths of a sample of specific apps should be explored in order to determine how this observation translates from the macro to the micro domain.

The App Adoption Cycle looks at the trend of app adoption that can be observed on the market as a whole. As mentioned above, it is still unclear how this translates to one specific app. For a developer of an app, the repercussions of this adoption cycle on the lifecycle of their app would be of great interest. In this area, retention of users could potentially become critical, the effects of which should be studied in more detail. Moreover, the implications and reasons for the drastic drop in market clicks in the “Gap of Insignificance” are still largely unclear.

The analysis of the market of apps with downloads in the range of 500,000-1,000,000, reveals that the drastic reduction in market clicks and subsequently installs does not stem from an overcrowded market. In fact, at this point the market has already filtered out a considerable amount of apps. From this one can conclude, that the problem lies elsewhere and that the currently available forms of app discovery do not address these issues. The current popularity and trending algorithms cannot solve this problem, although they are sophisticated and apt at determining the right apps above and below this range. In this area the applicability of popularity and trending breaks down and an new paradigm needs to be found, which can adequately propose apps that fit the users needs. One way that should be explored in an attempt to pick the winners from the losers, is to give the user the fine grained control to discover exactly those apps that they specify. This could be done through LDA clustering of app descriptions and then giving the topics in the topic model semantic meaning or assigning certain top keywords as tags on a per app basis. The Game Genome Project⁴¹ attempts to do this with just games. The downside of this approach is manual tagging, which does not scale, especially in the face of the exponential growth of the app market.

⁴¹ <http://gamegenomeproject.com/>

9 References

- [1] ABECASSIS, A. Can Mobile App Discovery Be Fixed? <http://mashable.com/2012/06/14/mobile-app-discovery/>, June 2012. Accessed: 2012-11-08.
- [2] ANDERSON, C. *The Long Tail: How Endless Choice is Creating Unlimited Demand*. London: Random House Business Books, 2006.
- [3] ANDRÉ, P., TEEVAN, J., AND DUMAIS, S. T. Discovery Is Never by Chance : Designing for (Un) Serendipity.
- [4] BAINBRIDGE, D., CUNNINGHAM, S. J., AND DOWNIE, J. S. VISUAL COLLAGING OF MUSIC IN A DIGITAL LIBRARY. *ISMIR 2004 5th International Conference on Music Information Retrieval (2004)* (2004), 397–402.
- [5] BOSWELL, W. How Do People Find Apps? Influencing Factors of App Discovery. <http://software.intel.com/en-us/blogs/2012/12/03/how-do-people-find-apps-influencing-factors-of-app-discovery>, December 2012. Accessed: 2012-12-10.
- [6] BOUCHARD, G. 4 Ways to Help Consumers Discover Your Mobile App. <http://searchenginewatch.com/article/2169475/4-Ways-to-Help-Consumers-Discover-Your-Mobile-App>, April 2012. Accessed: 2012-12-10.
- [7] BRODER, A. A taxonomy of web search. *ACM SIGIR Forum* 36, 2 (Sept. 2002), 3.
- [8] CHURCH, K., AND SMYTH, B. Understanding mobile information needs. *Proceedings of the 10th international conference on Human computer interaction with mobile devices and services - MobileHCI '08* (2008), 493.
- [9] DACHSELT, R. Mambo: a facet-based zoomable music browser. *Proceedings of the 6th international conference* (2007), 110–117.
- [10] FALAKI, H., MAHAJAN, R., KANDULA, S., LYMBEROPOULOS, D., GOVINDAN, R., AND ESTRIN, D. Diversity in smartphone usage. *Proceedings of the 8th international conference on Mobile systems, applications, and services - MobiSys '10* (2010), 179.
- [11] FENN, J., AND RASKINO, N. *Mastering the Hype Cycle: How to Choose the Right Innovation at the Right Time*. Harvard Business Press, 2008.
- [12] GULIK, R. V., AND VIGNOLI, F. Mapping music in the palm of your hand, explore and discover your collection. *Conference on Music* (2004).

- [13] GUNTHER, C. Android Market Now Offers over 250,000 Apps. <http://androidcommunity.com/android-market-now-offers-over-250000-apps-20110714/>, July 2011. Accessed: 2012-12-12.
- [14] HUDSON, C. Mobile App Discovery Is a Developer Problem, Not a Consumer Problem. <http://www.charleshudson.net/mobile-app-discovery-is-a-developer-problem-not-a-consumer-problem>, June 2011. Accessed: 2012-11-10.
- [15] LOCALYTICS. First Impressions Matter! 26% of Apps Downloaded in 2010 Were Used Just Once. <http://www.localytics.com/blog/2011/first-impressions-matter-26-percent-of-apps-downloaded-used-just-once/>, January 2011. Accessed: 2012-11-08.
- [16] LOWE, S. Google Play celebrates 25 billion downloads with 25 cent apps, discounted books, music, and movies. <http://www.theverge.com/2012/9/26/3409446/google-play-25-billion-downloads-sale>, September 2012. Accessed: 2012-12-12.
- [17] LUNDEN, I. IBM: Thanksgiving Sales Data Shows Mobile Commerce Jumping, iPhone/iPad Driving over 20% Of Traffic, Social Nets Only 0.2%. <http://techcrunch.com/2012/11/22/ibm-thanksgiving-sales-data-shows-mobile-commerce-jumping-iphoneipad-driving-19-of-traffic-social-nets-only-0-2/>, November 2012. Accessed: 2012-11-15.
- [18] MARGINE, C. Analytics Framework for Android App Publishers, 2012.
- [19] NIELSEN. State of the Media: U.S. Digital Consumer Report Q3-Q4 2011. http://www.iab.net/media/file/Nielsen_Digital_Consumer_Report_FINAL.pdf, 2011. Accessed: 2012-11-12.
- [20] NIELSEN. Smartphones Account for Half of all Mobile Phones, Dominate New Phone Purchases in the US. http://blog.nielsen.com/nielsenwire/online_mobile/smartphones-account-for-half-of-all-mobile-phones-dominate-new-phone-purchases-in-the-us/, 2012. Accessed: 2012-11-12.
- [21] NIELSEN, J. Writing Style for Print vs. Web. <http://www.useit.com/alertbox/print-vs-online-content.html>, June 2008. Accessed: 2012-11-14.
- [22] NORTH, C. Toward Measuring Visualization Insight. 6–9.
- [23] SHARMA, C. Sizing up the Global Mobile Apps Market. <http://www.slideshare.net/pmork/sizing-up-the-global-mobile-apps-market>, March 2010. Accessed: 2012-12-12.

-
- [24] TORRENS, G. MUSICSTRANDS TM : A PLATFORM FOR DISCOVERING AND EXPLORING MUSIC. *of the... International Computer Music* (2005).
- [25] TORRENS, M., AND HERTZOG, P. Visualizing and exploring personal music libraries. *Proc. ISMIR* (2004).
- [26] VAN DER GRAAF, P. App Store Optimization. <http://searchenginewatch.com/article/2152859/App-Store-Optimization>, February 2012. Accessed:2012-11-07.
- [27] XIE, H. Users' evaluation of digital libraries (DLs): Their uses, their criteria, and their assessment. *Information Processing & Management* (2008), 1–34.
- [28] YAN, B., AND CHEN, G. AppJoy: personalized mobile application discovery. ... *of the 9th international conference on Mobile systems*, ..., February (2011).
- [29] YI, J., KANG, Y.-A., STASKO, J., AND JACKO, J. Understanding and characterizing insights: how do people gain insights using information visualization? ... *for Information Visualization* (2008), 1–6.
- [30] ZIEGLER, C. Android Market stats: South Korea uses most apps, games most popular category. <http://www.theverge.com/2011/12/8/2621035/android-market-stats-infographic>, December 2011. Accessed: 2012-12-12.

A Methods of App Discovery

Looking at what is important to users, one can see that many current discovery solutions provide a suboptimal process or do not support users and developers in the proper way. Listed below in Table 3 are some of the ways that users engage in app discovery.

Method of Discovery		Problems/Issues
Blog Posts/Reviews	Blogs like Mashable, Wired or Lifehacker review and blog about apps on a daily basis. These lists of curated apps are a key source for users.	Most of the recommended apps are already trending or popular. This method also does not directly address the long tail of apps.
Browse Categories	Categories are similar to music genres. They group the apps and show some degree of similarity. Users can browse them to find similar apps, or ones related to specific topics	It takes time and effort to search through different categories. Mixing different categories is not possible.
Recommendations	Social recommendation or recommendation engines showing similar apps.	Needs a profile of the user to give accurate recommendations.
Search	Keywords, filters, categories	The user needs to come up with keywords or a “seed” app. This is a more lean-forward method, whereas discovery should be lean-back.
Word of mouth	People see an app and show their friends	It is a very powerful tool, but takes time and mindshare to acquire users in this way. Also this is something that is not well supported digitally.

Table 3: How users discover apps

B Preliminary Visualizations

The preliminary visualizations were contrived in the view of the fact that people not only explore apps, but also need an entry point into the visualization that allows them to discover apps. This was based on Yi et al's [29] finding that there is a step that precedes exploration in which the user needs an overview before going about further inquiry. This resulted in a two phase approach where the first step is the entry point and the second is the exploration phase, both of which need to be designed appropriately.

B.1 Step 1: Starting Point

B.1.1 Simple Search

If the user knows what they want to explore, they can just search for that app and explore from there. For the use case of this particular visualization, this does not make much sense, because people using this visualization will most likely not know what they are looking for. They may or may not have a general idea of what they are looking for. The aim should be to help the user find something they did not know they were looking for when they started.

B.1.2 Advanced Search

This is useful if the user has a general idea of what they are looking for (some topic or a problem). For this we need to be able to “recommend” certain categories and apps associated with keywords inputted by the user. For these recommendations the user can select a specific app and explore from there. An initial graph or “appstrand” can be used as a starting point.

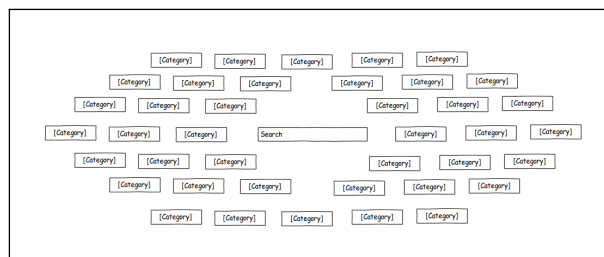


Figure 23: A possible search visualization

B.1.3 Collage

Show a collage of apps and the user selects one that looks interesting. This is a good way to start if the user does not know what they are looking for. There can be some intelligence behind the collage (ordering, filtering, search) to narrow down and specify what types of apps the user wants to see. This integrates the search aspect of finding

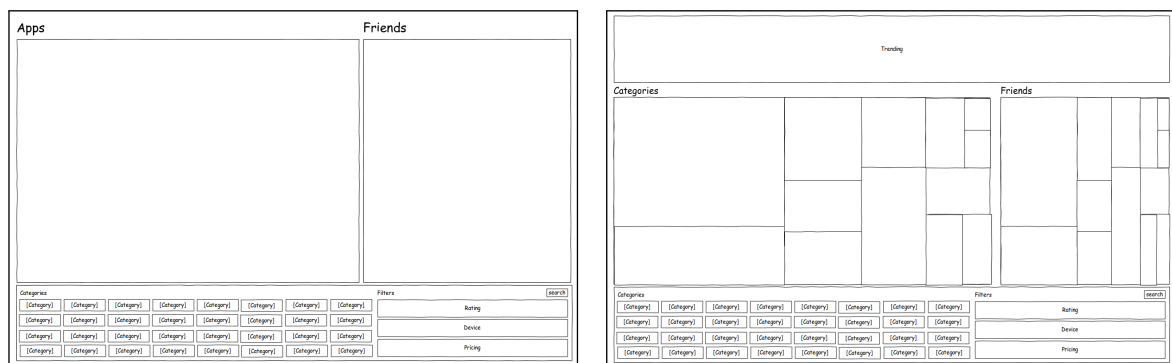
the seed point more naturally. The simplicity of the visualization makes it comprehensible for all users.

B.1.4 TreeMap

This is an improvement on the classical way of selecting a category and then an app in that category. Instead of a list, the categories and apps are arranged in an ordered form. The user can also change the order of the TreeMap. This predicates that the user knows what category the app is in. The approach is a bit like a collage but ordered. This approach is more analytical, and might be over the head of most users. The question is if the visual affordance is properly interpreted, and not too complicated for comprehension (can people see the relationship between different data and spot trends easily). Also there is the issue if this is useful (popular apps already have enough exposure). People might also not be interested in this level of detail.

B.1.5 What's New

What's New is a visualization that takes elements from the Spotify⁴² “What’s new” tab. Sometimes people do not know what they are looking for (analogy of window shopping), in which case it can make sense to have a display window that helps users find something that might interest them. It also provides a starting point for users that just want to explore and find new apps. The existing top lists can be visualized to form something like a display window. Additionally search, category and filter options can be integrated as other access methods for users that have a better idea of what they want to explore. The collage or tree map approach can be of help by allowing a wider range of users with a wider range of “search maturity” to easily find the right app. The approach allows users with a clear idea, to users with no idea what they are looking for, to use the same entrance interface without having to think too much about the maturity of their thoughts themselves.



(a) A basic approach to a start screen

(b) An example of an integrated start screen

Figure 24: What a Spotify style start screen could look like

⁴² <http://www.spotify.com/>

B.2 Step 2: Exploring Apps

B.2.1 Simple Graph

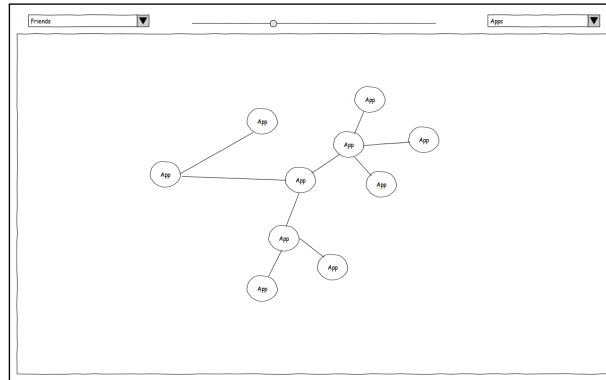
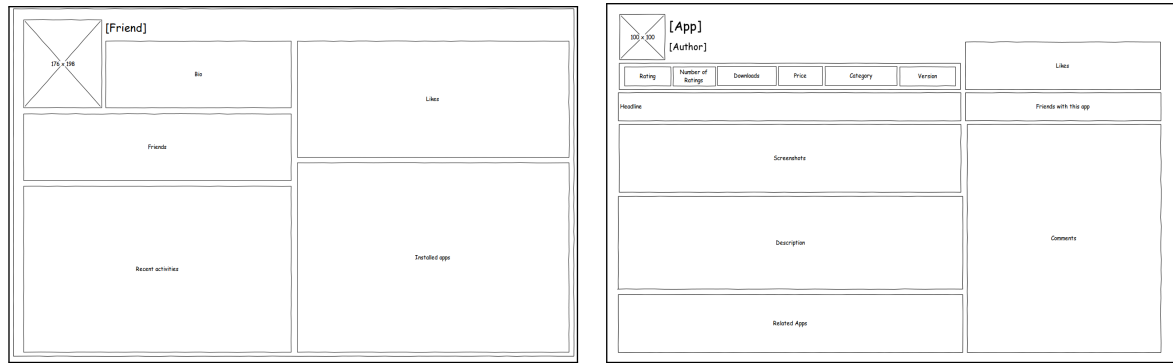


Figure 25: A simple node link graph visualization

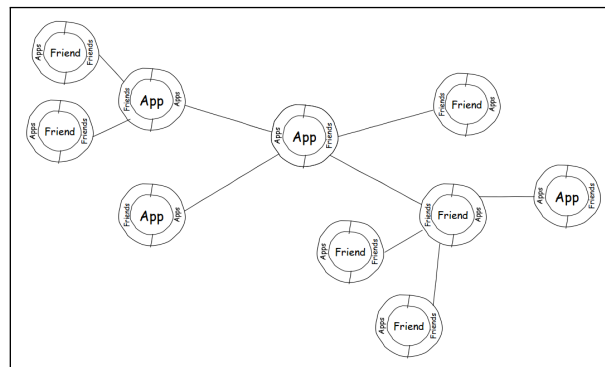
- Switch between social and app graph. Allows the exploration of the two, but not much integration between them.
- Show interconnectivity of apps (use of LDA clustering and some heuristics)
- Problem: this is very static, needs some sort of different process for the initiation
- No filtering
- Bringing in some more information with color coding and node size for “hotness” and rating could aid users.

