Figure 1: A version of Pokémon GO prototyped with Lake. Enabled by using a camera element that uses the live camera stream from the tester's device. When the tester flicks the Poké Ball image element, the designer wizards the ball's movement across the screen.

Lake: A Digital Wizard of Oz Prototyping Tool

Andrew Finke
Northwestern University
Evanston, IL, USA
finke@u.northwestern.edu

ABSTRACT

Mobile app designers aim to develop the best mobile software interfaces in the least amount of time, and rely on testing ideas with prototypes in lieu of building costly, fully functioning applications. Yet, designers cannot effectively prototype some complex app experiences, including augmented reality applications like Pokémon GO, because existing tools lack the needed features, or because prototyping in them is too time intensive to be feasible. To solve this problem, we introduce Lake, a mobile application prototyping tool that enables the creation of complex mobile applications with the same ease as paper prototyping. By leveraging the Wizard of Oz technique used in paper prototyping in our digital medium, we enable designers to prototype at the same low cost as paper, but at a much higher fidelity. Through a pilot study (N=6), we find that designers are able to gather organic in-context feedback from complex prototypes made with Lake.

KEYWORDS

mixed-fidelity prototyping; lo-fi prototyping; mobile apps

CHI'19 Extended Abstracts, May 4–9, 2019, Glasgow, Scotland UK

ACM ISBN 978-1-4503-5971-9/19/05.
https://doi.org/10.1145/3290607.3308455

Figure 2: A version of Lyft prototyped with Lake that uses the tester's GPS location to show drivers around them. The designer can simulate a car coming to pick up the rider by moving the car elements on their designer app.

ACM Reference Format:
Andrew Finke. 2019. Lake: A Digital Wizard of Oz Prototyping Tool. In *CHI Conference on Human Factors in Computing Systems Extended Abstracts (CHI'19 Extended Abstracts), May 4–9, 2019, Glasgow, Scotland UK*. ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3290607.3308455

INTRODUCTION

Mobile app designers can innovate more effectively by prototyping app experiences and testing them in realistic situations of use [4, 6, 7], but struggle to do so when prototyping forward-thinking, complex app experiences using existing tools. For example, designers prototyping a Pokémon GO app [2] would need a tool that supports testing augmented reality (AR) interactions, but existing tools fail to effectively prototype the desired interaction or are too time-intensive to use. Not being able to create low-cost prototypes reduces opportunities to learn and iterate, resulting in a costly and prohibitive design process that yields unintuitive interfaces.

Low-cost prototyping techniques consist of inflexible, pre-canned interactions that limit the scope of interactions a designer can test. A designer prototyping Pokémon GO with low-cost hotspot-based tools like Marvel [1] or paper prototyping cannot adequately test and learn about interactions like AR. Since AR interactions are focused on the world around the tester, pre-canned interactions, or paper prototypes in a lab, are unable to capture the environment the tester is in.

While higher-cost prototyping techniques enable testing more complex interactions than low-cost tools, more time is required to build prototypes using them, resulting in less iteration time for designers and ultimately the creation of unfriendly interfaces as iterating later in the development process becomes increasingly more costly. For example, to prototype Pokémon GO with high-cost tools like Origami [9] or Form [8], designers would be forced to spend a significant amount of time writing code to trigger events based on what the tester sees in the camera stream or where they are located instead of spending time testing the prototype with users.

These existing approaches are fundamentally limited because they force designers to build prototypes at the same level of fidelity as the output (e.g. paper prototypes require a paper medium, high-functioning prototypes requires code or complex patches). Instead, the core conceptual approach of this work is to *maintain two distinct representations of a prototype across modalities and be able to seamlessly translate between those representations for designers and testers.*

We build this approach into Lake, a mobile application prototyping system built for the iPad and iPhone, enabling the creation of prototypes that leverage native components, device sensors, and tester-generated data in a rapid mobile prototyping environment. Similar to paper prototyping, Lake's approach uses the Wizard of Oz prototyping technique where designers manipulate native iOS elements on the iPad application. Testers use a representation of the shared prototype on their device in the same way they would use any other app, while designers use a different representation

