Guerrilla Productivity: 
Gamification and Design-Related Touch Interfaces

Peter Schiller

A thesis
submitted in partial fulfillment of the
requirements for the degree of

Master of Science in Architecture

University of Washington
2014

Committee:
Rob Corser
Mehlika Inanici

Program Authorized to Offer Degree:
Architecture
© Copyright 2014
Peter Schiller
Guerrilla Productivity: Gamification and Design-Related Touch Interfaces

Peter Schiller

Chair of the Supervisory Committee:
Associate Professor Rob Corser
Architecture

In recent years, applications for touch screen devices have experienced tremendous growth and distribution. While some of the most widely-adopted applications offer game-based drawing experiences, tools designed specifically for digital drawing and sketching have experienced only modest popularity. Drawing games (and “gamified” experiences) have the ability to captivate users in ways that productivity-focused applications and tools cannot.

This thesis describes a history of touch interfaces for designers, including a more thorough analysis of a few contemporary examples. It then identifies strengths and weaknesses in current drawing applications, core principles to abide by when creating new applications, as well as unfulfilled market opportunities. Ultimately, these findings inform the development of a prototype application that serves to demonstrate how principles of gamification and play might be leveraged for a sort of guerrilla productivity — that is, user experiences that possess dual-citizenship within the realms of work and play.
# TABLE OF CONTENTS

1. INTRODUCTION .......................................................................................................................................................... 3

2. HISTORICAL OVERVIEW OF RESEARCH REGARDING PHYSICAL INTERFACES .............................................. 4

3. HISTORICAL OVERVIEW OF RESEARCH REGARDING TOUCH SURFACES ...................................................... 7
   3.1. TABLE-BASED AND WALL-MOUNTED SYSTEMS .............................................................................................. 8
   3.2. MOBILE DEVICES ............................................................................................................................................... 10

4. GAMIFICATION AND SUCCESSFUL MOBILE APPLICATIONS .................................................................................. 13

5. DESIGN TENETS ................................................................................................................................ ...................... 18
   5.1. SIMPLICITY ......................................................................................................................................................... 18
   5.2. UNRESTRICTING UI ......................................................................................................................................... 19
   5.3. SOCIAL INTERACTION ..................................................................................................................................... 20
   5.4. BRANDED AND SHAREABLE RESULT ........................................................................................................... 21
   5.5. INVITE PLAYFULNESS; DON'T REQUIRE IT ...................................................................................................... 22
   5.6. PRIORITIZE USEFULNESS ............................................................................................................................ 22

6. ANDROID PROTOTYPE APPLICATION ...................................................................................................................... 23
   6.1. SKETCHING AS COMMUNICATION .................................................................................................................. 26
   6.2. THE ARCHETYPAL NAPKIN .......................................................................................................................... 26

7. HYPOTHETICAL USE CASES .................................................................................................................................... 27
   7.1. PLAYFUL USE CASE ......................................................................................................................................... 27
   7.2. ANNOTATIVE USE CASE ................................................................................................................................ 28
### GUERRILLA PRODUCTIVITY

7.3. DESIGN DEVELOPMENT USE CASE .......................................................... 29

7.4. FIELD REPORT USE CASE ........................................................................ 31

8. PROTOTYPE OBSERVATIONS AND FUTURE WORK ....................................... 32

8.1. DRAWING INTERFACE .............................................................................. 33

8.2. COLOR SELECTION .................................................................................. 33

8.3. NUMBER OF PARTICIPANTS ...................................................................... 35

8.4. LAYERING, ORDERING AND IMPORTING .................................................. 37

8.5. ACHIEVEMENTS ...................................................................................... 38

8.6. RICH MEDIA ............................................................................................. 39

8.7. COMPANION APPLICATIONS ................................................................... 40

8.8. SHARING EXPERIENCE .......................................................................... 41

8.9. ADDITIONAL SCREENS TO CONSIDER .................................................. 42

9. CONCLUSIONS ............................................................................................ 45

### APPENDICES

APPENDIX A: APPLICATION URLS
1. INTRODUCTION

Touch interfaces for designers have been investigated by many researchers over several decades. A review of existing literature indicates that they fall into two primary categories: physical interfaces and touch surfaces. While both categories are described in detail below, touch surfaces seem to be a more active area of research. This is due, in part, to the ubiquity of mobile devices that has emerged in recent years.

In May of 2010, just four percent of American adults owned tablet computers. Three years later, in May of 2013, that number had risen to 34%. As of September of 2013, 56% of all Americans owned smartphones (Brenner, 2013). This increase of mobile device adoption – and the subsequent increase of mobile application consumption – indicates an auspicious outlook for the development of design-oriented mobile applications.

Many of the most popular mobile applications can be categorized as games or “gamified” experiences. Gamification, described in chapter 4 of this document, refers to “the use of video game elements in non-gaming systems to improve user experience and user engagement” (Deterding et al, 2011a). The incorporation of gamification in design-related applications has the potential to encourage more widespread usage of such applications by the design community.

Before discussing the implications of gamification, however, it is important to understand a historical context of touch interaction as it relates to the tasks of designers. The vast majority of touch-based interaction (in fact, everything covered in this document) involves the manipulation of a graphical user interface (GUI). The manipulation can be indirect, as with
physical touch interfaces, or direct, as with touch surfaces. While the history of the graphical user interface is not the focus of this document, one cannot understand the fundamentals of touch-based interaction without a cursory understanding of GUIs. Readers who are unfamiliar with this topic are encouraged to explore it independently before proceeding.

2. HISTORICAL OVERVIEW OF RESEARCH REGARDING PHYSICAL INTERFACES

Physical interfaces, also known as haptic interfaces, allow for the manipulation of computer data (e.g., the geometry within a CAD model) via interaction with a physical artifact. These physical artifacts are sometimes known as “physical widgets” or “phidgets.” Greenberg and Fitchett first described phidgets in 2001:

Physical widgets or phidgets are to physical user interfaces what widgets are to graphical user interfaces. Similar to widgets, phidgets abstract and package input and output devices: they hide implementation and construction details, they expose functionality through a well-defined API, and they have an (optional) on-screen interactive interface for displaying and controlling device state. Unlike widgets, phidgets also require: a connection manager to track how devices appear on-line; a way to link a software phidget with its physical counterpart; and a simulation mode to allow the programmer to develop, debug and test a physical interface even when no physical device is present.
However, the emergence of haptic interfaces predates the coining of the term “phidget.” This chapter covers a limited set of contemporary examples which support design and design-related activities.