B.2.2 Social Graph

- Integrate app and social graphs in one coherent visualization.
- Apps on one side of the node, Friends on the other
- Question of organization and efficiently using the space and creating a graph that is easily comprehensible
- Two types of nodes: Apps and friends.
- Allows both app and social graph exploration
- Is a bit of a gimmick, but this is useful. The goal is to attract users and give them an immersive app exploration experience.
- Concept of “Appstrands”: like playlists, a graph of related apps that can be shared with others
- Allows exploration out from an seed app or person



(a) Details shown when a Friend node is clicked (b) Details shown when an App node is clicked



(c) A node link graph incorporating both the social and app dimensions

Figure 26: The social and app graph integrated into one visualization

- Problem: like the simple graph, how do we start?

B.2.3 Advanced Social Graph

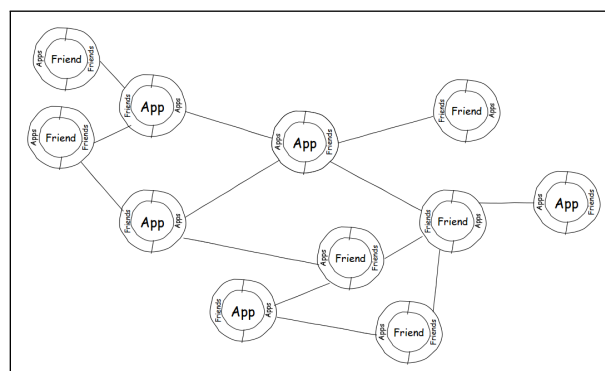


Figure 27: Expanded social graph visualization

- Add additional interconnectivity
- Problem: makes for a complicated, unclear and confusing graph (good for analytics, but not normal users)

B.2.4 TreeMap Categories

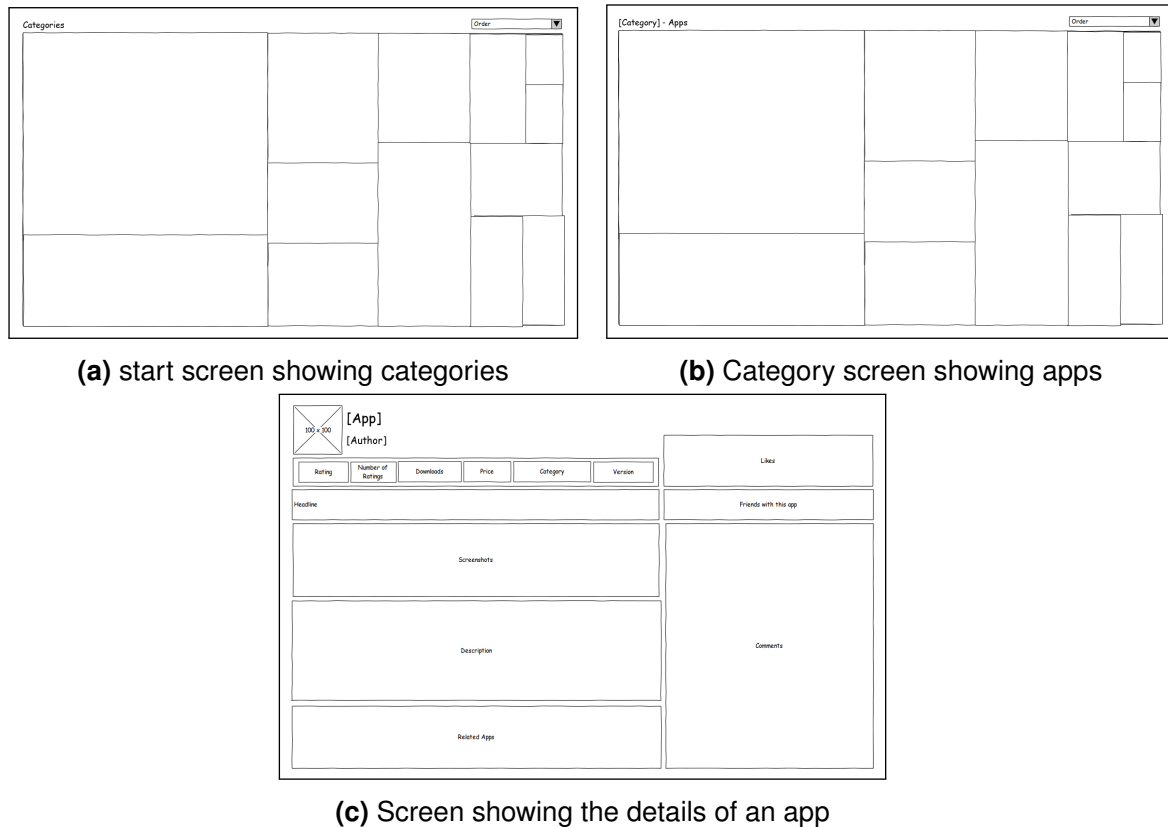
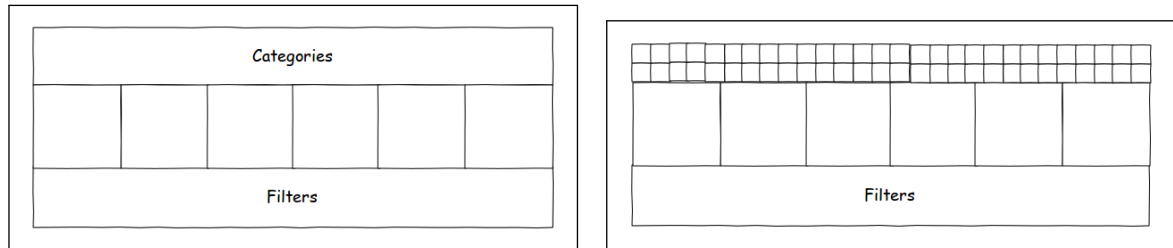


Figure 28: A TreeMap visualization focusing on apps

- Put categories in a TreeMap
- Ranking is done by applying some heuristic that represents the popularity of the apps in that category (can be changed by the user)
- Next screen shows apps in that category with the same ranking
- Integrates step 1 and 2 (start and exploration)
- Problem: Appstrands concept is lost, No browsing over more than one category (no category mixing). Also, social exploration is not possible.

B.2.5 Ribbons



(a) Representation showing the structure of the visualization

(b) An example of how the ribbons look like in practice

Figure 29: Visualization using ribbons

- Show ribbons of related apps in a hierarchal format (abstraction of a simple local relationship, but does not show the whole graph)
- Very simple, does not uses graphing and can be applied to smaller screens
- Has the explorative nature needed.
- Problem: relationships cannot really be represented (hierarchical format). The social aspect is not as integrated. Also does not solve the starting point problem. Showing appstrands (sharing of app discovery paths) is not possible.
- Process: select an app, browse thought apps, select on and then browse though related apps. Note: one category is friends. Categories drop down and enable browsing back to the start.

B.2.6 Flipboard

Use the TreeMap approach, but allow the user to “browse” thought the categories by flipping. Flipping through something is a natural motion that we do when we search for something (looking for records in a record store).

- Pictures are important, recognition better than recall.
- Tiles is an up and coming idea for visualization. There are many tools using a similar visualization.
- Mixes everything together, no clear difference between the app and social graph
- Problem: less of a device for exploration. There is no immersion in app discovery
- Does not work well with a distinct starting point. There might be an option for app discovery if the user does not know where to start.

C Brainstorming Visualizations



Figure 30: App Matcher comparison screen

C.1 Designs

C.1.1 Radar

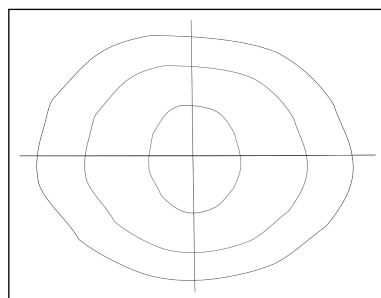


Figure 31: Radar visualization that categories apps

The radar visualization has three rings where the apps shown are based on how similar they are to a seed app. In the innermost ring are core apps that are most related to the target app, the ones in the second are less related and the apps in the third are “far out there”.

C.1.2 Wheel of Fortune

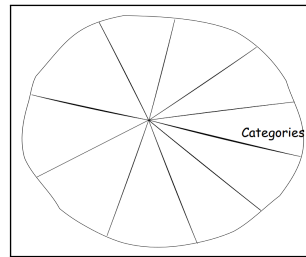


Figure 32: Wheel of fortune visualization that emphasizes randomness

The wheel of fortune visualization is based on the idea of randomness. The user spins the wheel and the category that is chosen is the one that is explored.

C.1.3 Clouds

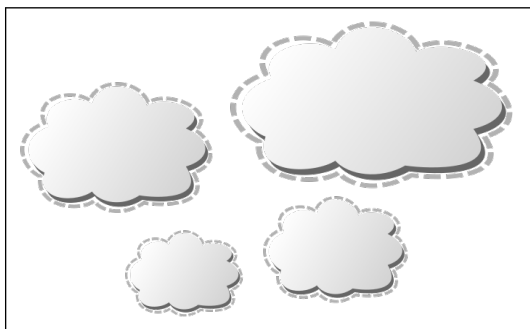


Figure 33: Clouds of similar apps floating across the screen that takes a lean back approach

In the clouds visualization, groups of apps float across the screen and a user can click on an app they find interesting. The apps are grouped according to their similarity. This type of visualization focuses on the relationship between apps, while maintaining a certain amount of randomness (the clouds float by in a random manner). With such a design the user cannot control the visualization, instead it is the system that dictates the path of exploration to some extent.

C.1.4 Clothes Hanger

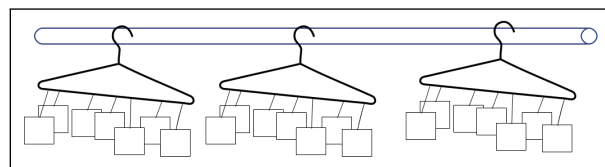


Figure 34: A similar lean back approach with app hangers visualized as clothes hangers

The clothes hanger (or app hanger) idea is similar to clouds, in that apps are grouped by similarity. The difference is that here the user can control the kind of apps they would like to have in the “closet”.

C.1.5 Slot Machine

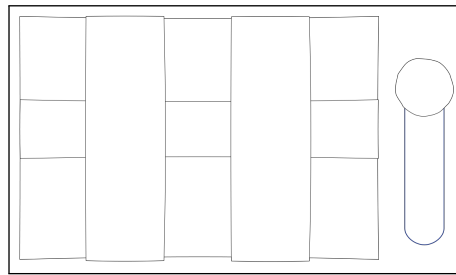


Figure 35: A random gameified approach in the form of a slot machine

The slot machine is another game approach that leverages randomness. Similar to the wheel of fortune, the category or apps are selected by the system and not the user. Thus the user does not have the initiative in app discovery.

C.1.6 “Spotify Radio”

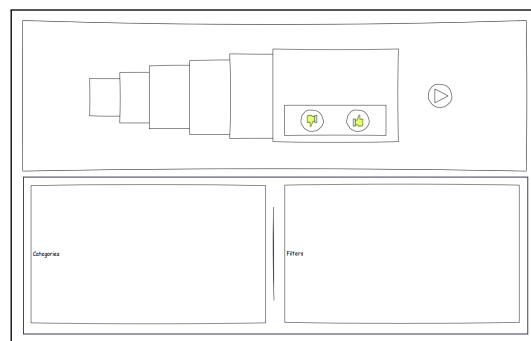


Figure 36: A visualization based on a recommender system that allows user tweaking

This visualization is based off of the radio feature in Spotify. The user can select certain categories, which create a radio or set filters to find an app to base the radio off of. The user can then browse through the apps, giving a thumbs up or thumbs down, so that the recommendations are tuned to the profile of apps that the user would like to see.

C.1.7 Boxes/Crates

This visualization is based on the analogy of looking for vinyls in a record shop. The same idea present in record stores, namely that of looking though boxes and crates of fuzzily grouped records applies. People choose a crate of apps that are similar and then manually browse through apps, finding ones that could be interesting.

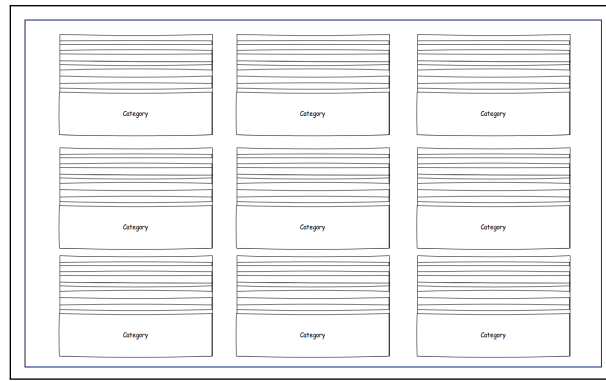


Figure 37: A category centeric approach to browsing apps

C.1.8 Poster Roll

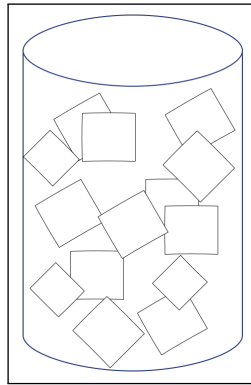


Figure 38: Apps are placed on a poster roll in a random collage visualization

A poster roll is a roll of apps in the form of a collage. The idea is similar to a pinboard, where people can pin any objects that seem interesting. The relationships between the apps shown are random, such that the visualization promotes serendipitous discovery.