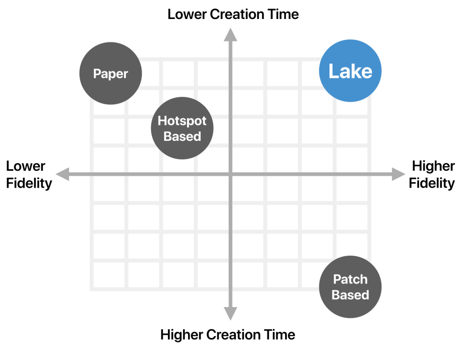

Figure 3: Paper prototyping, while low-cost, is limited to low-fidelity interactions. Hotspot-based tools, like Marvel, are also low-cost and slightly higher fidelity than paper, but not necessarily high enough to test certain interactions, like AR. Patch-based tools, like Origami and Form, offer the opportunity to create high fidelity prototypes, but at a significant cost. By translating across representations, Lake is able to offer designers a low-cost creation representation that translates to a high-fidelity representation for testers to use.

of the same prototype that allows wizarding the app's interactions. For example, to prototype an app like Lyft, the designer's representation would allow them to easily move car images around the map, while the tester's representation shows the car images as non-interactable map overlays that update in real-time. We hypothesize that designer's ability to do real-time Wizard of Oz using native components will enable designers to create low-cost prototypes in their representation that translate into a high-fidelity representation with complex functionality for the tester.

To evaluate the feedback designers can receive when prototyping apps in Lake, we conducted a pilot study with six designers. We collected data of the experience of using the prototype as a tester and the verbal feedback given from testers to designers. The results of the study show that Lake offers the opportunity to gather organic feedback from testers when using complex prototypes in-context.

RELATED WORK

Existing mobile prototyping tools force designers to make opposing tradeoffs between wanting to test complex functionality and needing to minimize construction cost of each prototype (Figure 3).

Patch-based approaches–where app logic is created by connecting feature "patches" together–used by Origami [9] and Form [8] allow designers to generate functional prototypes by writing less code than if they built out the software from scratch, but context-dependent app experiences still require significant effort. For example, one cannot easily prototype an interaction based on a user's location or on the image in the user's camera viewfinder without spending significant time programming complex app logic either with complicated patches or actually writing code.

Hotspot-based prototyping tools–where a user's actions trigger pre-specified presentation transitions in the application–such as Marvel [1] enable designers to create prototypes with less effort than patch-based tools, but with significantly less functionality. For example, a designer could prototype an app that allowed a tester to navigate through a menu, but little to no other complex functionality could be included, resulting in prototypes that can only respond to simple tester actions.

Paper prototyping heavily relies on the Wizard of Oz technique [5], where a tester interacts with a paper prototype and the designer "wizards" the intended response to the tester's actions by manually adjusting the prototype. While low cost and able to represent some features of a fully functioning prototype, paper prototyping is limited as it must be tested in a lab setting, and thus is difficult to test usage in other environments. As an extension, prior work in Remote Paper Prototype Testing (RPPT) [3] enables designers to remotely wizard paper prototypes, allowing for the cheap prototyping and in-context testing. However, the use of a low-fidelity paper representation still limits the interactions that designers can prototype and prevents testers from accessing the core functionality of the mobile device. For example, this setup cannot allow the user to take a photo or modify it in any way.

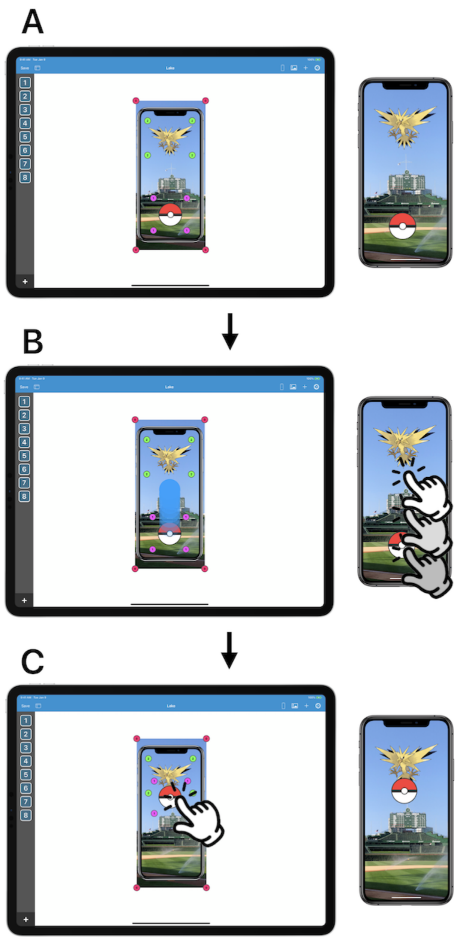

Figure 4: The designer begins by adding elements like a camera feed and images to the interface (A). Then, as the tester interacts with the prototype (B), the touch metadata is sent back to the designer and rendered on their representation of the prototype. When the designer sees the touch, they are able to wizard their representation of the prototype in response (C) and the tester's representation of the prototype reflects their changes.