In 1998, Thomas Massie described “a tangible goal for 3D modeling.” He noted the shortcomings of the mouse as an interface device for 3D graphics, stating that it had become “obvious that 3D computer graphics could achieve much more with a more intuitive user interface mechanism.” Specifically, Massie focused on manipulating a digital model (virtual clay, as he called it) with a “phantom” haptic interface, whereby users placed their fingers within a thimble or around a stylus, which received specific vibratory forces to create the illusion of interaction with a physical object. In his research, Massie also discussed the rendering techniques required for a sufficient level of interactivity with the system.

Two years later, in 2000, Gregory, Ehmann and Lin created inTouch: Interactive Multiresolution Modeling and 3D Painting with a Haptic Interface. inTouch coupled visual display with force feedback, and could be used to “sculpt” and “paint” models in virtual space.

In 2001, Lin worked with a different team of researchers to develop ArtNova. ArtNova expanded upon the framework of inTouch and presented users with the opportunity to “interactively [apply] textures onto 3D surfaces directly by brush strokes, with the orientation of the texture determined by the stroke.” ArtNova also incorporated rendering functionality that allowed for haptic manipulation of the model’s visual presentation on the screen. In particular, the system incorporated “dynamically adjusted viewing,” which automatically repositioned the user’s model in the viewport and didn’t require switching between “haptic editing” and “camera repositioning” (Foskey, Otaduy, and Lin, 2002).
Liu, Dodds, McCartney and Hinds, in 2003, published their research on *Virtual DesignWorks*. The publication described what they perceived as a shortcoming of then-current haptic CAD systems: haptic feedback was used primarily for verification rather than interaction with virtual models. The researchers cited, as a reason for this, a difference in geometric representation between CAD models and haptic models. While employing a stylus similar to Massie’s, *Virtual DesignWorks* allowed users to interact directly with three dimensional geometry in B-rep models with operations such as push, pull and drag.

Contrary to the shortcomings presented by Liu et al., the *CUBIK* interface (Lertsithichai and Seegmiller, 2002) was developed specifically for haptic interaction as opposed to verification. The system consisted of a cube in physical space and a corresponding cube in virtual space. *CUBIK* was “bi-directional,” so any manipulation to either cube would be reflected by its counterpart. The implication was that designers could interact directly with physical models while digital representations of these models would be generated and updated automatically.

Finally, in 2008, Smith, Thomas and Piekarski developed *Digital Foam*, an interface similar to *CUBIK* that allowed users to manipulate sheets and spheres of foam (imbued with pressure sensors) in the real world in order to apply changes to geometry in the computer. The supported geometry was limited by the relatively low-resolution foam interfaces, but the project demonstrated another concept of “sculpting” in physical space to affect change in virtual space.
3. HISTORICAL OVERVIEW OF RESEARCH REGARDING TOUCH SURFACES

The category of touch surfaces (also known as multi-touch surfaces when they can recognize gestures using more than one finger) can be further divided into two distinct subcategories: portable and non-portable. Portable devices (or mobile devices) include modern tablet computers and smartphones; non-portable devices include table-based and wall-mounted interfaces. Though table-based touch interfaces have been actively researched in recent years, portable touch interfaces hold significantly more promise as avenues for future inquiry and innovation.

Mobile devices have only recently been able to accommodate the computational power designers require, so much of the extant research in multi-touch CAD technology has been conducted on table-based interfaces. While inferior in many ways to newer, more portable interfaces, table-based interfaces allow for a different sort of collaboration than current mobile devices. Mobile devices present users with easy opportunities for sharing and sending content; however, table-based interfaces essentially force participants to gather and collaborate in real-time. That is, with a single table-based system and multiple users, interaction between users is almost unavoidable. Thus, much of the design research pertaining to such interfaces focuses on collaboration.

Sijie Wang (2010) examined users’ effectiveness and preference when solving spatial problems (specifically, a jigsaw puzzle) using both tangible and table-based interfaces. Her research outlines the advantages and disadvantages of each representation (physical or digital) of the puzzle. Her conclusions indicate that most users found it easier to use the tangible
puzzle but more enjoyable to use the touch interface. It is somewhat remarkable that users derived more enjoyment from the system they found more difficult to use. Whether because of the “novelty” of touch interfaces or some other factor, this distinction indicates a compelling opportunity: improvement of multi-touch interfaces could lead richer, more immersive, more delightful user experiences.

There are several different technologies that support multi-touch interaction: resistance-based, capacitance-based, surface wave touch surfaces, and optical-based touch surfaces (Schöning et al, 2008). Optical methods, which cannot practically be used in portable applications, were the only way multi-touch sensing could be achieved before Jeff Han presented alternative techniques at a TED conference in 2006 (Fried, 2010). It was Han’s work that initiated obsolesce of table-based interfaces and the multi-touch revolution for mobile devices.

3.1. TABLE-BASED AND WALL-MOUNTED SYSTEMS

In 2003, Scott, Grant and Mandryk published System Guidelines for Co-Located, Collaborative Work on a Tabletop Display. The publication catalogs some early examples of table-based touch displays, including DiamondTouch (Dietz and Leigh, 2001).

DiamondTouch was an optical, multi-touch system that used a projector to display images on a table. The system had an array of small radio frequency transmitters under its tabletop. When a user interacted with the surface of the table, a circuit was completed between the user, the table, and the user’s chair (which contained a special receiver). Consequently, DiamondTouch could differentiate between users of the system.
In 2006, Wigdor, Shen, Forlines and Balakrishnan identified design requirements for “table-centric interactive spaces.” Their intent was to facilitate real-time collaboration and decision making by leveraging the innate characteristics of table-based touch interfaces. The authors express the utility of touch-based interfaces, but also the lack of knowledge regarding how to successfully incorporate them into spaces that could be used effectively and consistently. In hindsight, this line of inquiry (along with Han’s advancement of multi-touch technology around the same time), indicated the push towards mobile devices as an alternative to stationary interfaces.

Additionally, in 2006 (at a conference aptly titled TableTop 2006), Ryall et al. presented Experiences With and Observations of Direct-Touch Tabletops. The researchers described tabletop interfaces as “still in [their] infancy” and “not yet well understood,” even though DiamondTouch had been released five years earlier. In particular, they focused on tabletop interfaces “in the wild,” rather than “controlled user studies.”