C.1.9 Social Nests

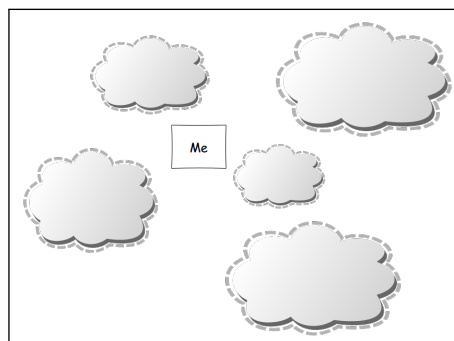


Figure 39: An approach focusing on social groups

Social nests is similar to the clouds approach in that it groups apps. In this case the grouping parameter is social. The visualization shows the user in the center with his/her friends around the perimeter together with the apps they have installed and liked.

C.1.10 Bar

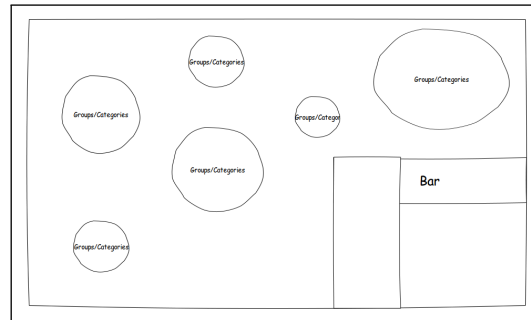


Figure 40: A social grouping approach based on the analogy of a bar

The bar visualization came out of a session where the focus was on brainstorming ideas that are social at their core. In the real world the bar is a place where people interact with one another. In the world of app discovery the idea of a bar translates to different social groups and categories of apps clustered within a “room” and a bar, where discovery is less social and more structured.

C.1.11 Photo Album

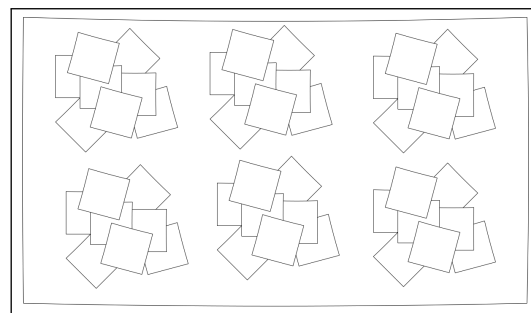


Figure 41: A collage based visualization integrating category grouping

The photo album is based off of a collage visualization similar to the poster roll. Where the poster roll was more random and unstructured, the photo album is more category centric, allowing people to make and share their own “albums” of apps.

C.1.12 Concert

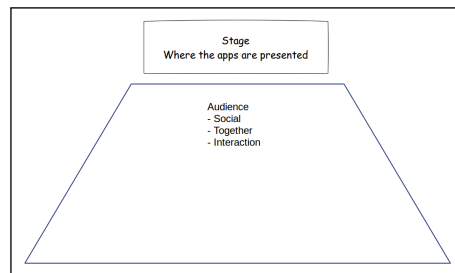


Figure 42: A social visualization that integrates social and app graphs

The concert visualization was born out of the brainstorming session where the focus was on social visualizations of app discovery. The stage represents the area where the apps are presented and the audience are the users that engage with one another, as well as the apps presented on the stage. This is less of a concrete visualization and more of a concept, off of which feasible visualizations could be generated.

C.2 Excursus into Gamification

Figure 43 shows possible ideas that resulted from looking into the gamification of app discovery. During brainstorming, the idea of creating a fun and interactive way to explore apps arose. Some of the subsequent ideas included some gamification aspects, but the idea of creating a game was eventually dropped. The thought which prompted the foray into games was the elements of fun and engagement, both of which were retained in the final visualization.

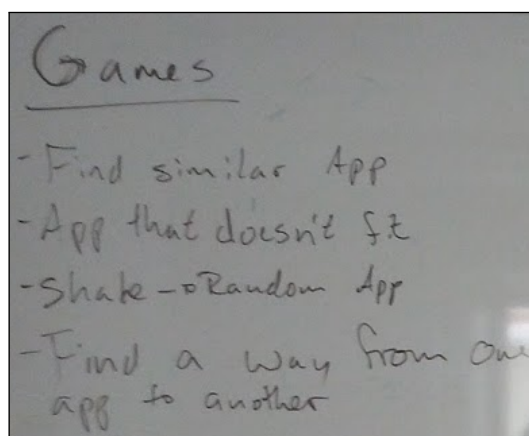


Figure 43: Possible ideas for games

Three of the ideas were puzzle games. One of the ideas concentrated on app discovery by leveraging the element of exploration by having the user find a path from one given app to another by consecutively finding the next similar app. This form of game is

comparable to the “Wiki Game”⁴³. Two other approaches centered on comparing apps, either finding the odd one out or a similar app. That last idea was using a game element (Shaking the phone) to foster random discovery. In the end these ideas were used and implemented to some extent, but not strictly as games.

D Slider Dimensions

In order to associate the traits with the three category sliders, three corresponding dimensions were created, as described in the section Recommender System. After applying a rank for all the traits to all categories, these rankings need to be transposed along two dimensions, so that the six traits could be mapped to the three dimensions of the sliders.

The value per trait associated to each category could range between 0 and 5. Figure 44 shows each category as well as the rank it has for each of the six traits. The allocation of a rank for a category is not based on any numeric metric, but on a simple analysis of a sample of apps within that category. Based on this, I used my intuition to choose a rank that seemed to make sense. For some categories, such as the six games categories (“Arcade&Action”, “Brain&Puzzle”, “Card&Casino”, “Casual”, “Racing”, “Sports Games”) this process was fairly trivial. For others such as “News&Magazines” or “Shopping” it was more difficult, because these categories consisted of a larger array of different kinds of apps.

In a second step the traits needed to be mapped to dimensions. On either side of a dimension there is a trait. Between the two are 10 intermediary steps, where 1 represents the strongest association to the “low” trait and 10 to the “high” trait. Low and high refer to where they are positioned on the slider i.e. for the entertainment slider, entertainment is the high value (activated when the slider is all the way up) and work the low value (active when the slider is all the way down). The naming of the sliders was done by taking the name of the high trait. The method used to place the category within the space between the two traits of the dimension is relatively simple. For every category the two traits of the dimension were placed within the dimension relative to one another. The values of the two traits of one dimension were averaged and spread out across the 10 increments of the dimension. The results of this transformation can be seen in Figure 45a for the entertainment dimension, in Figure 45b for the productivity dimension and in Figure 45c for the lifestyle dimension.

⁴³ http://en.wikipedia.org/wiki/Wikipedia:Wiki_Game

	Entertainment	Work	Productivity	Games	Lifestyle	Utilities	
1 Books and Reference	2	5	2	2	6	1	0
2 Business	1	5	1	3	7	2	3
3 Comics	5	1	9	0	3	4	1
4 Communication	2	3	4	3	0	8	4
5 Education	1	5	1	3	1	7	2
6 Entertainment	5	0	10	2	4	3	5
7 Finance	1	5	1	4	0	9	0
8 Health & Fitness	2	4	3	2	2	5	4
9 Lifestyle	3	2	6	2	1	6	5
10 Media & Video	5	2	8	4	1	8	3
11 Medical	1	5	1	4	0	9	1
12 Music & Audio	5	2	8	3	1	7	4
13 News & Magazines	4	2	7	3	0	8	3
14 Personalization	2	3	4	1	1	5	1
15 Photography	3	3	5	4	4	5	3
16 Productivity	0	4	1	5	0	10	0
17 Shopping	2	4	3	4	0	9	3
18 Social	3	4	4	5	2	6	5
19 Sports	4	3	6	2	3	4	4
20 Tools	0	4	1	5	0	10	1
21 Transportation	2	4	3	5	2	8	1
22 Travel & Local	3	3	5	2	3	4	1
23 Weather	2	3	4	2	1	6	1
24 Libraries & Demo	0	3	2	5	0	10	1
25 Arcade & Action	5	0	10	1	5	1	4
26 Brain & Puzzle	4	0	9	1	5	1	3
27 Cards & Casino	5	0	10	1	5	1	5
28 Casual	4	0	9	1	5	1	5
29 Racing	5	0	10	1	5	1	4
30 Sports Games	5	0	10	1	5	1	5

Figure 44: Traits rank per category

Work	1	2	3	4	5	6	7	8	9	10 Entertainment
1	0	0	0	0	0	0	0	0	0	0
2 Business	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0
5 Education	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0
7 Finance	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0
11 Medical	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0
16 Productivity	0	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0
20 Tools	0	0	0	0	0	0	0	0	0	0
21	0	0	0	0	0	0	0	0	0	0
22	0	0	0	0	0	0	0	0	0	0
23	0	0	0	0	0	0	0	0	0	0
24	0	0	0	0	0	0	0	0	0	0
25	0	0	0	0	0	0	0	0	0	0
26	0	0	0	0	0	0	0	0	0	0
27	0	0	0	0	0	0	0	0	0	0
28	0	0	0	0	0	0	0	0	0	0
29	0	0	0	0	0	0	0	0	0	0
30	0	0	0	0	0	0	0	0	0	0

(a) Work/Entertainment dimension

Games	1	2	3	4	5	6	7	8	9	10 Productivity
1	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0
21	0	0	0	0	0	0	0	0	0	0
22	0	0	0	0	0	0	0	0	0	0
23	0	0	0	0	0	0	0	0	0	0
24	0	0	0	0	0	0	0	0	0	0
25 Arcade & Action	0	0	0	0	0	0	0	0	0	0
26 Brain & Puzzle	0	0	0	0	0	0	0	0	0	0
27 Cards & Casino	0	0	0	0	0	0	0	0	0	0
28 Casual	0	0	0	0	0	0	0	0	0	0
29 Racing	0	0	0	0	0	0	0	0	0	0
30 Sports Games	0	0	0	0	0	0	0	0	0	0

(b) Games/Productivity dimension

Utilities	1	2	3	4	5	6	7	8	9	10 Lifestyle
1	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0
4 Communication	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0
14 Personalization	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0
20 Tools	0	0	0	0	0	0	0	0	0	0
21	0	0	0	0	0	0	0	0	0	0
22	0	0	0	0	0	0	0	0	0	0
23 Weather	0	0	0	0	0	0	0	0	0	0
24 Libraries & Demo	0	0	0	0	0	0	0	0	0	0
25	0	0	0	0	0	0	0	0	0	0
26	0	0	0	0	0	0	0	0	0	0
27	0	0	0	0	0	0	0	0	0	0
28	0	0	0	0	0	0	0	0	0	0
29	0	0	0	0	0	0	0	0	0	0
30	0	0	0	0	0	0	0	0	0	0

(c) Utilities/Lifestyle dimension

Figure 45: Category ranking across dimensions

E Cognitive Walkthrough

Goal/Overall Task: "Find an app you want to install."
music

User Action: Open the application

Before User Action:



User clicks on the AppleTV application icon.

After User Action:



Remove caption

1

1. Will the user be trying to produce whatever effect the action has?
Yes, because it's part of his goal ~~not~~ (if he knows what AppleTV is for).

2. Will the user be able to notice that the correct action is available?
Yes, the icon is visible + former experience.

3. Once the user finds the correct action at the interface, will she know that it is the right one for the effect she is trying to produce?
Yes, former experience in opening app.

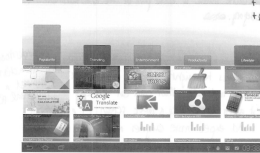
4. After the action is taken, will the user understand the feedback given?
Yes, a new screen shows up showing also the app name in the browser bar.

2

General: I'm wondering how people ~~will~~ might try to use the slider & how it down is a specific category.
In the first half 'music' app = ~~entertainment~~ (4) + ~~entertainment~~ (4) + ~~entertainment~~ (4) + ~~entertainment~~ (4)!

User Action: Set Entertainment Slider

Before User Action:



User slides entertainment slider into the 4th position.

After User Action:



3

1. Will the user be trying to produce whatever effect the action has?
(Remember he's looking for a music app)
Yes, because there is ~~another~~ part of the entertainment category, ~~music~~

2. Will the user be able to notice that the correct action is available?
No, I guess he might click on it because there's no indication that you can slide and it also looks like a static screen (i.e. ~~same~~ ~~same~~ I would expect more categories at the left or right).

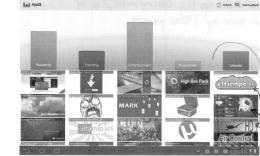
3. Once the user finds the correct action at the interface, will she know that it is the right one for the effect she is trying to produce?
Yes, former experience.

4. After the action is taken, will the user understand the feedback given?
Yes, feedback from the system

4

User Action: Set Productivity Slider

Before User Action:



User drags the productivity slider into the 4th position.

After User Action:



Do the columns look better together? i.e. with different app.

(General question: is it possible to undo changing the slider and get the old list back?)

5

Now, I know that it is a vertical sliding action :)

1. Will the user be trying to produce whatever effect the action has?
Maybe! Depending on whether he's looking for a "real" music app (e.g. creating music) or for a fun music app (e.g. piano)

2. Will the user be able to notice that the correct action is available?
Yes, former experience

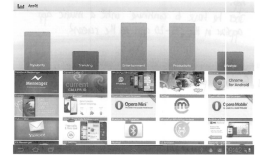
3. Once the user finds the correct action at the interface, will she know that it is the right one for the effect she is trying to produce?
Yes, former experience

4. After the action is taken, will the user understand the feedback given?
Yes, feedback from the system is clear.

6

User Action: Set Lifestyle Slider

Before User Action:



User drags Lifestyle slider into the 5th position.

After User Action:



7

1. Will the user be trying to produce whatever effect the action has?
Maybe! He might also start wanting to browse through the app. Does he have to continue until a music app appears in the 5-10 app on the screen?

2. Will the user be able to notice that the correct action is available?
Yes, former experience.

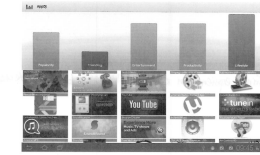
3. Once the user finds the correct action at the interface, will she know that it is the right one for the effect she is trying to produce?
I guess since he again is to navigate down the app.

4. After the action is taken, will the user understand the feedback given?
Yes, former experience from the last step.

8

User Action: Click on SoundHound App

Before User Action:



Click on the SoundHound feature graphic to display the details of the app.

After User Action:



9

1. Will the user be trying to produce whatever effect the action has?
Yes, because it's part of his goal.

2. Will the user be able to notice that the correct action is available?
Yes, given experience in selecting options on a touch screen device.

3. Once the user finds the correct action at the interface, will she know that it is the right one for the effect she is trying to produce?
Yes, given experience.

4. After the action is taken, will the user understand the feedback given?
Yes, given experience, feedback on the screen.

10

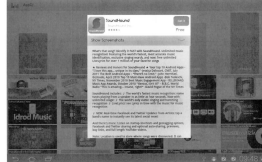
User Action: Look at Description

Before User Action:



The user clicks (or drags) the "Show Description" bar to display the app description.

After User Action:



11

1. Will the user be trying to produce whatever effect the action has?
Probably, if he wants more info.

2. Will the user be able to notice that the correct action is available?
Yes, descriptive label in the UI.

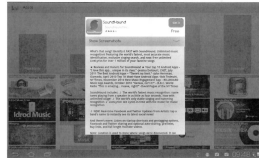
3. Once the user finds the correct action at the interface, will she know that it is the right one for the effect she is trying to produce?
Yes, descriptive label in the UI.