SYSTEM DESCRIPTION

Lake is composed of the Lake Designer iPad app, for designers to create prototypes, and the Lake Tester iPhone and iPad app, for testers to run prototypes created by designers.

The Lake Designer iPad app allows designers to create and iterate on complex, functional prototypes with minimal cost by mimicking paper prototyping using native iOS interface elements. Designers start with a blank digital canvas, to which elements including as drawings, text, and map views can be added. Each of these elements can be panned, rotated, and annotated on through a stylus or touch input on the iPad, enabling rapid manipulation of elements. Additionally, dynamic properties can be configured for elements, such as adding a live filter to the camera.

Testers can test prototypes by opening the Lake Tester app on their device and entering the designer's unique pin code. Testers can be anywhere in the world with a network connection, enabling remote in-context testing regardless of the designer's location with no setup time. When testing a prototype, the tester uses the app as they would any other, through taps, gestures, and other interactions. This is done by showing the tester a representation of the designer's app that has been translated from the designer's representation into one with native, interactive components (Figure 4A). When a tester taps or swipes an element in the prototype, the system sends the tester's touch input back to the designer. The designer sees translucent blue dots briefly appear on the interface indicating touches (Figure 4B) and wizards the prototype's response to that interaction. When a designer updates the prototype, the tester's representation of the prototype reflects the change (Figure 4C). Designers can also passively view a tester's context data including location, physical activity (walking, running, driving, etc.), and step count to better understand how testers use prototypes in the real world.

PILOT STUDY

In our pilot study, we aimed to understand how mobile apps with complex, interactive experiences, such as AR, can be prototyped and tested using Lake. Specifically we aim to address the following research question: *can in-context testing of complex app interactions enable designers to gather organic feedback from testing sessions?*

To understand how designer and testers react to the system and to validate that designers could receive in-context feedback from testers using prototypes built with Lake, we conducted a pilot study with 1 undergraduate student and 5 graduate students. We recruited designers so that they would both be familiar with observing testers interacting with prototypes, and be able to articulate their experience while testing the application. Participants were paired off and brought into the lab for an hour long study session. Participants were compensated with a $25 Amazon gift card for their time.

As we were seeking to understand the feedback that could be generated by testers using complex app prototypes in-context, we gave designers two pre-built prototypes using Lake: an AR selfie spot

Figure 5: The AR selfie spot finder app prototype that enables users to navigate interior spaces with augmented reality to find the ideal spot to take a selfie. It was created in Lake by using camera, image, and text elements. The designer then wizards the direction arrow in response to tester movement.

Figure 6: The half-half photo capture app prototype that enables two people to blend their contexts into one photo. It was created using a camera element and an image element. The camera element is configured to allow testers to take pictures when they find the best photo. After the tester takes a picture, the designer, with the Lake Designer app, is able to display the captured photo later in the prototype.

finder (Figure 5) and a half-half photo capture app (Figure 6). We used these prototypes as they both featured complex interactions (AR and camera usage) that could not be prototyped at the same low-cost with existing tools, in addition to their focus on the user's environment (navigation and finding physical objects). Each prototype took less than 15 minutes to create with Lake. During a testing session, one designer would act as the tester using the first prototype for ten minutes and the other designer would observe their actions using the Lake Designer app as we wizarded the prototype. After the tester completed using the prototype, the designer was given ten minutes to ask questions to the tester about their experience to gather feedback about the prototype's features and usability. We then asked questions to each about their experience in a semi-structured interview, based on what each participant did during the prototype test. This procedure was then repeated with the second prototype with the designer and tester roles switched. We collected audio recordings of the participants and screen recordings from the Lake Designer app.

In our analysis, we looked at the transcripts from the audio recordings we collected. We identified feedback from testers that related both to the in-context nature of the prototype test and the complex functionality of the prototype interaction, such as feedback referencing a feature of the prototype to accomplish a task in the tester's environment. This would indicate that having a system that enabled complex prototype testing in-context yielded valuable organic feedback from testers.

Results - AR Selfie Spot Finder App

We saw that designers were able to get in-context organic feedback by looking at instances where testers mentioned both the AR navigation feature and how it related to their environment. Testers had concerns about hitting nearby objects: *"I feel like I'm going to bump into a wall"* (P1), informing designers that some users may be uncomfortable using AR navigation in small spaces. Testers were also unsure of when they reached their destination, *"I was expecting...that once I was starting to get closer to the destination either would change colors or something about the arrow would change so that I knew I was starting to reach it..."* (P5). By testing in-context, designers were able to learn that for AR based navigation, users needed a stronger indication of destination. Getting the same feedback could have been infeasible without a high-functioning prototype and realistic testing scenario.