The following year, Katz, Gabayan and Hamid demonstrated a camera-based multi-touch system with basic capabilities and a finger painting program. Also in 2007, Lin (not to be confused with M. C. Lin, mentioned previously) and Chang published A Camera-Based Multi-touch Interface Builder for Designers. The premise of their research was that multi-touch technology was burdened by unrecognized potential, and they sought to develop a toolkit for the generation of better multi-touch interfaces.

Finally, in 2009, Wigdor et al, created WeSpace, a “collaborative work space that integrate[d] a large data wall with a multi-user multi-touch table.” The system allowed users to walk up with their own personal laptops and devices and share content with one another.
GUERRILLA PRODUCTIVITY

WeSpace, like the projects before it, exemplifies the difficulty encountered when attempting to encourage collaboration organically and consistently on table-based touch interfaces. Today, as research begins to move away from large, stationary touch surfaces, the vast majority of touch interaction occurs on mobile devices.

3.2. MOBILE DEVICES

Contemporary examples of mobile devices include Apple’s iPad, Microsoft’s Surface, and many variations of Android-powered tablets, as well as countless smartphones powered by similar operating systems.

Tablet computers, when used on table tops, are functionally indistinguishable from table-based interfaces (apart from their screen dimensions). Consequently, most of the lessons learned from research on table-based touch interfaces can be easily applied to portable interfaces as well. Portable interfaces embody additional challenges and design considerations, such as providing users with a level of functionality sufficient for the completion of design tasks “on the go.” Whereas table-based interfaces are most often employed in offices and workspaces, mobile applications must be robust enough to be used in any environment.

As today’s designers rarely work alone, it seems prudent to incorporate collaborative features into design-related applications for mobile devices. Many existing applications incorporate collaboration in some manner, e.g., by allowing users to accomplish interpersonal tasks more efficiently. Other apps provide interconnectivity in different ways; for instance, by aggregating and distilling ideas for individual consumption. However, while many design-
related applications exist, few have seen widespread adoption among design professionals. A sampling of relatively popular design-related applications can be categorized as follows:

- **Model Viewers**, which permit the viewing and basic manipulation of 3D models. Examples include *iRhino 3D* and *CadFaster Collaborate*.
- **Site Aids**, which have features such as geo-tagging of photos and may incorporate GIS. For instance, *Avenza*.
- **Logistical Support** apps such as *PlanGrid*, which incorporate document management, issue tracking and more.
- **Portfolio Apps** like *Morpholio* and *Archtizer for iPad*, which allow designers to transport and present digital presentations of their work.
- **Sketching Apps**, which include *AutoCAD 360*, *Moleskine Journal* and *Paper by FiftyThree*, among many others.
- **Measurement/Planning Aids**, which offer features like unit conversion and assisted floor plan generation. Examples include *Architect’s Formulator*, *Architect Design Estimating*, and *inchCALC*. *Design Dimensions* is an app that lists the approximate sizes of common furniture and appliances. A more impressive example, *MagicPlan* allows a designer to simply photograph a space and automatically generate a dimensioned floor plan.

---

1 (Ultimate iPad Guide..., Top Android Apps...). A list of URLs for the applications in this section can be found in Appendix A.
- **Idea Databases**, which are similar to architectural magazines in that they simply offer users ideas and design inspiration. Examples include Houzz and Dwell.

“Best practices” and design guidelines specific to CAD on mobile devices are somewhat limited, but the topic has been previously explored. In 2012, Radhakrishnan compared standard mouse interaction in CAD systems to touch interaction, with the following goals:

1. Outline the key elements of the multi-touch interface for CAD
2. Identify the factors affecting the performance of a multi-touch enabled CAD modeling environment
3. Lay a foundation for future research and highlight the directions for extending the multi-touch interface for CAD and other engineering applications.

Radhakrishnan concluded that selection and manipulation of 3D geometry suffered higher error rates on multi-touch systems than on traditional mouse-and-keyboard systems. Indeed, many mobile applications for designers exist today, but none offer full drafting capabilities comparable to those found on desktop computers. Through further investigation, it might be concluded that touch interfaces are simply inferior to traditional computers for technically-demanding use cases like drafting. Other design tasks might be incorporated into mobile devices more successfully, and application developers might achieve better success by solving problems appropriate to touch interfaces.
4. GAMIFICATION AND SUCCESSFUL MOBILE APPLICATIONS

One of the most popular applications listed above is *Paper* by FiftyThree. *Paper* was the brainchild of developers from Microsoft’s *Courier*, a dual-screen, “booklet” style tablet computer that was never released (Epstein, 2012). The iPad sketching application offers several different virtual “media” and interaction paradigms to simulate sketching, drawing, inking, coloring, and other activities. Recently, FiftyThree released a Bluetooth stylus to accompany *Paper*, which they appropriately call *Pencil*. *Pencil* enhances interaction with *Paper* by incorporating new gestures and features such as finger smudging. *Pencil* can also be turned around and used as an eraser (Hamburger, 2013). According to the developers’ website, the *Paper* application has been downloaded over eight million times as of June, 2013 (FiftyThree News).

The term “gamification” was first used in the early 2000s. It can refer to applications that take inspiration from video games, applications that are themselves games, or even applications that simply incorporate “playful design” (Deterding et al, 2011b). “Gamified” experiences may include:

- Points or scoring mechanisms that allow users to measure their progress towards accomplishing a particular task. Points also allow users to measure their progress or skill level against other users. In games with high score or leaderboard mechanisms, users are able to measure their progress against the progress of strangers (perhaps even strangers all over the world, in the case of games with internet connectivity).
• Virtual awards – known as achievements – that can be “unlocked” by accomplishing tasks that are secondary to the primary goal or storyline. For example, a shooting-based game may present a user with an achievement if the user maintains a certain level of accuracy while completing a level. Achievements may be hidden from the user until tasks are accomplished, providing an element of surprise or delight. Alternatively, achievements may be presented as challenges to the user (e.g., challenging the user to maintain a certain level of accuracy in a shooting game).

• Levels, or discrete tasks that make up “chapters” of a story. Levels can be used to present a narrative to the user, and levels – much like points and achievements – allow the user to define and accomplish clear goals.