4. After the action is taken, will the user understand the feedback given?
Yes, system shows the description.

12

User Action: Flip through screenshots

Before User Action:



The user closes the description by clicking (or dragging) the "Show Screenshots" bar. Then the user flips through the screenshots using a right to left flick motion.

After User Action:



Wolfs not the user really right? Why did it look a different screenshot?

13

1. Will the user be trying to produce whatever effect the action has?
Probably, if he wants to learn more about this specific app.

2. Will the user be able to notice that the correct action is available?
Yes, label in the UI.
interface element

3. Once the user finds the correct action at the interface, will she know that it is the right one for the effect she is trying to produce?
Yes, descriptive label on the button/interface element.

4. After the action is taken, will the user understand the feedback given?
Yes, system showing screenshot.

14

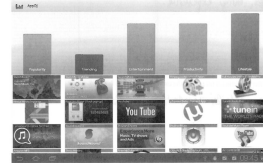
User Action: Dismiss Dialog

Before User Action:



The user clicks outside the app details popup (or the back button) to exit the app description.

After User Action:



15

1. Will the user be trying to produce whatever effect the action has?
Yes, since he removed all the information available for soundboard.

2. Will the user be able to notice that the correct action is available?
Probably because of given experience.

3. Once the user finds the correct action at the interface, will she know that it is the right one for the effect she is trying to produce?
Yes, given experience.

4. After the action is taken, will the user understand the feedback given?
Yes, details window/pop-up disappears.

16

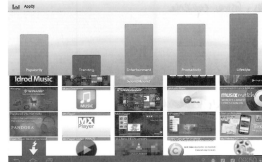
User Action: Scroll

Before User Action:



The user presses, holds and drags the grid-view of app feature graphics up in a scrolling motion.

After User Action:



17

1. Will the user be trying to produce whatever effect the action has?
Yes, if he wants to browse more app.

2. Will the user be able to notice that the correct action is available?
Maybe because of given experience but to me it feels somewhat like you can't scroll up

→ That line doesn't look like you can scroll under

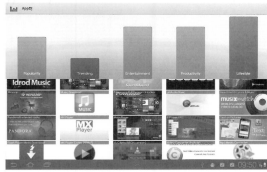
3. Once the user finds the correct action at the interface, will she know that it is the right one for the effect she is trying to produce?
Yes, given experience.

4. After the action is taken, will the user understand the feedback given?
Yes, system's feedback.

18

User Action: Click on Pandora app

Before User Action:



User clicks on the Pandora feature graphic.

After User Action:



19

1. Will the user be trying to produce whatever effect the action has?

Probably, if he wants to learn more about this specific app.

2. Will the user be able to notice that the correct action is available?

Yes, given experience.

3. Once the user finds the correct action at the interface, will she know that it is the right one for the effect she is trying to produce?

Yes, given experience.

4. After the action is taken, will the user understand the feedback given?

Yes, details pop up as expected.

20

User Action: Install

Before User Action:



User clicks on the "Get it" button, which opens the Google Play Store from which the app can be installed.

After User Action:



21

1. Will the user be trying to produce whatever effect the action has?

Yes, part of his goal.

2. Will the user be able to notice that the correct action is available?

Yes, labeled button + given experience.

3. Once the user finds the correct action at the interface, will she know that it is the right one for the effect she is trying to produce?

Yes, labeled button.

4. After the action is taken, will the user understand the feedback given?

Yes, given experience.

22

User Action: Open the application

Before User Action:



User clicks on the App2D application icon.

After User Action:



1

1. Will the user be trying to produce whatever effect the action has?

Yes, because of app description / label.

2. Will the user be able to notice that the correct action is available?

Yes, because it's like launching any other app.

3. Once the user finds the correct action at the interface, will she know that it is the right one for the effect she is trying to produce?

Yes, given exp.

4. After the action is taken, will the user understand the feedback given?

Yes, because app is visible on the screen.

2

User Action: Set Entertainment Slider

Before User Action:



User slides entertainment slider into the 4th position.

After User Action:



3

1. Will the user be trying to produce whatever effect the action has?

Assuming that she is looking for entertainment and she has prev. exp. in discovering the list shows where to look for entertainment.

2. Will the user be able to notice that the correct action is available?

Yes, because of label, but: can I simply pull the slider up?

3. Once the user finds the correct action at the interface, will she know that it is the right one for the effect she is trying to produce?

Not sure, user could pull the slider up, or just touch it and expect a message "change level's setting".

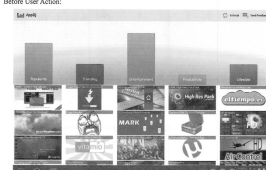
4. After the action is taken, will the user understand the feedback given?

Yes, "wait".

4

User Action: Set Productivity Slider

Before User Action:



User drags the productivity slider into the 4th position.

After User Action:



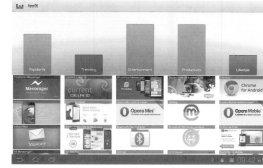
5

1. Will the user be trying to produce whatever effect the action has?
 Yes to desired level - yes but "4th position?"
 Don't get it. Not sure if the feedback is enough.
2. Will the user be able to notice that the correct action is available?
 If the user realizes that she can drag the slider - yes,
 otherwise not sure. Also, not sure what the position means.
3. Once the user finds the correct action at the interface, will she know that it is the right one for the effect she is trying to produce?
 Not before the "next" msg disappears :)
4. After the action is taken, will the user understand the feedback given?
 Yes, "next"

6

User Action: Set Lifestyle Slider

Before User Action:



User drags lifestyle slider into the 5th position.

After User Action:



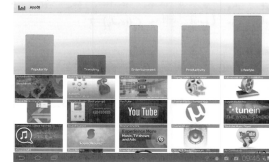
7

1. Will the user be trying to produce whatever effect the action has?
 It's similar to the entertainment slide action,
 so if they figured out the previous one, this is just the same.
2. Will the user be able to notice that the correct action is available?
 Yes, s.o.
3. Once the user finds the correct action at the interface, will she know that it is the right one for the effect she is trying to produce?
 Yes, feedback on msg + next msg.
4. After the action is taken, will the user understand the feedback given?
 Yes, "next"

8

User Action: Click on SoundHound App

Before User Action:



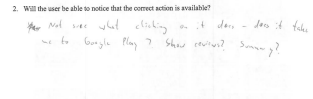
Click on the SoundHound feature graphic to display the details of the app.

After User Action:

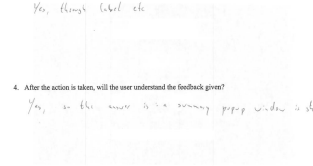


9

1. Will the user be trying to produce whatever effect the action has?
 She wants to install SoundHound, so I guess "yes".
 But why do I choose something from the column
 with the (word setting (ready)?
2. Will the user be able to notice that the correct action is available?
 Not sure what clicking on it does - does it take
 me to Google Play? Shows content? Something?



3. Once the user finds the correct action at the interface, will she know that it is the right one for the effect she is trying to produce?
 Yes, through label etc.



10

4. After the action is taken, will the user understand the feedback given?
 Yes, as the name is in wrong prop window is shown.

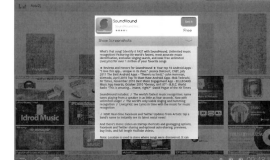
User Action: Look at Description

Before User Action:



The user clicks (or drags) the "Show Description" bar to display the app description.

After User Action:



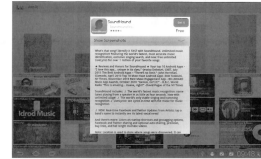
11

1. Will the user be trying to produce whatever effect the action has?
 Yes, because she wants to know more about the app.
2. Will the user be able to notice that the correct action is available?
 Yes, label.
3. Once the user finds the correct action at the interface, will she know that it is the right one for the effect she is trying to produce?
 Yes, because the small arrow on the right.
4. After the action is taken, will the user understand the feedback given?
 Yes - text is visible.

12

User Action: Flip through screenshots

Before User Action:



The user closes the description by clicking (or dragging) the "Show Screenshots" bar. Then the user flips through the screenshots using a right to left flick motion.

After User Action:



13

1. Will the user be trying to produce whatever effect the action has?
 (I'm blind...) - the screenshots are being shown.
 Yes, she wants picture.
 That is, opening screenshots is obvious, but flipping not so much.
2. Will the user be able to notice that the correct action is available?
 Yes, because description option is visible.
 But flipping: no, no hints visible.
3. Once the user finds the correct action at the interface, will she know that it is the right one for the effect she is trying to produce?
 Opening screenshots - yes (click), flipping - maybe not.
4. After the action is taken, will the user understand the feedback given?
 Yes - screenshots are being shown.

14

Before User Action:



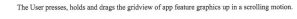
- known through prev. exp.

- yes - Prev. exp.

- Yes, prev. exp.

- Yes - window is closed.

Before User Action



- In this particular example, the app ~~is~~ ^{provides} us
tormented, so it has scintillating affluences along the case?
No wonder as the right.

- Maybe, if the screen always looks like this.
But scrolling ~~it~~ would make it more obvious.

- Yes, from prev. exp.

- Yes, because apps are scrolling

Before User Action



- Yes, some as clicking as SoundHand.

- Yes, label of app.

- Yes, prev. exp.

- Yes, app preview visible.

Before User Action:



- Look for an install ball.

- Probably, though 'get -it' is a sub-optimal option.
What about 'install'? A Google Play icon?

- Yes, because of captives

- $\frac{1}{2}$ App stor.

Before User Action:



1. Will the user be trying to produce whatever effect the action has?
part of goal ✓

2. Will the user be able to notice that the correct action is available?
Know they have to look for desktop (recognize name/label) ✓

3. Once the user finds the correct action at the interface, will she know that it is the right one for the effect she is trying to produce?
✓

4. After the action is taken, will the user understand the feedback given?
Yes, know OS stand. behavior ✓

User Action: Set Entertainment Slider

Before User Action:

User slides entertainment slider into the 4th position

After User Action:

1. Will the user be trying to produce whatever effect the action has?
Yes, purpose of ~~using~~ using this app (part of orig. goal)
2. Will the user be able to notice that the correct action is available?
No bars don't have visual indication (affordance) of being interactive sliders.
• If text came up to top of slides, might be more obvious
• No way to see what the possible positions are
3. Once the user finds the correct action at the interface, will she know that it is the right one for the effect she is trying to produce?
No
• Label unclear, what does entertainment mean? vs. "lifestyle" for example. Is "games" a better label?
4. After the action is taken, will the user understand the feedback given?
• Would be better if feedback text reflected the setting you picked

User Action: Set Productivity Slider

Before User Action:

User drags the productivity slider into the 1st position.

After User Action:

1. Will the user be trying to produce whatever effect the action has?
—//—
2. Will the user be able to notice that the correct action is available?
—//—
3. Once the user finds the correct action at the interface, will she know that it is the right one for the effect she is trying to produce?
Productivity somewhat clear label. But what is the meaning of dragging the slider to the lowest level? Will it still include any productivity apps?
4. After the action is taken, will the user understand the feedback given?
—//—

User Action: Set Lifestyle Slider

Before User Action:

User drags lifestyle slider into the 5th position.

After User Action:

1. Will the user be trying to produce whatever effect the action has?
In general, it's hard to interpret what the slider positions mean combination of

2. Will the user be able to notice that the correct action is available?

3. Once the user finds the correct action at the interface, will she know that it is the right one for the effect she is trying to produce?

4. After the action is taken, will the user understand the feedback given?

User Action: Click on SoundHound App

Before User Action:

Click on the SoundHound feature graphic to display the details of the app

After User Action:

1. Will the user be trying to produce whatever effect the action has?
Yes, part of user's goal
2. Will the user be able to notice that the correct action is available?
If app off-screen, may not realize that you can scroll. With a lot of apps, hard to browse this mosaic view
3. Once the user finds the correct action at the interface, will she know that it is the right one for the effect she is trying to produce?
Yes, OS standard, tiles clickable
4. After the action is taken, will the user understand the feedback given?
Yes... but what does the screen show? Is it a screenshot?

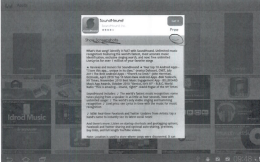
User Action: Look at Description

Before User Action:



The user clicks (taps) the "Show Description" bar to display the app description.

After User Action:



11

1. Will the user be trying to produce whatever effect the action has?

Yes, sure. But might also want reviews, instead of full description

2. Will the user be able to notice that the correct action is available?

Yes, clear label and arrow sign indicates interactivity

3. Once the user finds the correct action at the interface, will she know that it is the right one for the effect she is trying to produce?

Yes, clear label

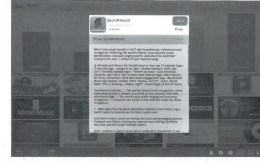
4. After the action is taken, will the user understand the feedback given?

Yes, it shows the description.

12

User Action: Flip through screenshots

Before User Action:



The User closes the description by clicking (or dragging) the "Show Screenshots" bar. Then the user flips through the screenshots using a right to left flick motion.

After User Action:



13

1. Will the user be trying to produce whatever effect the action has?

Maybe? It's not a necessary step, but some might want to do it

2. Will the user be able to notice that the correct action is available?

1. Yes. Show screenshots visible, and they remember the sliding from previous step.
2. Possibly seeing the edge of screenshot indicates where scrolling but may not be obvious depending on tablet/exp.
3. Once the user finds the correct action at the interface, will she know that it is the right one for the effect she is trying to produce?
1. Yes. Label obvious.
2. Maybe. See above

4. After the action is taken, will the user understand the feedback given?

Yes, directly interactive

14

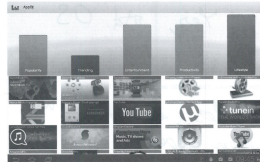
User Action: Dismiss Dialog

Before User Action:



The user clicks outside the app details popup (or the back button) to exit the app description.

After User Action:



15

1. Will the user be trying to produce whatever effect the action has?

Sure, if they don't want this app.

2. Will the user be able to notice that the correct action is available?

Yes, OS standard

3. Once the user finds the correct action at the interface, will she know that it is the right one for the effect she is trying to produce?

Yes, std. OS

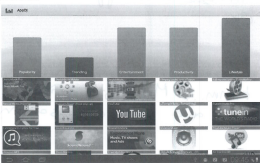
4. After the action is taken, will the user understand the feedback given?

Yes, exactly what they expected

16

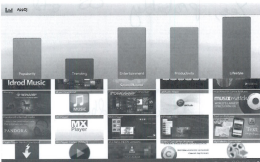
User Action: Scroll

Before User Action:



The User presses, holds and drags the gridview of app feature graphics up in a scrolling motion.

After User Action:



17

1. Will the user be trying to produce whatever effect the action has?

To see more apps?
OK.

2. Will the user be able to notice that the correct action is available?

Hmm... May not notice there are more off-screen.

3. Once the user finds the correct action at the interface, will she know that it is the right one for the effect she is trying to produce?

Yes, probably. OS std.