Results - Half-Half Photo Capture App

We saw that designer's were able to receive in-context feedback regarding the experience of finding an object to take a picture of in the testers' environment, such as noticing testers often did not look at the app when searching for an object, *"I spent a lot less time looking at the screen... only using the phone when I actually needed to take the photo"* (P2), which let designers know that holding the device was inconvenient when looking for real world objects. Testers also thought it was difficult to align two objects with different physical shapes and all requested that the app provided on-screen alignment help, *"If something is helping me line [the objects] up, maybe I can take a much better photo...and that experience would be better"* (P6), and one tester suggested gamify the process, *"It would be good to have the camera... like a video game, have a target aligner with a box..."* (P4). Here, designers learned that users needed additional affordances to aid in capturing the real world object.

DISCUSSION AND FUTURE WORK

The results of our study show that Lake offers designers a way to test complex app interactions in-context that result in organic, substantial feedback. This feedback may have been impossible to obtain from low cost tools, due to the limited fidelity, or would have required a significant time investment in a high fidelity prototype tool, that may have required the use of pre-canned data, that also would not have resulted in the same feedback. Lake's ability to translate across representations allowed both prototypes used in the study to be created in less than 15 minutes, yet were translated to high fidelity prototypes for testers, allowing testers to provide real in-context feedback.

Future evaluations of Lake will explore more closely how Lake compares to existing prototyping tools. We will seek to better understand how Lake enables the creation of a larger range of apps that would have previously been too costly with existing tools. We also will evaluate the quality and quantity of feedback designers receive from testers using Lake versus existing tools.

Future work on Lake as a prototyping tool could explore enabling designers to prototype experiences with multiple participants, enabling prototyping video chat apps, multiplayer games, and other collective experiences across multiple groups of people.

ACKNOWLEDGMENTS

We thank Haoqi Zhang, Kapil Garg, Meg Grasse, Garrett Hedman, Sarah Lim, and the rest of the Design, Technology, and Research community for their support. We also thank Ian Baird, Jenny Chen, Grace Chin, Bill Dudney, Ben Fearnley, Craig Federighi, Matthew Firlik, Peter Hajas, Tim Isted, and Cheryl Thomas for their mentorship and guidance. This work was supported by the National Science Foundation under Grant 1464315, and an Undergraduate Research Grant from Northwestern University.

REFERENCES

[1] 2018. Marvel. https://marvelapp.com/

[2] 2018. Pokémon GO | Pokémon Video Games. https://www.pokemon.com/us/pokemon-video-games/pokemon-go/

[3] Kevin Chen and Haoqi Zhang. 2015. Remote Paper Prototype Testing. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*. ACM, New York, NY, USA, 77–80. https://doi.org/10.1145/2702123.2702423

[4] Henry Been-Lirn Duh, Gerald C. B. Tan, and Vivian Hsueh-hua Chen. 2006. Usability Evaluation for Mobile Device: A Comparison of Laboratory and Field Tests. In *Proceedings of the 8th Conference on Human-computer Interaction with Mobile Devices and Services (MobileHCI '06)*. ACM, New York, NY, USA, 181–186. https://doi.org/10.1145/1152215.1152254

[5] J. F. Kelley. 1984. An Iterative Design Methodology for User-friendly Natural Language Office Information Applications. *ACM Trans. Inf. Syst.* 2, 1 (Jan. 1984), 26–41. https://doi.org/10.1145/357417.357420

[6] Joanna Lumsden and Ryan MacLean. 2008. A comparison of pseudo-paper and paper prototyping methods for mobile evaluations. In *Proceedings of the On the Move to Meaningful Internet Systems: OTM 2008 Workshops*. 538–547.

[7] Christian Monrad Nielsen, Michael Overgaard, Michael Bach Pedersen, Jan Stage, and Sigge Stenild. 2006. It's worth the hassle!: the added value of evaluating the usability of mobile systems in the field. In *Proceedings of the 4th Nordic conference on Human-computer interaction: changing roles*. ACM, 272–280.

[8] RelativeWave. 2017. RelativeWave - Form. http://relativewave.com.

[9] Origami Studio. 2017. Origami Studio - Design Prototyping. http://origami.design.