All of the gamification strategies listed above serve to keep the user engaged with the system. User engagement means enthusiastic or prolonged use of the product, which can lead to higher sales in a variety of pricing structures. These pricing structures may include pay-for-play models, in-game purchases, or other strategies; however, pricing structures are beyond the scope of this thesis.

*Paper* isn't particularly gamified, but it provides the user with a novel and engaging experience. *Paper* attempts to bring the best aspects of physical artistic media to mobile devices, and the product succeeds among users seeking a digital, mobile, touch-enabled equivalent to traditional means of artistic expression.
In terms of popularity, however, *Paper*’s success is not nearly as significant as that of *Draw Something*, a drawing game for Apple and Android mobile devices. Within 50 days of launching on the iPhone, *Draw Something* had passed 50 million downloads in the Apple App Store. As of April 2012, users were creating more than 3000 drawings per second with the application (F. A., 2012).

In *Draw Something*, users play collaboratively with each other: a user is presented with a word and tasked with depicting the word in a drawing. Another user then attempts to identify to the word by viewing the drawing. For instance, User A might be assigned the word “golf,” and attempt to draw a golf ball. User B would then receive the drawing of the golf ball and try to guess the correct word for the picture. User B is presented with a fixed number of letters – some of which are in the word and some of which are not – and a number of blank spaces (akin to the popular game *Hangman*). Both users are awarded virtual “coins” – which can be used to unlock new drawing colors – when a word is identified correctly. Once User B correctly identifies the drawing, User B attempts to draw a picture for User A to identify (and so the game continues). The interface is quite rudimentary; users can select from a few colors and draw with their fingers. Additionally, users can toggle between three brush sizes. In terms of artistic expression, *Paper* clearly offers more capability.

*Draw Something*’s success can be attributed to its categorization as a game. The task is trivial and fun. The drawings are ephemeral, but still engaging. And even though *Draw Something* isn’t a tool for designers, it certainly encourages collaborative, creative exchanges between users. As a result of the game’s popularity, *Draw Something*’s parent company
OMGPOP was sold by its creators to a gaming company called Zynga for almost $200 million dollars (Streitfeld, 2012).

*Snapchat* is another touch-screen application in which users can draw with their fingers. The premise behind *Snapchat* is picture messages that disappear after a certain number of seconds. Users, especially teenagers, enjoy the app because they can send “weird, ugly or banal” pictures to each other without worrying about the pictures being reproduced or republished (Dredge, 2013).

Drawing on pictures is one of the few affordances of *Snapchat*’s interface. Even though it isn’t the central focus of the application, it certainly contributes to the application’s appeal. Users are also allowed to annotate their photographs with captions. These features exist in many design-related applications, but *Snapchat* is about fun rather than productivity. In fact, the automatic deletion of the photographs in *Snapchat* makes the application decidedly unproductive. Rather, *Snapchat* is evidence of gamification’s appeal. As of early 2014, while *AutoCAD 360* had been downloaded between five and ten million times on Android devices, *Snapchat* had been downloaded between ten and fifty million times (Android Apps). Reportedly, in 2013, *Snapchat*’s developers refused a three billion dollar buyout offer from the social networking site Facebook (Dredge, 2013a).

If the properties of *Snapchat* and *Draw Something* can be captured in drawing applications for designers, there exists the potential for wider application adoption and increased user satisfaction.

As further evidence for the merits of game-based experiences, in March of 2014, in the Apple App Store, the combined categories of Games and Entertainment were more than two
and a half times as popular as the combined categories of Business and Productivity (see Figure 4.a)².

Figure 4.a: Graph of popular app categories in Apple App Store, March 2014, by share of available applications

² Statista, 2014
5. DESIGN TENETS

A set of design tenets, or best practices, can be extrapolated from the precedents in chapters 2 through 4. These observations are presented as original research, and do not rely extensively on previously established claims. The incorporation of these principles can’t ensure the success of a mobile application; however, the exclusion of the principles will likely make success much more difficult to achieve.

5.1. SIMPLICITY

When it comes to the design of mobile applications, the adage “less is more” deserves serious consideration. For example, while Draw Something only provides the user with a set of rudimentary drawing affordances, the application has been much more widely adopted than a myriad of more robust drawing programs. One possible explanation for this is that users are averse to learning new things when they feel it’s unnecessary. In fact, some usability experts advocate that simplicity is the most important law of usability (Krug, 2006). An aggressively simple interface can be universally intuitive.

Another important consideration is the skill level (independent of technical savvy) of the user. Users who are unskilled in drawing are less likely to be intimidated by a simple set of tools; however, users who are comfortable with drawing should still be able to accomplish their goals with a limited interface. In addition, with this level playing field, the experience may be perceived as more fun or impressive when users create complex drawings. Interface restrictions could even contribute towards engaging sentiment, such as “how did you do that?”
or “let me try.” In applications like Draw Something, users of all skill levels are able to express themselves and compare their abilities with the application’s limited drawing capabilities.

Simple interfaces afford interface designers fewer opportunities to make mistakes; however, with each feature representing a larger portion of the interface, mistakes have the potential to be more severe in magnitude. Simplicity doesn’t necessarily mean reduced development cost or quicker development time. Great care must be taken when defining the set of features to be included in an application, and each feature must be designed to accomplish exactly what it must, but no more.

5.2. UNRESTRICTING UI

It’s almost a corollary to the first point, but the interface of the application shouldn’t pigeonhole the user into certain use cases. In Snapchat, users can either draw complex illustrations or write and send simple messages. Sometimes, a user may augment a photograph with a detailed sketch, while other uses cases require a simple annotation on the image. In one use case, a user might take a photo of a restaurant, annotate the picture with the restaurant’s name, and send it to a friend. The message would serve as a simple communication of what the user is currently doing. In another use case, a user might capture a photo of a friend, superimpose a humorous illustration onto the image, and share it to a social media site.

Snapchat doesn’t have a full set of message-based features, nor does it have a full set of drawing features. Its agnostic utility allows millions of users to sketch and send picture messages for a variety of purposes. In contrast, applications like Autodesk SketchBook and
Paper each have a number of advanced capabilities. The applications provide excellent interfaces for artists and designers to record their ideas digitally. Consequently, they neglect an entire market segment of casual users who simply want to interact with their friends.