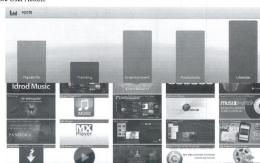
4. After the action is taken, will the user understand the feedback given?

Yes, direct feedback.

18

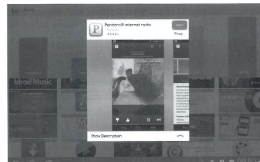
User Action: Click on Pandora app

Before User Action:



User clicks on the Pandora feature graphic.

After User Action:



19

1. Will the user be trying to produce whatever effect the action has?

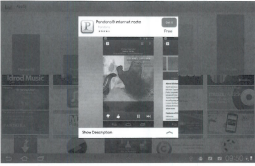
2. Will the user be able to notice that the correct action is available?

3. Once the user finds the correct action at the interface, will she know that it is the right one for the effect she is trying to produce?

4. After the action is taken, will the user understand the feedback given?

20

User Action: Install
Before User Action:



User clicks on the "Get it" button, which opens the Google Play Store from which the app can be installed.

After User Action:



21

1. Will the user be trying to produce whatever effect the action has?

Yes, task

2. Will the user be able to notice that the correct action is available?

Yes, obvious button

3. Once the user finds the correct action at the interface, will she know that it is the right one for the effect she is trying to produce?

Is get it "Android std. label?"
Isn't it "Install"?

4. After the action is taken, will the user understand the feedback given?

No, but thought it would install directly.
(But they can probably deal with it.)

22

User Action: Open the application
After User Action:



23

1. Will the user be trying to produce whatever effect the action has?

Yes, assuming they want more information on the app

2. Will the user be able to notice that the correct action is available?

Yes, the "info" bar is visible

3. Once the user finds the correct action at the interface, will she know that it is the right one for the effect she is trying to produce?

Yes, if they have already interacted with the other bars

4. After the action is taken, will the user understand the feedback given?

Yes, same as previous app action

24

User Action: Get Entertainment Editor
After User Action:



25

1. Will the user be trying to produce whatever effect the action has?

Maybe? Assuming they want entertainment-oriented apps?

2. Will the user be able to notice that the correct action is available?

Maybe? Yes, the entertainment bar is visible.

3. Once the user finds the correct action at the interface, will she know that it is the right one for the effect she is trying to produce?

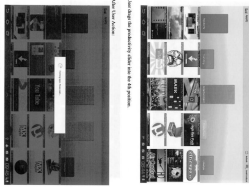
Maybe? It is not completely obvious that the bars are interactive, and it will likely be an impressionistic type of interaction (e.g., not a button click).

4. After the action is taken, will the user understand the feedback given?

Yes, feedback is visible and the message bar communicates clearly that the app is a "category" (e.g., not a button click).

26

User Action: Get Productivity Editor
After User Action:



27

1. Will the user be trying to produce whatever effect the action has?

Maybe? Assuming they want more information on the app

2. Will the user be able to notice that the correct action is available?

Yes, the productivity bar is visible

3. Once the user finds the correct action at the interface, will she know that it is the right one for the effect she is trying to produce?

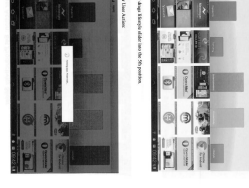
Yes, if they found out that the entertainment bar was interactive, they should know that the other bars are too.

4. After the action is taken, will the user understand the feedback given?

Yes - same as previous action. User may not understand why they got the app they got.

28

User Action: Get Lifestyle Editor
After User Action:



29

1. Will the user be trying to produce whatever effect the action has?

Maybe - that seems to be a connection between lifestyle & music

2. Will the user be able to notice that the correct action is available?

Yes, it's visible

3. Once the user finds the correct action at the interface, will she know that it is the right one for the effect she is trying to produce?

Yes, if they have already interacted with the other bars

4. After the action is taken, will the user understand the feedback given?

Yes, same as previous app action

30

User Action: Click on Soundboard App
After User Action:



31

1. Will the user be trying to produce whatever effect the action has?

Yes, if they are interested in seeing more info for this app

2. Will the user be able to notice that the correct action is available?

Yes, because it looks visible

3. Once the user finds the correct action at the interface, will she know that it is the right one for the effect she is trying to produce?

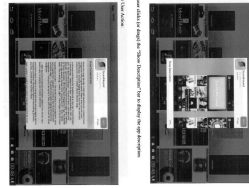
Maybe - it is not 100% obvious that the app icons are interactive, but it's likely they'll try to interact with them.

4. After the action is taken, will the user understand the feedback given?

Yes, feedback is visible and immediate.

32

User Action: Click on Description
After User Action:



33

1. Will the user be trying to produce whatever effect the action has?

Yes, assuming they want more information on the app

2. Will the user be able to notice that the correct action is available?

Yes, the "info" description bar is visible

3. Once the user finds the correct action at the interface, will she know that it is the right one for the effect she is trying to produce?

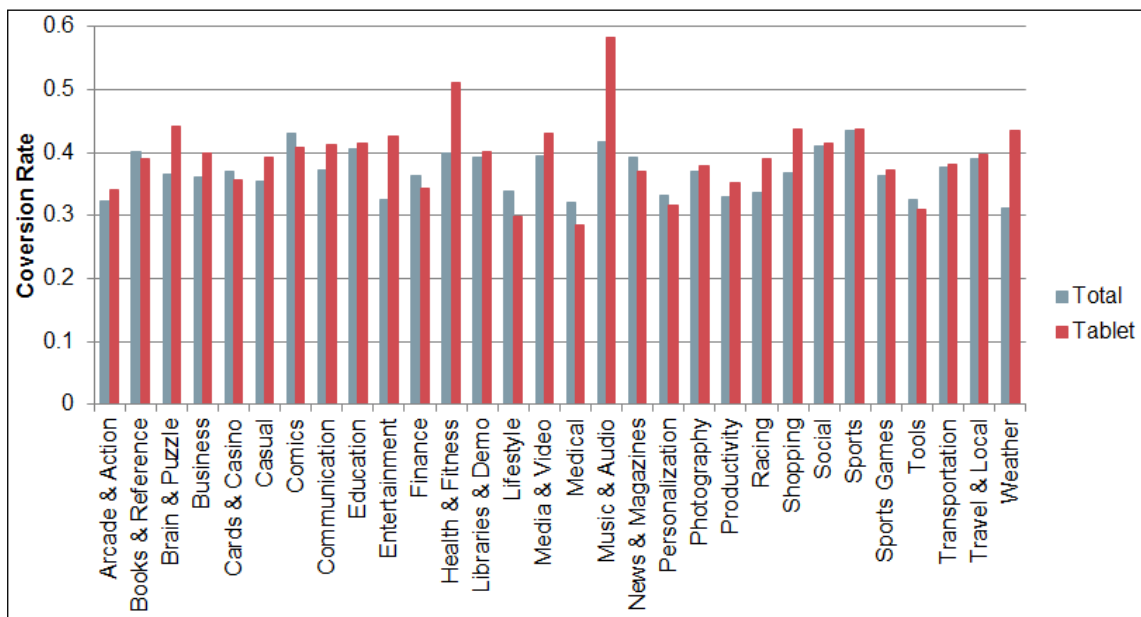
Yes, it's labeled. It may not be obvious that it can go to the next screen, but the link is clearly labeled.

4. After the action is taken, will the user understand the feedback given?

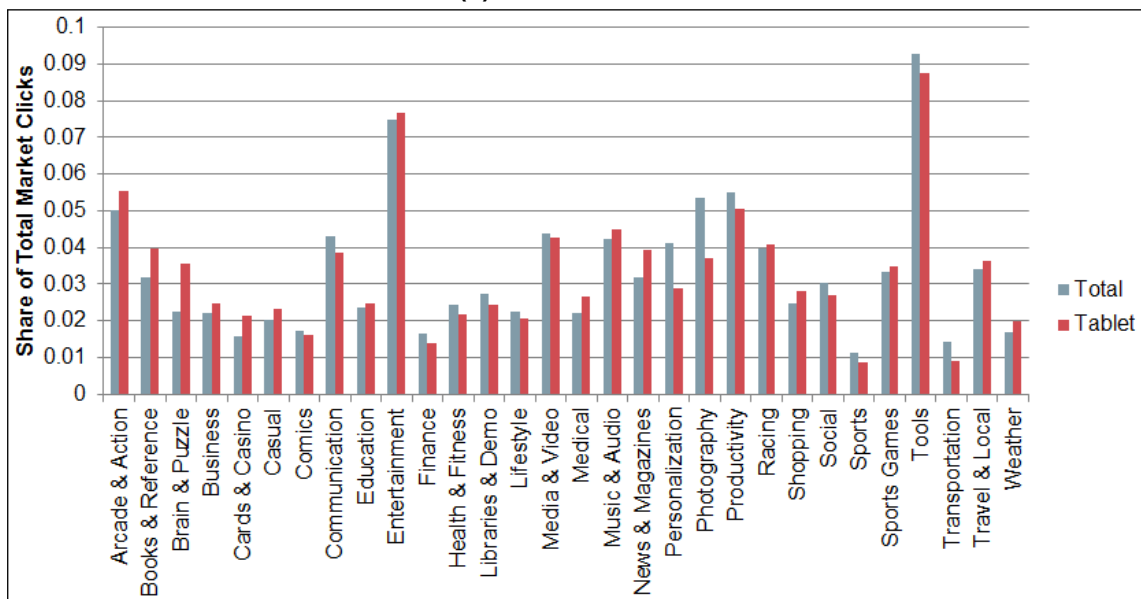
Yes - description is shown - information is visible, and expected.

34

F Analytics



(a) Conversion rate



(b) Share of total

Figure 46: Market click distribution across categories

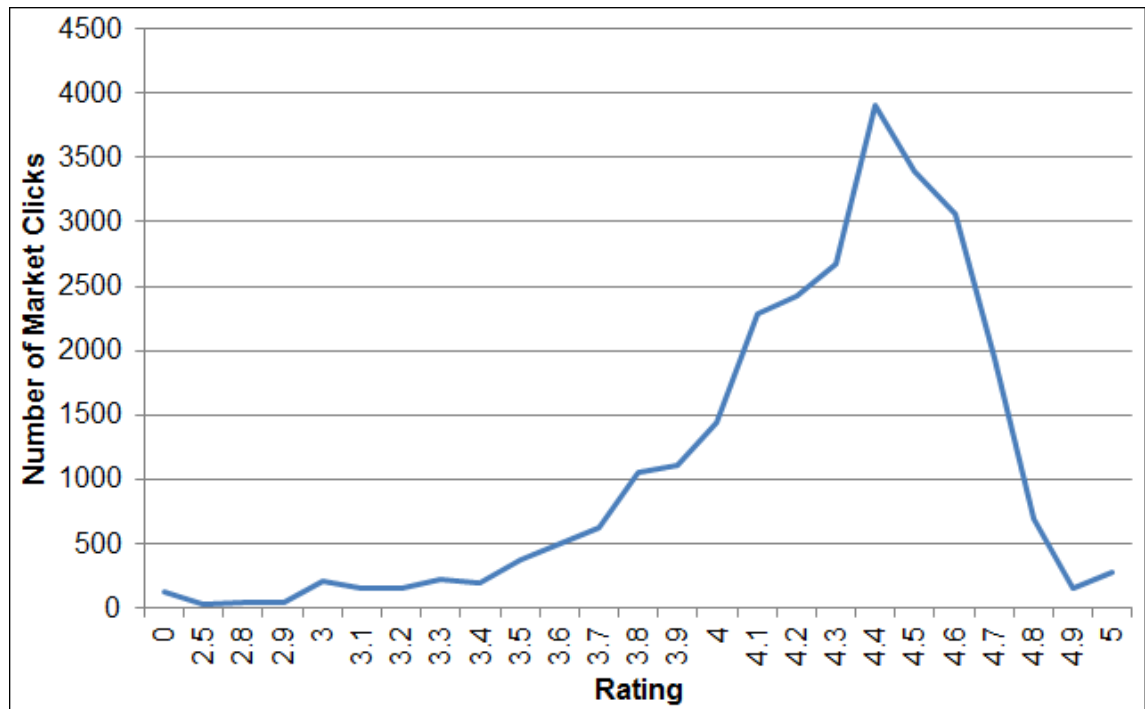


Figure 47: Distribution of ratings based on app market clicks in AppDJ

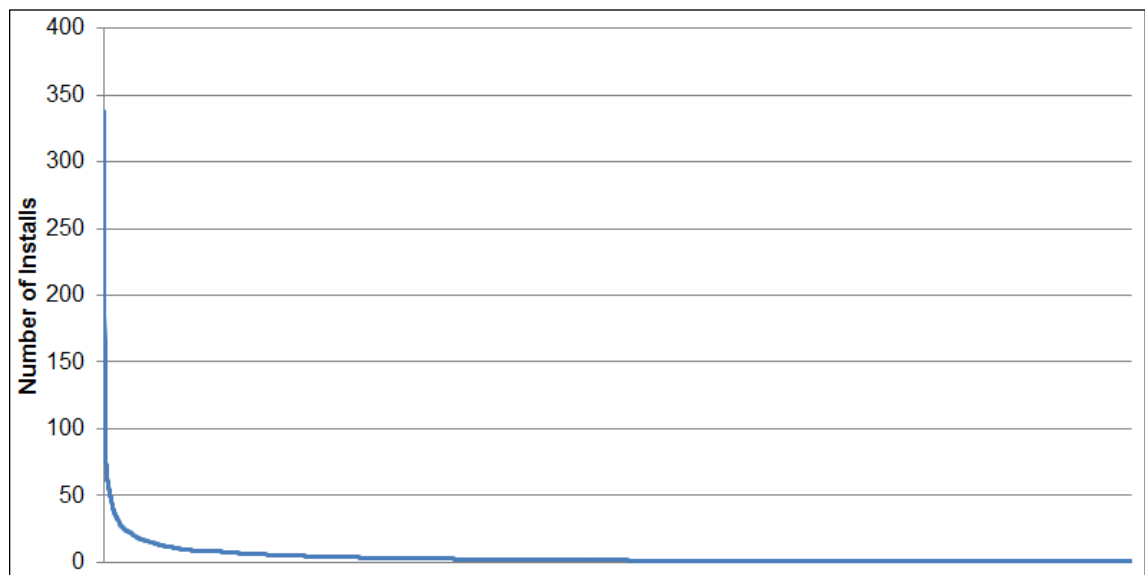


Figure 48: Apps ordered by absolute number of market clicks

G User Evaluation

G.1 Questions

All questions in the questionnaire could be left blank if the user did not want to answer them. A total of 10 questions were asked in the form of the following question types.

Questions using the Likert Scale (Agree, Somewhat agree, Neither agree or disagree, Somewhat disagree, Disagree):

1. I find the AppDJ interface easy to use.
2. I use AppDJ because it's fun.
3. I find the sliders easy to understand.
4. I have used AppDJ to find specific apps.
5. I have used AppDJ to discover new apps.

Open ended questions:

6. Why do you use AppDJ?
7. What do you particularly like about AppDJ compared to other ways of finding and installing apps?
8. What do you particularly dislike about AppDJ compared to other ways of finding and installing apps?
9. What additional features would you like to see added to AppDJ?