5.3. SOCIAL INTERACTION

Any game with more than one player encompasses social interaction in its experience. Within the realm of applications designed for play, social interaction is equally prominent. Design, like most professions, happens in the context of other people and not in a vacuum. Applications that allow users to connect with one another benefit from an immersive experience that cannot be provided with a solitary user interface. In Q2 of 2103, 9 of the 10 most popular applications were messaging or social media applications. The only exception was Google Maps (Fox, 2013)

As demonstrated in Snapchat’s popularity, the excitement and anticipation of drawing something for a friend and waiting for their reaction can be more compelling than any virtual sketchbook, no matter how full-featured it may be. Furthermore, applications that feature social interaction prominently may benefit from a sort of feedback loop. Users spend time within the application and use it to connect with other users, who then engage with the application themselves. As they interact socially with even more users, a cycle of community engagement begins to emerge. Often, this manifests in the viral growth of newly created applications.
5.4. BRANDED AND SHAREABLE RESULT

Another prominent feature of successful mobile applications is the relationship between branding and sharing. Many people are familiar with what Snapchat images look like, even if they've never used the application. This is because of coverage in news stories, discussion on social media, and other means of exposure. Sharing Snapchat or Draw Something screenshots on social media sites allows users to show off their talent, particularly because other users can relate to the experience. By allowing users to share their work, the gamification principle of competition can be implied in the application, even if it isn't explicitly incorporated.

Recently, an application called Bitstrips became popular on sites like Facebook. In November of 2013, it was the second-most downloaded mobile app in the entire world. (Dredge, 2013b). It isn't a drawing application, but it does incorporate a certain aspect of creativity. In the application, users create cartoon by selecting backgrounds, avatars, props and captions. Bitstrips markets its application as “instant comics starring you and your friends” (Bitstrips).

The resulting images are immediately recognizable, but they don't explicitly say where they're from. No logos or branding marks are added to comics created with the software. Consequently, the application capitalizes on a sort of branded mystique. Facebook users want to discover what new thing their friends have found, and then many want to join in on the fun.
5.5. **INVITE PLAYFULNESS; DON’T REQUIRE IT**

The goal of *guerrilla productivity* is to create applications that can fit within the realms of both work and play. This dual-citizenship requires a careful balance. While aspects of work may be unwelcome in playful contexts, aspects of play can be downright detrimental in professional contexts. Consequently, an application should allow the user to decide what’s fun rather than presenting an explicitly frivolous experience.

One way this might be accomplished is with an overall levity of tone. An application can present tools for productivity in playful ways: affordances such as friendly dialog messages or delightful UI elements accomplish a lot in this regard. Applications for productivity should be designed for non-productive circumstances. For instance, instead of designing tools to be used inside of office environments, imagine a typical user opening the application while passing time at a cafe, or while commuting on public transit.

5.6. **PRIORITIZE USEFULNESS**

Finally, designers of mobile applications should be careful not to alienate professional users. If there is any chance that a new feature might hinder users who simply need to get work done, the feature should be reevaluated or removed. When demonstrating a harmony between playfulness and usefulness, application developers must be vigilant in preserving user productivity. Any feature that unnecessarily trivializes the experience should be omitted.
6. ANDROID PROTOTYPE APPLICATION

As a proof of concept of these ideas, this thesis includes a working Android prototype application. It is important to recognize that this prototype serves as just one example of how these design tenets might be embodied; it is not a single manifestation.

The prototype allows a user to take a photo with the device’s camera. In the case of devices with front and rear facing cameras, either camera may be used. After taking a photo, the user is presented with a drawing interface. From this screen, the user can sketch onto the photo with a finger or a stylus. The user also has the ability to add a caption to the image. Sketches can then be saved to individual “layers” or “frames.” Conceptually, this is similar to drawing on a pad of paper and tearing off the sheets (in practice, however, the drawing in the application is preserved after a new frame is created). After the user has finished creating new frames, the final image can be exported as an animated .GIF (Graphics Interchange Format).

The drawing interface has a few primary affordances (see Figure 6.a):

- A color picker\(^3\)
- An eraser, which can clear the screen when long-pressed
- A field to enter a caption
- A button to create new layers

\(^3\) Colors from http://flatuicolors.com/
After creating at least one frame, the user drags a seekbar at the top of the screen to transition between completed frames. Essentially, this serves as a preview of the animated .GIF. Each frame contains a sketched-on photo and a caption, and the user can delete an exported frame by long-pressing on it and confirming that it should be deleted. If the user drags the seekbar all the way to the right, a page is displayed with options to export the drawing. Conceptually, this means that the drawing interface is always “on the left,” and the sharing interface is always “on the right.” Between them exists a variable number of drawings (see Figure 6.a).
In an exported .GIF, the first frame might last longer than subsequent frames, and it could feature the app name and/or logo below the original photograph. In each frame, the caption text is placed where the UI for drawing was previously located, eliminating the need for an aspect ratio adjustment to fill the screen again. This presents an opportunity for a recognizable aesthetic, comparable to the universally recognizable photos taken by Polaroid instant cameras.

After this frame displays, the other frames each display for some amount of time (which could be adjusted, depending on the purpose of the animated image). Frames would continue to display until the animation was finished, at which point the .GIF would restart (loop). Examples of exports in both work and play scenarios can be seen in chapter 7.

There are several technical considerations to the application that lie beyond the scope of this document. Chiefly, the management of multiple large bitmaps in device memory requires careful creation and recycling of objects. Furthermore, the preservation of system state when switching between views requires a robust data model. Finally, asynchronous operations should be employed when saving images to the file system, as they can take quite a long time to export on high-resolution devices.
6.1. SKETCHING AS COMMUNICATION

One novel feature of the application that could contribute to a multiplayer experience is the preservation of the drawing layer. Imagine that User A takes a photograph and sketches onto it. User A then sends the sketched-on photograph to User B (who also has the application installed). When User B receives the sketched-on photograph, User B can modify User A’s sketch without defacing the photograph below it. Thus the drawing layer of the sketch/photograph pair is preserved in transit.

This feature provides tremendous potential in both professional and playful contexts. In the case of serious work, the editable drawing layer means that multiple users can use the application to develop and refine an idea. The application can be used as a method of communication between designers and clients, designers and managers, clients and builders, or anyone needing to communicate ideas visually.

In the case of playful users, the editable drawing could present a source of entertainment as the users exchange amusing permutations of the sketch. From a gamification perspective, this feature could enable either collaboration or competition. By enforcing no particular use case, it aligns with the design tenet of an unrestricting UI.

6.2. THE ARCHETYPAL NAPKIN

Within architectural tradition, there exists the canonical idea of the napkin sketch: the architect and the client are sitting at a table in a dimly-lit bar, when the architect, in a moment of epiphany, begins an impromptu sketch on a napkin. After some iteration and discussion, both parties leave the bar with a fully developed concept of what the building will look like.
The romantic notion of sketching on a napkin is really about documenting good ideas quickly. Today, in any given context, smart phones and tablets are probably more ubiquitous than napkins. Additionally, touch devices provide affordances that napkins do not. Photos can be sketched on, drawings can be revised, and ideas can be shared. The concepts of guerrilla productivity align with the motivation of the napkin sketch: being able to record and communicate ideas whenever the need arises.

7. HYPOTHETICAL USE CASES

This chapter details several hypothetical use cases of the application. The playful use case is inherently convincing, so more attention will be directed towards explaining the productive use cases.

7.1. PLAYFUL USE CASE

Figure 7.1.a demonstrates how the application might be used for a playful purpose. Here, photos of a dog have been modified for humorous effect. In this example, a single image is used for every frame of the animation. The use of a single photo creates an interesting effect, as each new photo alters the image, but doesn't replace it. A user could share the animated version of this image via social media or other means.
7.2. ANNOTATIVE USE CASE

Figure 7.2.a shows how the application might be used to annotate a photographic punch list at a job site. In this demonstration, each frame of the .GIF has a separate photo. This is not something the prototype currently allows, but the modification required to introduce such a feature is trivial.
7.3. DESIGN DEVELOPMENT USE CASE

Figure 7.3.a shows how the application might be used as a communication tool. Two hypothetical users take turns revising the redesign of an intersection. The photograph and some of the design ideas were taken from a Better Block project in Oklahoma City. \(^4\)

\(^4\) Lackmeyer, 2012
Figure 7.3.a: The application as a communication tool for the improvement of a crosswalk.
7.4. FIELD REPORT USE CASE

Figure 7.4.a demonstrates how the application has the potential to greatly increase productivity. On the left, a single image is annotated to show four (fictional) problems with a construction project: spray paint needs to be removed from the column, the concrete needs to be patched, the linoleum is scratched, and there are loose wires around the electrical outlets. The image is captioned with the location of the room in the building. If this image were to be sent to a contractor who had the application installed, the field report would be done.

On the right, a current way of creating a field report is demonstrated. The image is loaded into an e-mail application on the mobile device, and the user is required to type out a long message explaining what the problems are. This can be especially cumbersome, as users may find it difficult or tedious to compose long e-mail messages on small software keyboards. Furthermore, whereas sketched annotations leave little room for misunderstanding, there is less of a guarantee that the user’s textual descriptions of problems will be understood by the e-mail recipient.
Figure 7.4.a: A single image with multiple problems annotated

Figure 7.4.b: A photograph inside an e-mail with a textual description of problems

8. PROTOTYPE OBSERVATIONS AND FUTURE WORK

Many aspects of the prototype merit a deeper investigation. Below, several key interface decisions are examined individually. Because the prototype serves only to represent the larger idea of guerrilla productivity, the features described below should be analyzed as individual possibilities. Some of features might not be practical when combined with one another.
8.1. DRAWING INTERFACE

The first design tenet, simplicity, is a primary strategy of the drawing interface. While the interface is intentionally rudimentary, it might be improved with the addition of certain features. For example, the ability to undo/redo drawing actions might be more effective than the eraser when making revisions to a sketch.

Another consideration is the size of the brush in comparison to the size of the screen. Users on small devices might desire thinner line weights, but not line weights so thin that they're difficult to connect or easily obscured by fingers. Drawing with a stylus might allow thinner line weights to be used more effectively; for this reason, providing the ability to set the line weight could be beneficial to the experience. In the interest of UI simplicity, the user could be presented with a few discrete options rather than a sliding scale of values. Draw Something features line weight selection in this manner. The application might also recognize if the user is drawing with a stylus or a finger, and adjust the behavior of the drawing accordingly.

Finally, the UI of the prototype possesses an important and novel affordance: the area of the screen that holds the controls during the drawing experience becomes an area for the captions during the viewing experience. When images are shared, this affordance contributes a recognizable aesthetic to the application.

8.2. COLOR SELECTION

Deciding on how many (and which) colors to provide for drawing is an interesting challenge. A full RGB/HSB color picker (like the color pickers found in image editors such as Adobe Photoshop [see Figure 8.2.a]) would be too technical for the application's purpose.
Similar to the problem of line weight selection, presenting the user with too much granularity makes it difficult for the user to select the same value twice, or even select the appropriate value once.

![Color Picker (Foreground Color)](image)

Figure 8.2.a: Adobe Photoshop color picker

Given that the set of colors should be limited, how are the appropriate colors chosen? One approach is to provide the user with just enough basic colors to communicate ideas effectively: green for vegetation, blue for water, brown for wood, etc. However, this approach could lead to drawings with little aesthetic appeal, which would make the application seem incapable of satisfying users’ needs.
A more refined set of colors could contribute to the brandibility of the experience. Consider, for instance, a drawing interface that only offered shades of red and blue. While it may seem difficult to use these colors to draw a snowy landscape or a human face, the restriction has the potential to make the application seem playful. More importantly, drawings exported from the application and shared on social media will distinguish themselves from drawings created in other applications. Choosing a limited set of colors could allow the application to impose a visual identity on all user-created content. Notwithstanding, the color limitations should not hinder the functionality required by professional users.

Another possibility is to provide the user with different color palettes/schemes. Draw Something’s virtual coins can be used to purchase additional colors with which users can sketch. The coins themselves can be earned slowly by completing drawing challenges, or they can be purchased rapidly in-app (with real money). In the first case, they present users with an incentive to continue playing the game. The benefit to the developer of the in-app purchases requires no explanation.

### 8.3. NUMBER OF PARTICIPANTS

While the design development use case in Figure 7.3.a demonstrates a conversation between two participants, there’s not necessarily a reason to limit conversations to only two users. That stated, introducing more users into the conversation significantly alters the experience and presents organizational challenges as well.