Multiple choice question, which displayed all the categories in the Google Play Store and allowed the user to click on the ones they install apps from:

10. What kind of apps do you install?

G.2 Responses

G.2 Responses

Timestamp	I find the AppDJ interface easy to use.	I use AppDJ because it's fun.	I find the sliders easy to understand.	have used AppDJ to find specific apps.	I have used AppDJ to discover new apps.	Why do you use AppDJ?
11/9/2012 16:11:22	Agree	Agree	Agree	No	Yes	Congratulations for your project
11/9/2012 17:12:18	Somewhat agree	Neither agree or disagree	Disagree	No	Yes	
11/9/2012 17:32:43	Agree	Neither agree or disagree	Agree	No	Yes	
11/9/2012 17:48:56	Agree	Agree	Agree	No	Yes	
11/9/2012 18:06:27	Neither agree or disagree	Agree	Somewhat disagree	No	Yes	Gave me recommendation about app I've never heard before
11/9/2012 20:33:24	Somewhat agree	Somewhat Agree	Neither agree or disagree	No	Yes	
11/9/2012 21:14:13	Agree	Neither agree or disagree	Agree	No	Yes	For help in finding applications to keep my tablet entertaining.
11/9/2012 22:46:39	Somewhat agree	Neither agree or disagree	Somewhat agree	No	Yes	to be exposed to new apps that are maybe ones I would not have searched out on my own and yet turn out to be really cool apps I can use and show others.
11/9/2012 23:29:34	Agree	Somewhat Agree	Somewhat agree	No	Yes	
11/10/2012 1:05:29	Agree	Somewhat Agree	Agree	Yes	Yes	To find free apps that replace paid apps To find innovative apps
11/10/2012 1:51:32	Agree	Agree	Agree	Yes	Yes	Find new apps that i didnt know existed
11/10/2012 3:17:27	Agree	Agree	Agree	No	Yes	Find apps I didn't know exist.
11/10/2012 4:14:05	Somewhat agree	Neither agree or disagree	Agree	Yes	No	A newer app feature
11/10/2012 4:45:40	Agree	Somewhat Agree	Agree	No	Yes	
11/10/2012 5:28:42			Somewhat agree	No		
11/10/2012 5:36:06	Agree	Agree	Agree	Yes	Yes	
11/10/2012 6:16:17	Agree	Agree	Agree	Yes	Yes	可以更方便地找到我想要的应用。
11/10/2012 7:40:12	Agree	Somewhat Agree	Agree	No	Yes	I really like the concept and user interface.

11/10/2012 11:33:41	Agree	Agree	Agree	No	Yes	For exploring new apps, its so easy to find new apps with appdj
11/10/2012 12:35:52	Agree	Agree	Agree	No	Yes	Easy to use and links to several download sources.
11/10/2012 13:13:45	Agree	Agree	Agree	Yes	Yes	Facilita la ubicación de las mejores App de forma rápida, amena y eficiente. Gracias por haberla creado y por su constante actualización.
11/10/2012 15:12:44						
11/10/2012 15:22:42	Agree	Agree	Agree	No	Yes	I use appdj because I think it's a great way to find new apps. The play store just let's you see the standard apps all the time. I have to thank coolosertech on youtube for reviewing this app.
11/10/2012 15:34:30	Somewhat disagree	Agree	Agree	Yes	Yes	The idea behind the discovery is very intriguing.
11/10/2012 15:45:21	Agree	Somewhat Agree	Agree	Yes	Yes	
11/10/2012 16:18:12	Agree	Agree	Agree	No	Yes	It's different and allows new apps to be discovered.
11/10/2012 16:27:16	Somewhat agree	Somewhat Agree	Agree	No	Yes	Not much else like it for android market
11/10/2012 17:53:07	Somewhat agree	Agree	Agree	No	Yes	
11/10/2012 19:00:55	Agree	Agree	Agree	Yes	Yes	Because it's easy fun unique features and more
11/10/2012 19:24:12	Agree	Somewhat Agree	Agree	No	Yes	
11/10/2012 19:56:44	Agree	Agree	Agree	Yes	Yes	Because it is easy to use
11/10/2012 21:16:55	Agree	Agree	Agree	No	Yes	
11/10/2012 21:30:46	Agree	Neither agree or disagree	Agree	No	Yes	
11/10/2012 23:32:22	Agree	Agree	Agree	No	Yes	

11/11/2012 1:45:30 Agree	Agree	Somewhat agree	No	Yes	App discovery
11/11/2012 1:46:21 Agree	Agree	Agree	Yes	Yes	It's an interesting and interactive way of discovering new apps.
11/11/2012 2:45:03 Agree	Somewhat Agree	Agree	No	Yes	
11/11/2012 3:52:15 Agree	Agree	Agree	No	Yes	
11/11/2012 9:46:12 Agree	Agree	Somewhat agree	Yes	Yes	In the play store, they always have the top apps and games but how about the others? So I use AppDJ to discover other great apps out there that hasn't been popular like the others.
11/11/2012 13:27:16 Neither agree or disagree	Somewhat disagree	Agree	No	Yes	
11/11/2012 14:34:15 Agree	Disagree	Agree	No	Yes	To discover new apps
11/11/2012 15:32:26 Agree	Somewhat Agree	Agree	No	Yes	Nice interface and most of the time it provides new and unique apps
11/11/2012 16:07:24 Agree	Agree	Somewhat agree	No	Yes	
11/11/2012 18:52:42 Agree	Agree	Agree	Yes	Yes	
11/11/2012 21:27:22 Agree	Somewhat Agree	Somewhat agree	No	Yes	Finding apps
11/11/2012 23:48:09 Agree	Somewhat Agree	Agree	Yes	Yes	
11/12/2012 1:04:01 Agree	Neither agree or disagree	Agree		Yes	
11/12/2012 3:16:35 Agree	Agree	Agree	No	Yes	It's an easy way to discover new apps
11/12/2012 6:05:15 Agree	Neither agree or disagree	Agree	No	Yes	App discovery
11/12/2012 7:36:50 Agree	Disagree	Agree	No	Yes	
11/12/2012 8:48:03 Agree	Somewhat Agree	Agree	No	Yes	
11/12/2012 9:21:24 Agree	Agree	Agree	No	Yes	
11/12/2012 10:21:59 Agree	Agree	Neither agree or disagree	No	Yes	It's an interesting concept

11/12/2012 16:26:24 Agree	Somewhat disagree	Agree	No	Yes	To find new apps
11/12/2012 18:25:16 Agree	Neither agree or disagree	Agree	No	Yes	
11/13/2012 0:19:54 Agree	Somewhat Agree	Neither agree or disagree	Yes	Yes	
11/13/2012 1:00:02 Somewhat agree	Somewhat Agree	Somewhat agree	No	Yes	
11/13/2012 1:37:10 Agree	Agree	Agree	No	Yes	
11/13/2012 5:03:31 Somewhat agree	Somewhat Agree	Agree	Yes	Yes	Great interface to find apps
11/13/2012 16:27:36 Agree	Agree	Agree	Yes	Yes	Because is easy and original to find new app
11/13/2012 17:00:24 Agree	Somewhat Agree	Neither agree or disagree	No	Yes	
11/14/2012 3:45:48 Agree	Somewhat Agree	Agree	No	Yes	new to smart phones. found it to be a cool way to discover new things.
11/14/2012 5:49:12 Agree	Somewhat Agree	Agree	No	Yes	
11/14/2012 6:25:55 Somewhat agree	Agree	Agree	Yes	Yes	
11/14/2012 19:05:27 Somewhat agree	Somewhat Agree	Agree	Yes	Yes	
11/14/2012 19:17:15 Agree	Somewhat Agree	Agree	No	Yes	Because the settings startup ratndomly, I find useful apps that I wouldn't ordinarily have searched for.
11/14/2012 20:01:44 Somewhat agree	Agree	Agree	Yes	Yes	Tot tijd nee interesting apps
11/14/2012 20:49:26 Agree	Agree	Agree	No	Yes	to find out information on apps that may not show up in a cursory search
11/15/2012 0:29:21 Agree	Agree	Agree	No	Yes	I use appdj to discover new apps. I wish there was some way to search for a specific app or game.
11/15/2012 5:26:06 Somewhat agree	Agree	Somewhat disagree	No	Yes	Discover new apps
11/15/2012 7:37:52 Agree	Somewhat Agree	Neither agree or disagree	Yes	Yes	
11/15/2012 18:19:49 Agree	Agree	Agree	No	Yes	Because I like it and it's fu and easy to use.
11/15/2012 19:26:18 Agree	Agree	Agree	Yes	Yes	Fun and intuitive. Hats off to the developer

G.2 Responses

11/16/2012 6:34:00	Agree	Somewhat Agree	Agree	No	Yes	
11/16/2012 10:18:59	Somewhat agree	Agree	Neither agree or disagree	Yes	Yes	
11/16/2012 14:16:12	Agree	Agree	Agree	No	Yes	It guides me to new and interesting applications
11/17/2012 2:56:28	Agree	Somewhat disagree	Somewhat agree	No	Yes	To find new apps.
11/17/2012 16:25:12	Agree	Agree	Agree	Yes	Yes	
11/17/2012 17:25:24	Agree	Agree	Agree	No	Yes	Easy to find cool apps I didn't know I needed.
11/17/2012 19:06:11	Neither agree or disagree	Neither agree or disagree	Neither agree or disagree	No	No	
11/17/2012 23:18:46	Agree	Somewhat Agree	Agree	No	Yes	
11/18/2012 19:36:47	Agree	Neither agree or disagree	Agree	No	Yes	
11/18/2012 21:53:18	Agree	Somewhat Agree	Somewhat agree	No	Yes	great apps
11/19/2012 13:46:09	Agree	Agree	Agree	Yes	Yes	To find new apps of interest to me
11/19/2012 14:28:44	Agree	Agree	Agree	Yes	Yes	
11/19/2012 21:45:46	Agree	Agree	Agree	No	Yes	It is a way of finding apps you otherwise wouldn't know about.
11/19/2012 23:35:11	Somewhat agree	Somewhat Agree	Agree	No	Yes	
11/20/2012 1:46:45	Agree	Somewhat Agree	Agree	No	Yes	I've recently been introduced to the android platform, so I'm looking for new apps to install and AppDJ does just that.
11/20/2012 2:10:38	Somewhat agree		Neither agree or disagree	No	Yes	
11/20/2012 5:13:53	Somewhat agree	Neither agree or disagree	Agree	No	Yes	I sue this app to discover intersting apps
11/20/2012 9:49:08	Agree	Agree	Agree	No	Yes	To find good apps allacross the spectrum
11/20/2012 14:17:39	Agree	Agree	Agree	No	Yes	to see trending, popular apps
11/20/2012 15:36:52	Agree	Somewhat Agree	Somewhat agree	No	Yes	
11/21/2012 0:07:57	Somewhat agree	Somewhat Agree	Somewhat agree	No	Yes	Essentially Stumble Upon for the play store. Pretty sweet and nice work!

11/21/2012 5:53:03 Somewhat agree	Neither agree or disagree	Somewhat agree	No	Yes	
11/21/2012 17:17:31 Agree	Neither agree or disagree	Agree	No	Yes	To find new apps that other wise I wouldnt have searched for
11/22/2012 3:02:25 Somewhat agree	Agree	Agree	No	Yes	
11/22/2012 11:15:47 Agree	Agree	Somewhat agree	No	Yes	To discover apps that I wouldn't have found on my own.
11/22/2012 16:49:45 Agree	Agree	Agree	Yes	Yes	
11/22/2012 19:10:09 Agree	Agree	Agree	Yes	Yes	
11/22/2012 20:19:16 Agree	Somewhat Agree	Agree	No	Yes	To find unusual apps that might not be brought to my attention otherwise.
11/22/2012 20:22:31 Agree	Agree	Agree	No	Yes	The user interface is extremely easy. I use this application to discover other applications I otherwise would not find. Thank you for such an awesome tool!
11/23/2012 0:52:28 Agree	Neither agree or disagree	Agree	Yes	Yes	
11/23/2012 1:33:26 Somewhat agree	Somewhat Agree	Somewhat agree	No	Yes	
11/23/2012 2:34:49 Agree	Agree	Agree	Yes	Yes	This app is fresh. New not the same olds thing. Great idea. And it also pulls apps I would never see because I would get bored just scrolling down an app line.
11/23/2012 17:02:52 Agree	Somewhat Agree	Somewhat agree	No	Yes	
11/23/2012 19:38:13 Agree	Agree	Agree	Yes	Yes	Its a very good app!
11/23/2012 19:51:53 Agree	Agree	Agree	No	Yes	
11/24/2012 0:14:30 Somewhat disagree	Somewhat Agree	Somewhat disagree	Yes	Yes	
11/24/2012 3:27:17 Somewhat agree	Agree	Agree	No	Yes	Concept is new,UI is new,n its fun.
11/24/2012 7:34:43 Agree	Agree	Somewhat agree	No	Yes	Something different
11/24/2012 10:35:14 Somewhat agree	Agree	Somewhat agree	No	Yes	
11/24/2012 11:23:34 Agree	Agree	Agree	No	Yes	
11/24/2012 15:13:48 Somewhat agree	Agree	Somewhat agree	No	Yes	Much easier to find out about new non-mainstream apps.
11/25/2012 19:07:48 Agree	Neither agree or disagree	Disagree	No	Yes	

G.2 Responses

11/26/2012 10:23:21	Agree	Agree	Somewhat agree	No	Yes	Trying many markets to use the best and easiest one
11/26/2012 17:26	Agree	Agree	Agree	No	Yes	I love finding cool new apps and have been looking for an app just like this. Infact appdj is even better than i was looking for. Thank you for your hard work.
11/27/2012 16:31	Agree	Somewhat Agree	Agree	No	Yes	
11/27/2012 16:43	Agree	Agree	Agree	No	Yes	
11/27/2012 18:45	Agree	Somewhat Agree	Somewhat agree	No	Yes	
11/27/2012 22:37	Agree	Agree	Agree	Yes	Yes	It's a easy way to see what are some new apps people are getting, and see if I would like it too.
11/28/2012 2:19	Somewhat agree	Somewhat Agree	Somewhat agree	Yes	Yes	To find new apps.
11/29/2012 4:17	Somewhat agree	Agree	Agree	No	Yes	To discover new apps without tedious searching.
11/29/2012 21:55	Agree	Somewhat Agree	Agree	No	Yes	
11/29/2012 22:07	Somewhat agree	Agree	Agree	No	Yes	
12/1/2012 0:03	Agree	Somewhat Agree	Somewhat agree	No	Yes	To find new apps.
12/1/2012 0:07	Agree	Somewhat Agree	Neither agree or disagree	No	Yes	to find new apps
12/1/2012 3:11	Agree	Agree	Agree	No	Yes	It lets u know at random new apps u mit not know about
12/2/2012 15:28	Agree	Agree	Agree	Yes	Yes	I like experimenting with new apps. This app allows me to find new apps.