For instance, with three users participating, how should messages (i.e. drawings) be ordered? Perhaps one user simply sends a drawing and other users in the conversation
receive it, adding it to their local copies of the conversation. This sort of “blind broadcast” strategy is relatively easy to implement, but there's no guarantee that all users will receive the messages in the same order. Furthermore, there's no mechanism for resolving conflicts if the order of messages is important in the conversation.

Notwithstanding, there's a certain serendipity to the idea of the blind broadcast system. In the play scenario, it may generate interesting and divergent routes for the conversation. In the work scenario, it may help to stimulate creativity between participants.

Having multiple participants also complicates the logic for who gets to draw and when. With two users, the interaction might simply be turned-based (e.g., A, B, A, B, etc.). With multiple users, turn-based interaction could be cumbersome and slow. This raises a few interesting questions:

- Does it matter if one user submits two drawings before other users can respond?
- Should there be a rate limit on the number of drawings that a user can send before receiving a response, or within a certain time period?
- Is it possible (or important) to encourage users to contribute equally?
- Should there be an indication of authorship on each drawing, or perhaps in the exported image?
- If complex data models – like an “undo history” – are stored locally, should these be transmitted along with the drawing/photo pair?
• Are interruptions relevant? An indication that another user is currently drawing (similar to the “_________ is typing...” notifications on many instant messaging clients) could be introduced as mechanism to prevent conflicts. It might also introduce a playful “race” component to conversations where more than one participant is drawing.

8.4. LAYERING, ORDERING AND IMPORTING

The prototype doesn’t currently allow for the layering of drawings – that is, users are not allowed to take two drawings and adjust the opacity so that both are visible. This feature could greatly enhance the viability of the application as an ideation tool, while also providing easier use in playful scenarios. With opacity adjustments and layering, the metaphor for the application becomes less like a pad of drawing paper and more like a roll of tracing paper.

Currently, the drawing is preserved within the drawing interface when users create new frames, so a user is able to create frames with incremental changes. However, if a user wants to create a frame based on older ideas (e.g., “I liked the idea we were discussing 30 minutes ago... let’s start from there and try again.”), the user would have to recreate the drawing manually. The ability to promote a drawing to the front of the “stack,” or back to the drawing interface, would provide a richer experience. Conceptually, this change would allow users to create a branching tree of drawings, rather than a linear progression of drawings.
The complexity of a layered or re-ordered experience would be greater, which could make the application more difficult to use. Thus, the addition of these features would require careful consideration.

Users might also appreciate the ability to upload an existing photograph rather than using the device’s camera. This would not increase the application’s complexity very much, but it would provide powerful improvements in both playful and productive contexts. Moreover, the ability to import an animated .GIF – perhaps one created within the application, perhaps not – is a feature worth examining.

8.5. ACHIEVEMENTS

An achievement is a virtual award or trophy that a user receives for accomplishing a certain goal. Achievement-earning goals can be hidden from the user, so that completing an achievement provides a delightful surprise. Alternatively, they can be presented as challenges to the user to keep the user engaged.

Example achievements:

- Sketch an entire .GIF with a single photograph
- Use a different photograph on each frame
- Share/create a number of sketches (e.g., 5, 10, 20, etc.)
- Use all available colors within a single .GIF
- Sketch with a single color per page

Example rewards:
• More colors
• Different color palettes
• Higher number of maximum frames allowed in a single .GIF

One potential benefit of an achievement/reward system is the slow introduction of new features. Users could, upon first downloading the application, be presented with a simple interface. As they use the application and unlock rewards, learning about more complicated features of the application could be made into an enjoyable experience (because the user has “earned” the feature, it might be perceived as less of a chore). The net result is an initial reduction in cognitive load, and an increased amount of application usage (and potentially, a better longevity or “replayability”).

However, forcing users to earn certain features has grave potential to alienate professional users. In professional contexts, users of the application have specific tasks in mind. Any barrier to accomplishing these tasks will likely frustrate the user and result in an unfavorable opinion of the application. One possibility to overcome this discrepancy is to have two mods of use – one for work and one for play – but this directly conflicts with the design tenet of an unrestricting interface with agnostic utility.

8.6. RICH MEDIA

An application with a simple or rudimentary interface can still benefit from the advanced features of today’s touch-enabled devices. One of these features is location data. Drawings created within the prototype could be tagged or manipulated based on their
locations. At the very least, the location of a drawing could be recorded for reference. Perhaps the application could behave differently if it understood where you were or what you were doing at the time.

The inclusion of sound is another feature that could provide enhanced productivity and play. Audio narrations would allow professional users to more accurately and easily record their thoughts as they sketched. The narrations could be made during the act of sketching, or while navigating between completed drawings. In a playful context, narrations could provide a much richer and more engaging experience to the user. Other sound effects could be employed as well.

The export itself could provide additional features, such as the ability to select the transition time between frames or add transition effects. These could be applied individually or globally, depending on implementation. Of course, this feature would add complexity to the interface, and might not provide a substantial return.

With the addition of sound and transitions, the application seems better suited by a video export format rather than an image export format. This change brings both positives and negatives: videos provide richer and more immersive experiences, but suffer from a more specialized compatibility than .GIF files, which can be used almost anywhere on today’s devices and websites.

8.7. COMPANION APPLICATIONS

One benefit of .GIF files is that many photo editors load the individual frames as separate layers. Thus, an animated .GIF exported from this application could be taken into a
photo editing program, modified on a per-frame basis, and saved as individual image files. This enables professional users to easily bring ideas from concept (in the application) to realization (on the computer). Imagine, for instance, an architect who takes photos of a site, sketches onto them, imports them into an image editor, and then uses the resulting files as the basis for a three dimensional CAD model.

The prototype could greatly benefit from one or more companion applications – desktop-based or mobile – that allow specialized operations to be performed on exported files. For instance, an application that takes photos from a field report and puts them into an official form would be especially useful for an architect. With companion applications, the interface of the primary application can remain simple and widely-applicable, while users who require specific tasks can also be satisfied. In fact, an individual user may only use the companion application a small percentage of the time, using the generalized interface in most circumstances.

A proprietary file format might enhance the power of a companion application. Albeit, a proprietary format wouldn't be as useful when shared, so it would probably have to be offered as a secondary option.

8.8. SHARING EXPERIENCE

Exporting a completed animation to the device’s storage is an obvious requirement, but the application should also allow easier sharing with other platforms and services. Foremost, sharing with social media services is easy to implement (most major websites have Application Programming Interfaces, or APIs, for such purposes). The application should also provide an
easy way to move finished sequences of drawings from the field to the office, via e-mail or some proprietary mechanism (perhaps one relying on a companion application, as described in section 8.7).

Another interesting feature would be a corresponding web interface, similar to many multimedia-based social networking services. Instagram (a social media site for photographs) and Vine (a social media site for videos) both provide web interfaces. A web interface for this application would certainly increase exposure – and thus, likely increase popularity – but it could also be used to further incorporate aspects of gamification. Leaderboards of most popular sketches could be publicized, and users could be allowed to interact with other users online. A web interface presents several technical and legal challenges that are beyond the scope of this document.

8.9. ADDITIONAL SCREENS TO CONSIDER

In addition to the user experiences investigated in the prototype, a commercial application would require the design of several other screens before its release. Many of these interface elements are detailed below, but this list should not be considered exhaustive:

- Contact list
  - This page displays a list of contacts with whom the user can create and share drawings. The contacts would also need to have the app installed, so some mechanism of verification would be necessary. This screen would also alert the user to new “drawing conversations” (or responses to existing conversations, potentially including indications of whether or
not a drawing has been viewed). Furthermore, this screen would require a UI affordance for adding new contacts to the list.

- Send action
  - After drawings have been created, they must be sent to other users to start or continue conversations. The user experience for this would have to be carefully designed, as it presents one of the best opportunities to encourage social interaction between users. The application must make it simple to send a drawing to another user... otherwise, users are less likely to share their drawings (and, given the importance of social interaction, less like to continue using the application).

- Export/share dialog
  - The prototype currently allows a user to download an animated .GIF of the drawing set. Downloading an animation does not end the drawing experience, and animations can be downloaded at any time during the drawing process. They can be thought of as traditional “save dialogs,” which only briefly interrupt the user’s workflow and create a copy (immutable, in this case) of the current drawing set.
  - This dialog would also need to present the user with the opportunity to share his or her exported drawing (e.g., to social networking sites or an e-mail application) or e-mail application. As described previously, this feature is paramount to the success of the application.
There are also screens which aren't required for core functionality, but encompass high priority concerns:

- **Loading screen**
  - What does the app show when it's loading? This is a question central to the branding and identity of the experience.

- **Registration/tutorial**
  - Ideally, the application's interface would be simple enough for a novice user to understand without help. However, a first-run tutorial could demonstrate key functions and concepts to new users.
  - Additionally, users would need to register accounts with the application before they could send each other drawings. This would require the creation of an internet database system, the details of which are beyond the scope of this thesis.

- **Find friends**
  - Users can't be expected to enter individual phone numbers or e-mail addresses into the application to find conversation participants. The application must provide an easy way for users to add their friends to a conversation. In addition to finding friends, this screen would present the best opportunity for users to *invite* friends to the application easily.
  - This is crucially important, because one of the biggest barriers to
developing a successful application is convincing people to install it in the first place.

9. CONCLUSIONS

The investigation of multi-touch affordances for designers is still in its infancy. However, because of the widespread adoption of mobile devices, there are enough statistics available to demonstrate which interaction paradigms work and which do not. Like many new technologies, the invention of portable, multi-touch devices was initially interpreted as a harbinger to the obsolescence of laptops and computers. Touch interfaces have merits and capabilities that traditional computers do not, but multi-touch interaction is not a panacea and traditional computers are certainly not obsolete.

Much work remains to remediate the difficulties users face when transitioning from traditional computer interfaces to modern touch-based interfaces. Future research and development should focus on solving the right problems. Authors of design-related applications might find success by reexamining the requirements of their software within the context of simpler, more game-like experiences.

As demonstrated by the swift adoption of mobile devices among American adults (and the impressive rates of application consumption), it is certainly possible to create captivating, marketable experiences for smartphones and tablet computers. Unfortunately, while there are many examples of this within the realm of games and entertainment, there are few within the realm of design-related applications. The design-related tools currently offered on touch devices present substantial opportunities for improvement.
Finding common ground between work and play – and learning from games – is paramount to the idea of *guerrilla productivity*. The design tenets described in this document provide clear indications of how the problem should be approached, and how developers might create applications that offer better utility and more delight in many disparate contexts.
WORKS CITED


<table>
<thead>
<tr>
<th>APPLICATION</th>
<th>URL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Architect's Formulator</td>
<td><a href="http://www.multieducator.net/formulator/architect.html">http://www.multieducator.net/formulator/architect.html</a></td>
</tr>
<tr>
<td>Architizer for iPad</td>
<td><a href="http://architizer.com/projects/architizer-for-ipad/">http://architizer.com/projects/architizer-for-ipad/</a></td>
</tr>
<tr>
<td>AutoCAD 360</td>
<td><a href="http://www.autodesk.com/products/autocad-360/overview">http://www.autodesk.com/products/autocad-360/overview</a></td>
</tr>
<tr>
<td>Avenza</td>
<td><a href="http://www.avenza.com/pdf-maps">http://www.avenza.com/pdf-maps</a></td>
</tr>
<tr>
<td>Design Dimensions</td>
<td><a href="http://designdimensionsapp.com/">http://designdimensionsapp.com/</a></td>
</tr>
<tr>
<td>Dwell</td>
<td><a href="http://www.dwell.com/dwell-mobile">http://www.dwell.com/dwell-mobile</a></td>
</tr>
<tr>
<td>Houzz</td>
<td><a href="http://www.houzz.com/mobileApps">http://www.houzz.com/mobileApps</a></td>
</tr>
<tr>
<td>inchCALC</td>
<td><a href="http://www.inchcalc.com/">http://www.inchcalc.com/</a></td>
</tr>
<tr>
<td>iRhino 3D</td>
<td><a href="http://www.rhino3d.com/ios">http://www.rhino3d.com/ios</a></td>
</tr>
<tr>
<td>Morpholio</td>
<td><a href="http://www.morpholioapps.com/morpholio/">http://www.morpholioapps.com/morpholio/</a></td>
</tr>
<tr>
<td>Paper by FiftyThree</td>
<td><a href="http://www.fiftythree.com/paper">http://www.fiftythree.com/paper</a></td>
</tr>
<tr>
<td>PlanGrid</td>
<td><a href="http://www.plangrid.com/">http://www.plangrid.com/</a></td>
</tr>
</tbody>
</table>