Timestamp	What do you particularly like about AppDJ compared to other ways of finding and installing apps?	What do you particularly dislike about AppDJ compared to other ways of finding and installing apps?	What additional features would you like to see added to AppDJ?	What kind of apps do you install?
11/9/2012 16:11:22	The system used			Books and Reference, Business, Comics, Communication, Education, Finance, Media & Video, Music & Audio, News & Magazines, Personalization, Productivity, Shopping, Sports, Tools, Travel & Local, Weather, Arcade & Action, Casual
11/9/2012 17:12:18				Books and Reference, Communication, Education, Entertainment, Health & Fitness, Music & Audio, News & Magazines, Productivity, Shopping, Social, Sports, Tools, Transportation, Brain & Puzzle
11/9/2012 17:32:43				Comics, Communication, Education, Entertainment, Media & Video, Photography
11/9/2012 17:48:56				Health & Fitness, Lifestyle, Productivity, Casual
11/9/2012 18:06:27	Gave me recommendation about app I've never heard before	No settings	Settings: free apps only	
11/9/2012 20:33:24				
11/9/2012 21:14:13	It's fun and simple.	Nothing comes to mind.		Comics, Education, Personalization, Productivity, Shopping, Tools, Arcade & Action, Brain & Puzzle, Casual
11/9/2012 22:46:39	having really discovered that this app is even effective in the achieving said goal.			Books and Reference, Business, Communication, Finance, Health & Fitness, Lifestyle, Media & Video, Music & Audio, News & Magazines, Personalization, Productivity, Shopping, Social, Tools, Travel & Local, Weather, Libraries & Demo, Casual
11/9/2012 23:29:34				
11/10/2012 1:05:29	Visually appealing	I would like it to scan my apps and find similar apps that perform better or provide additional functions I would also like it to suggest useful apps that are not present in my system		Books and Reference, Business, Communication, Education, Media & Video, Music & Audio, News & Magazines, Personalization, Photography, Productivity, Tools, Weather
11/10/2012 1:51:32	Its easy fun looks good	Nothn really	Cant think of anything	Books and Reference, Communication, Education, Entertainment, Health & Fitness, Lifestyle, Media & Video, Medical, Music & Audio, News & Magazines, Personalization, Productivity, Tools, Travel & Local, Weather, Arcade & Action, Brain & Puzzle, Casual, Sports Games
11/10/2012 3:17:27	Fun and interactive.			Lifestyle, Music & Audio, Photography, Productivity, Tools, Weather
11/10/2012 4:14:05	Not sure yet			Business, Social, Sports
11/10/2012 4:45:40				Entertainment, Lifestyle, Music & Audio, Photography, Productivity, Transportation
11/10/2012 5:28:42				
11/10/2012 5:36:06				Books and Reference, Comics, Communication, Education, Entertainment, Finance, Health & Fitness, Lifestyle, Media & Video, Medical, Music & Audio, News & Magazines, Personalization, Photography, Productivity, Shopping, Social, Sports, Tools, Transportation, Travel & Local, Weather, Libraries & Demo, Arcade & Action, Brain & Puzzle, Cards & Casino, Casual, Racing, Sports Games
11/10/2012 6:16:17	我比较喜欢它筛选应用的方式。			Communication, Lifestyle, Media & Video
11/10/2012 7:40:12	Though not apparatus delivering the result I appreciate the ease of use of the ui and what it intends to do	It's not really the app itself but more a problem with the play store that has many random and low quality apps which turn up in app dj	I would use app dj more if it also had a section, like a repository of curated apps that would weed out the sub standard apps of the play store	Books and Reference, Business, Communication, Education, Entertainment, Finance, Health & Fitness, Lifestyle, Media & Video, Music & Audio, News & Magazines, Personalization, Photography, Productivity, Shopping, Social, Sports, Tools, Travel & Local, Weather, Arcade & Action, Brain & Puzzle

11/10/2012 11:33:41	The fun way of searching		The eq for searching cold disappear so the app window can be bigger, you can make it like the address bar in the internet browser, when you scroll down it dissapers .	Education, Lifestyle, Media & Video, Music & Audio, Weather
11/10/2012 12:35:52	Nice graphical interface.	No search function	No search function	Entertainment, Lifestyle, Media & Video, Music & Audio, Photography, Productivity, Shopping, Tools, Arcade & Action, Casual, Racing
11/10/2012 13:13:45	Es de muy fácil manejo		Plugin del Adobe FHas Player	Books and Reference, Entertainment, Finance, Health & Fitness, Lifestyle, Medical, Music & Audio, Photography, Productivity, Shopping, Tools, Travel & Local, Weather
11/10/2012 15:12:44				Books and Reference, Communication, Finance, Music & Audio, Personalization, Photography, Productivity, Social, Tools, Libraries & Demo
11/10/2012 15:22:42	This app lets you find random apps that you would have never found with other apps	That it doesn't give you more apps at a time.	I'd like to see a searchbar just to make this app better than the play store. It would be nice if you could search in catagories as well. You guys did a great job making this app!!	Media & Video, Music & Audio, Personalization, Photography, Social, Tools, Weather, Arcade & Action, Cards & Casino, Racing, Sports Games
11/10/2012 15:34:30	It's more fun. And unexpected.	It gets a bit annoying and confusing sometimes. Especially when I just want to check what's out there, and I'm forced to randomly move the sliders to surprise myself.	I can't think of a way to make it different. I'm sure you guys can. :)	Communication, Entertainment, Lifestyle, Media & Video, Music & Audio, Personalization, Productivity, Social, Tools, Arcade & Action, Brain & Puzzle, Casual
11/10/2012 15:45:21				Productivity
11/10/2012 16:18:12	Easy to use, unique.	Need more sliders or choice to allow the user to narrow his search range.	More sliders.	Books and Reference, Business, Finance, News & Magazines, Personalization, Tools
11/10/2012 16:27:16	The sliders	low res pictures		Books and Reference, Communication, Education, Entertainment, Lifestyle, Media & Video, Music & Audio, Tools, Weather, Arcade & Action, Casual, Sports Games
11/10/2012 17:53:07				Business, Entertainment, Shopping
11/10/2012 19:00:55	It's faster and more accurate			Books and Reference, Business, Comics, Communication, Education, Entertainment, Finance, Health & Fitness, Lifestyle, Media & Video, Medical, Music & Audio, News & Magazines, Personalization, Photography, Productivity, Shopping, Social, Sports, Tools, Transportation, Travel & Local, Weather, Libraries & Demo, Arcade & Action, Brain & Puzzle, Cards & Casino, Casual, Racing, Sports Games
11/10/2012 19:24:12				Entertainment, Music & Audio, News & Magazines, Productivity, Sports, Tools, Cards & Casino, Casual
11/10/2012 19:56:44	the interface is nice and easy to use			Entertainment, Lifestyle, Sports, Sports Games
11/10/2012 21:16:55				Communication, Entertainment, Media & Video, Music & Audio, Sports, Tools, Arcade & Action, Sports Games
11/10/2012 21:30:46				Lifestyle, Personalization, Photography, Productivity, Tools, Weather, Arcade & Action, Brain & Puzzle, Racing
11/10/2012 23:32:22			A "similar apps" (by scores) would be nice. Show the scores in each catageory	Books and Reference, Business, Communication, Entertainment, Finance, Media & Video, Music & Audio, News & Magazines, Photography, Shopping, Tools, Travel & Local, Weather, Brain & Puzzle, Cards & Casino

11/11/2012 1:45:30	It gives different suggestions than other app discovery tools	It seems more random judging by the sliders and results.	Percentages or number values on sliders	Books and Reference, Communication, Entertainment, Finance, Lifestyle, Music & Audio, News & Magazines, Personalization, Photography, Productivity, Shopping, Social, Tools, Weather, Brain & Puzzle, Cards & Casino, Casual
11/11/2012 1:46:21	I like that it's a mixture of all sorts of apps and not just by category.	Sometimes there are repeats of apps.	More categories for the sliders.	Entertainment, Lifestyle, Personalization, Photography, Productivity, Shopping, Social, Tools, Transportation, Arcade & Action, Brain & Puzzle
11/11/2012 2:45:03	Interface		More sliders categories Choice of category (still mixing 4/5 criterias but being able to choose them)	Business, Communication, Entertainment, Finance, Lifestyle, Music & Audio, News & Magazines, Personalization, Productivity, Shopping, Social, Sports, Tools, Transportation, Arcade & Action, Brain & Puzzle, Cards & Casino, Racing, Sports Games
11/11/2012 3:52:15				Books and Reference, Entertainment, Lifestyle, Medical, News & Magazines, Productivity, Tools, Libraries & Demo, Brain & Puzzle, Cards & Casino
11/11/2012 9:46:12	It's organized..	None actually..	More categories, that's all.	Books and Reference, Entertainment, Health & Fitness, Lifestyle, Media & Video, Music & Audio, News & Magazines, Personalization, Photography, Productivity, Social, Tools, Arcade & Action, Brain & Puzzle
11/11/2012 13:27:16				Books and Reference, Education, Lifestyle, Media & Video, Music & Audio, News & Magazines, Photography, Productivity, Shopping, Social, Tools, Weather, Arcade & Action, Casual
11/11/2012 14:34:15	It mixes categories	It's perfect for the way I want to use it.		Books and Reference, Business, Communication, Education, Productivity, Social, Tools
11/11/2012 15:32:26	Nice interface and good results	I never know which apps are relative to my country	Sort by country	Books and Reference, Business, Communication, Education, Entertainment, Finance, Health & Fitness, Lifestyle, Media & Video, Medical, News & Magazines, Personalization, Sports, Transportation, Travel & Local, Weather, Arcade & Action, Brain & Puzzle
11/11/2012 16:07:24				Entertainment, Personalization, Photography, Productivity, Tools
11/11/2012 18:52:42			To add apps to your e-mail adres so you can install the apps at a later date.	Communication, Education, Entertainment, Lifestyle, Media & Video, Music & Audio, Personalization, Photography, Productivity, Tools, Weather, Arcade & Action
11/11/2012 21:27:22	Quite easy			Entertainment, Sports
11/11/2012 23:48:09				Communication, Entertainment, Music & Audio, Social, Sports
11/12/2012 1:04:01				
11/12/2012 3:16:35	Sliders are easy to use.	Lack of filters for free, paid or specific dollar values.	Filter by free, paid app or specific values.	Education, Music & Audio, Photography, Productivity, Tools, Arcade & Action, Brain & Puzzle, Casual, Racing, Sports Games
11/12/2012 6:05:15	Uncomplicated	Very simple to use but I'm not sure I need simple. Choose apps by category and then sort is useful.	Choose category then sort by popular and paid and free. More traditional. (I'm a power user fyi)	Entertainment, Media & Video, Music & Audio, Productivity, Social, Tools, Arcade & Action, Brain & Puzzle, Casual
11/12/2012 7:36:50	The random findings	The UI is ugly		Communication, Health & Fitness, Music & Audio, News & Magazines, Photography, Productivity, Social, Travel & Local, Weather, Racing, Sports Games
11/12/2012 8:48:03				Entertainment, Social, Tools, Casual
11/12/2012 9:21:24				Entertainment, Media & Video, Music & Audio, News & Magazines, Photography, Social, Sports, Travel & Local, Weather, Arcade & Action, Brain & Puzzle, Casual, Racing, Sports Games
11/12/2012 10:21:59	Lack of predictability	Lack of control	Explanations on what makes an app appear in the list corresponding to a particular slider	Entertainment, News & Magazines, Personalization, Productivity, Tools, Travel & Local, Casual

G.2 Responses

11/12/2012 16:26:24				Business, Communication, Health & Fitness, Media & Video, Music & Audio, Personalization, Photography, Productivity, Social, Tools
11/12/2012 18:25:16				Health & Fitness, Music & Audio, Weather, Brain & Puzzle, Cards & Casino
11/13/2012 0:19:54				Books and Reference, Business, Communication, Education, Lifestyle, Medical, Productivity, Shopping, Social
11/13/2012 1:00:02				
11/13/2012 1:37:10				Books and Reference, Entertainment, Music & Audio, News & Magazines, Personalization, Photography, Shopping, Social, Sports, Weather, Arcade & Action, Racing, Sports Games
11/13/2012 5:03:31	Fast and the interface	Like to be able to search paid or free and maybe categories need breaking down more	As above	Business, Communication, Media & Video, Music & Audio, News & Magazines, Photography, Productivity, Shopping, Social, Tools, Weather, Racing
11/13/2012 16:27:36				Communication, Entertainment, Health & Fitness, Media & Video, Music & Audio, Photography, Productivity, Tools
11/13/2012 17:00:24				Entertainment, Lifestyle, Media & Video, Music & Audio, Personalization, Sports, Tools, Arcade & Action, Brain & Puzzle, Racing, Sports Games
11/14/2012 3:45:48				Books and Reference, Communication, Education, Entertainment, Finance, Health & Fitness, Lifestyle, Music & Audio, Productivity, Social, Sports, Tools, Arcade & Action, Brain & Puzzle
11/14/2012 5:49:12				Communication, Productivity, Tools
11/14/2012 6:25:55				Business, Communication, Education, Finance, Lifestyle, Media & Video, Medical, News & Magazines, Personalization, Photography, Productivity, Shopping, Tools, Transportation, Travel & Local, Weather, Libraries & Demo, Brain & Puzzle, Cards & Casino, Casual
11/14/2012 19:05:27				Media & Video, Music & Audio, Personalization, Photography, Tools, Weather, Arcade & Action, Brain & Puzzle, Casual, Racing, Sports Games
11/14/2012 19:17:15	See above		I like it the way it is. Don't ruin it by overloading it. Keep it simple.	Business, Communication, Finance, Lifestyle, Personalization, Photography, Productivity, Tools, Travel & Local, Arcade & Action
11/14/2012 20:01:44	The sliders makes it fun	Nothing	Nothing	Business, Media & Video, Personalization, Productivity, Weather
11/14/2012 20:49:26	that you aren't searching for a specific criteria or name but using general parameters	it can be quite hit and miss. However that is really unavoidable with this implementation	maybe filter results by keyword.	Books and Reference, Communication, Education, Entertainment, Finance, Health & Fitness, Media & Video, Medical, Music & Audio, News & Magazines, Personalization, Photography, Productivity, Shopping, Social, Tools, Transportation, Arcade & Action, Casual, Sports Games
11/15/2012 0:29:21	You can discover new apps and games	You can't search for a specific app or game	The ability to search a specific game and a category where you can find the most downloaded games and apps.	Entertainment, Media & Video, Music & Audio, Arcade & Action, Racing, Sports Games
11/15/2012 5:26:06	Graphic interface	Sliders	Ability to search by generic names and categories	Books and Reference, Business, Communication, Education, Entertainment, Finance, Health & Fitness, Music & Audio, News & Magazines, Photography, Productivity, Shopping, Tools, Weather
11/15/2012 7:37:52				Entertainment, Health & Fitness, Productivity
11/15/2012 18:19:49	It's interface... It's actually really good and easy.	Nothing that I can think about		Entertainment, Lifestyle, Media & Video, Music & Audio, News & Magazines, Personalization, Productivity, Social, Sports, Tools, Arcade & Action, Brain & Puzzle, Cards & Casino, Casual, Racing, Sports Games
11/15/2012 19:26:18	Ease of use			Books and Reference, Comics, Media & Video, News & Magazines, Personalization, Brain & Puzzle

11/16/2012 6:34:00				Communication, Entertainment, Media & Video, Personalization, Photography, Productivity, Shopping, Tools, Travel & Local, Weather, Arcade & Action, Casual, Racing, Sports Games
11/16/2012 10:18:59				Books and Reference, Business, Comics, Education, Entertainment, Media & Video, Music & Audio, News & Magazines, Personalization, Productivity, Sports, Tools, Arcade & Action
11/16/2012 14:16:12	I am not limited by my lack of knowledge or insight			Education, Entertainment, Photography, Productivity, Tools, Brain & Puzzle
11/17/2012 2:56:28	Rather than having someone else tell me what they like I can tell the program what I am looking for	Nothing	I would like some more explanation as to what exactly is used to define the various sliders, for instance it would be helpful to know ow what appdj considers a productivity pp	Entertainment, Media & Video, Personalization, Productivity, Weather, Arcade & Action, Racing
11/17/2012 16:25:12				Entertainment, Arcade & Action
11/17/2012 17:25:24	Significantly better way to find apps that I wouldn't find out about otherwise.	Too many fart noise apps, otherwise consistently helps me find new and exciting apps.	Nothing stands out as needing improvement, besides having less fart apps. I find them crass and unnecessary.	Communication, Education, Entertainment, Finance, Health & Fitness, Lifestyle, Media & Video, Medical, Music & Audio, Personalization, Photography, Productivity, Social, Tools, Weather, Casual
11/17/2012 19:06:11				Books and Reference, Lifestyle, Weather
11/17/2012 23:18:46				Finance, Lifestyle, Productivity, Sports
11/18/2012 19:36:47				Lifestyle, Productivity, Tools
11/18/2012 21:53:18	i can select the searching trend...	nothing	select free or paid apps	Business, Communication, Medical, Music & Audio, News & Magazines, Personalization, Productivity, Tools, Weather
11/19/2012 13:46:09	Like the way u move the sliders it's easy & fun	Nothing in particular		Communication, Music & Audio, News & Magazines, Personalization, Productivity, Social, Sports, Tools, Weather, Casual
11/19/2012 14:28:44				Books and Reference, Business, Comics, Communication, Education, Entertainment, Finance, Health & Fitness, Lifestyle, Media & Video, Medical, Music & Audio, News & Magazines, Personalization, Photography, Productivity, Shopping, Social, Sports, Tools, Transportation, Travel & Local, Weather, Libraries & Demo, Arcade & Action, Brain & Puzzle, Cards & Casino, Casual, Racing, Sports Games
11/19/2012 21:45:46	Straightforward and simple			Entertainment, Lifestyle, News & Magazines, Photography, Tools, Casual
11/19/2012 23:35:11				
11/20/2012 1:46:45	The trending & popular features	No good use of the notifications option	Notifications of when any app on the wishlist drops in price	Books and Reference, Business, Entertainment, Personalization, Productivity, Social, Tools, Brain & Puzzle, Casual
11/20/2012 2:10:38				
11/20/2012 5:13:53	I rally like the way you can nove			Communication, Entertainment, Media & Video, News & Magazines, Photography, Sports, Tools, Arcade & Action, Brain & Puzzle
11/20/2012 9:49:08	There is little to no junk apps in the listing	Nothing in particular		
11/20/2012 14:17:39	it's harder to find new trending application s in play market . appdj with its slider makes super easy			Education, Entertainment, News & Magazines, Productivity, Tools, Transportation, Travel & Local, Weather, Brain & Puzzle, Cards & Casino, Casual, Racing, Sports Games
11/20/2012 15:36:52				Books and Reference, Comics, Communication, Education, Finance, Health & Fitness, Media & Video, Music & Audio, News & Magazines, Personalization, Productivity, Shopping, Tools, Travel & Local, Weather, Libraries & Demo, Arcade & Action, Brain & Puzzle, Cards & Casino, Sports Games
11/21/2012 0:07:57				Business, Communication, Music & Audio, Personalization, Productivity, Social, Sports, Tools, Travel & Local, Casual, Racing, Sports Games

G.2 Responses

11/21/2012 5:53:03				
11/21/2012 17:17:31	The categories it puts them in are really helpful in finding what your looking for	Sometimes it repeats the same apps throughout all categories	Search apps that were just realeased to the market	Productivity, Social, Sports, Tools, Arcade & Action, Brain & Puzzle, Sports Games
11/22/2012 3:02:25				Books and Reference, Personalization, Productivity, Social, Tools
11/22/2012 11:15:47	Finding apps I don't know I need.	Not really good for finding a specific app.	Ability to search for a specific app and to get recommended apps based on that.	Business, Entertainment, Finance, Health & Fitness, Lifestyle, News & Magazines, Personalization, Photography, Productivity, Sports, Transportation, Travel & Local, Weather
11/22/2012 16:49:45				Communication, Entertainment, Personalization, Photography, Productivity, Social, Sports
11/22/2012 19:10:09				Entertainment, Photography, Sports, Tools, Arcade & Action, Sports Games
11/22/2012 20:19:16	Intuitive interface, found good stuff that I liked, polished.	Nothing	Tracking of my installed apps to improve suggestions.	Books and Reference, Comics, Communication, Entertainment, Lifestyle, Media & Video, Music & Audio, News & Magazines, Personalization, Photography, Productivity, Tools, Transportation, Travel & Local, Weather, Arcade & Action, Brain & Puzzle, Casual, Racing
11/22/2012 20:22:31	The sliders!	N/A		Business, Communication, Education, Media & Video, Medical, Music & Audio, News & Magazines, Personalization, Photography, Productivity, Shopping, Tools, Transportation, Weather, Libraries & Demo, Casual
11/23/2012 0:52:28				Communication, Education, Entertainment, Finance, Music & Audio, Photography, Productivity, Sports, Tools, Weather, Casual, Sports Games
11/23/2012 1:33:26				Entertainment, Music & Audio, Productivity, Shopping, Tools, Arcade & Action, Brain & Puzzle, Cards & Casino, Casual
11/23/2012 2:34:49	Its nice. Its fresh. Its fun!	None off the top of my head		Entertainment, Lifestyle, Media & Video, Music & Audio, News & Magazines, Personalization, Photography, Productivity, Tools, Weather
11/23/2012 17:02:52				
11/23/2012 19:38:13	The user interface is very good, easy to use I have discovered alot of good apps from appdj	None!		Books and Reference, Communication, Entertainment, Lifestyle, Music & Audio, Personalization, Photography, Productivity, Social, Tools, Arcade & Action, Brain & Puzzle, Cards & Casino, Casual, Racing, Sports Games
11/23/2012 19:51:53				Communication, Entertainment, Lifestyle, Media & Video, Music & Audio, Personalization, Sports, Tools, Transportation, Weather, Arcade & Action, Sports Games
11/24/2012 0:14:30				
11/24/2012 3:27:17	discovering new apps which we generally don't get to explore in actual play store	Redirecting page to app store Wic takes time	will let u know if any thing is needed	Communication, Education, Entertainment, Finance, Health & Fitness, Lifestyle, Media & Video, Music & Audio, News & Magazines, Personalization, Photography, Productivity, Social, Tools, Travel & Local, Weather, Arcade & Action, Brain & Puzzle, Casual, Racing, Sports Games
11/24/2012 7:34:43	its interesting	See things in other languages i cant use.	filter by language	Books and Reference, Business, Entertainment, Productivity, Weather, Brain & Puzzle
11/24/2012 10:35:14				Comics, Entertainment, Media & Video, Music & Audio, News & Magazines, Personalization, Sports, Tools, Arcade & Action, Brain & Puzzle, Cards & Casino, Casual, Racing, Sports Games
11/24/2012 11:23:34				Personalization
11/24/2012 15:13:48	Very quick and easy way to tweak search criteria and get completely different results.	Not obvious why a given app is listed for a given set of criteria in quantifiable terms.	Gradation values for the sliders, even if arbitrary.	Communication, Music & Audio, Personalization, Productivity, Social, Tools, Arcade & Action, Casual
11/25/2012 19:07:48				Business, Communication, Media & Video, News & Magazines

11/26/2012 10:23:21	the interface till now	i dont have time till now to say what i dislike	filter to what is suitable for the device used to navigate AppdDJ	Books and Reference, Business, Education, Entertainment, Health & Fitness, Media & Video, Music & Audio, News & Magazines, Personalization, Productivity, Social, Sports, Tools, Arcade & Action, Brain & Puzzle, Racing
11/26/2012 17:26	The sliding catagories let me search for new apps without the trouble of putting the right keywords toedther. Iv managed to find far more aweso.e apps that suit my needs with appdj than just typing into google or some such app market.	The only negatives i can think of are superficial, in that i like a different kind of UI. But theres nothing really wrong. Just my personal preference.	App installation directly from appdj, without the redirect to the play store.	Books and Reference, Comics, Education, Music & Audio, Personalization, Productivity, Tools, Brain & Puzzle
11/27/2012 16:31				Personalization, Productivity, Social, Tools
11/27/2012 16:43		Only thing i dislike is not enough apps. It's a nice app and different		Entertainment, Media & Video, Music & Audio, News & Magazines, Personalization, Photography, Productivity, Shopping, Tools, Travel & Local, Weather, Arcade & Action, Cards & Casino, Sports Games
11/27/2012 18:45				Tools, Arcade & Action, Brain & Puzzle, Casual
11/27/2012 22:37	It's much simpler than browsing through lots of categories and lists.	You don't know what to really expect. But I have to say it gives consistent likable results.	You can make an option to browse through apps through categories and lists for users that may want to find very specific apps. That way the user would use this as an all-in-one app.	Business, Communication, Entertainment, Lifestyle, Music & Audio, News & Magazines, Photography, Productivity, Social, Sports, Tools
11/28/2012 2:19	It shows me a wide selection of apps.	It's good overall!		Books and Reference, Business, Communication, Entertainment, Health & Fitness, Lifestyle, Media & Video, Medical, Music & Audio, News & Magazines, Personalization, Productivity, Shopping, Social, Tools, Travel & Local, Weather, Arcade & Action, Brain & Puzzle, Cards & Casino, Casual
11/29/2012 4:17	Simplicity, surprises that il wasn't aware of without being random.	Would like a bit more granularity in refining search criteria.	More sliders for categories of apps.	Books and Reference, Business, Communication, Entertainment, Media & Video, Music & Audio, News & Magazines, Productivity, Tools, Arcade & Action
11/29/2012 21:55				
11/29/2012 22:07				Business, Comics, Communication, Entertainment, News & Magazines, Personalization, Photography, Productivity, Tools, Travel & Local, Arcade & Action, Brain & Puzzle, Casual
12/1/2012 0:03	I like being able to sort by category.	Not enough choices in categories.	Display app ratings.	Business, Entertainment, Finance, Health & Fitness, News & Magazines, Personalization, Productivity, Tools
12/1/2012 0:07	That you can set the criteria you want the search to be limited to	I am not sure how the sliders work... Some extra explanation would be appreciated	Hmmm...	Business, Communication, Entertainment, Personalization, Productivity, Tools, Transportation, Arcade & Action, Brain & Puzzle, Racing
12/1/2012 3:11	Different apps in one viewing of interest	no rating on apps unless u go to play store	rating and date app came out	Books and Reference, Education, Health & Fitness, Music & Audio, News & Magazines, Productivity, Shopping, Social, Tools, Weather, Arcade & Action, Brain & Puzzle, Cards & Casino
12/2/2012 15:28	It gives me recommendations based on things I deem are important.			Books and Reference, Education, Entertainment, Health & Fitness, Lifestyle, Media & Video, Music & Audio, Personalization, Productivity, Social, Tools, Libraries & Demo, Casual

H App Adoption Cycle

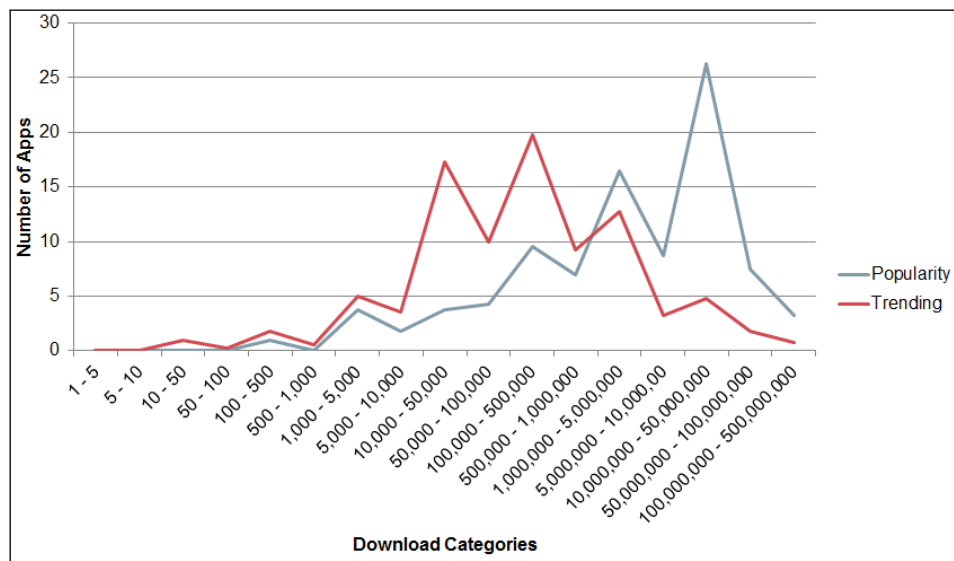


Figure 49: The average number of apps per download category in the popularity and trending list

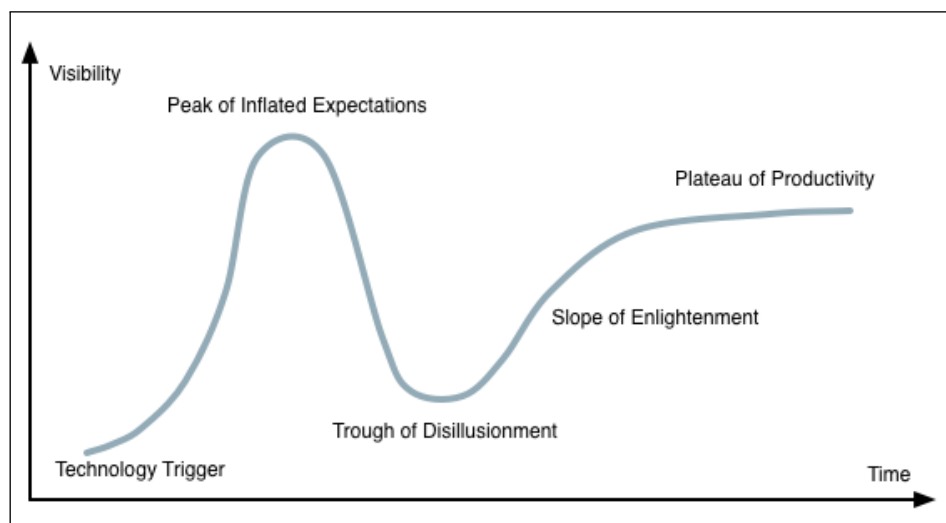


Figure 50: The Gartner Hype Cycle