

User Interface Prototyping for Handheld Mobile Augmented Reality Applications

Antonia Kampa^(✉), Kathrin Stöbener^(✉), and Ulrike Spierling^(✉)

Faculty of Design, Computer Science, Media, Hochschule RheinMain,
University of Applied Sciences, 65195 Wiesbaden, Germany
{antonia.kampa,kathrin.stoebener,ulrike.spierling}@hs-rm.de

Abstract. We introduce MockAR, a prototyping application for designing user interfaces for mobile handheld Augmented Reality, to be used by non-programming media designers. It has been successfully employed in the project SPIRIT, which develops location-based AR storytelling for outdoor historical sites, including complex interaction and navigation features based on mobile phone sensors and markerless image recognition.

Keywords: Augmented reality · User interface design · Prototyping

1 Introduction: AR User Interface Design Prototyping Tool

Interaction with Augmented Reality (AR) and the need for appropriate user interfaces have been topics of research for several decades [4, 6, 8]. However, available design frameworks mainly target expert developers and -users such as programmers. There are no standard or commonplace user interaction metaphors, as it has widely been a niche topic. For a long time, AR has also been associated with special devices such as head-mounted displays and extra hardware for tracking or tangible interfaces [6]. Only recently, new kinds of mobile apps for off-the-shelf phones and tablets become widespread. They include AR in form of a visual overlay of digital information on real objects or landmarks on a camera image. For entertainment and education purposes, such AR applications are to be conceived in a consumer market. This development produces an increasing need for non-programmers, such as media designers, to tackle AR elements as new media to be included in the conception and design of communication with their target groups. Like in other areas of media conception, interface design and graphics, designers need easy to use prototyping tools.

In the research project SPIRIT, a location-based mobile AR application has been developed that includes AR video overlay on specific spots of a cultural heritage site. Visitors shall experience entertaining on-the-spot storytelling, together with scavenger-hunt-like spatial interactions and information seeking, based on mobile device sensors. Storytellers and UI designers have been involved in the design of UI features integrated with the interactive story's concept. Figure 1 shows an early functional prototype, that had to be redesigned for more user guidance. This interdisciplinary development

afforded several stages of prototyping before the system's full implementation [1]. Easy prototyping of location-based AR experiences requires tools that go beyond existing approaches. As a side result in the SPIRIT research project, we developed the AR UI prototyping tool MockAR for the project's non-programming designers. It supports the widely known principle of 'wire framing', used to draft and test so-called click dummies in early development stages [9]. MockAR plays back image sequences that can be experienced based on a wide range of user interactions using handheld devices' sensors.



Fig. 1. Early functional prototype of the SPIRIT application

2 SPIRIT: Outdoor Cultural Heritage AR Video Entertainment

The SPIRIT AR application delivers location-based stories, targeted at outdoor and indoor cultural heritage education on site. The goal is to achieve a feeling of 'presence' on the spot supported by interaction with the application, so that visitors can engage with their historical surroundings. The AR overlay consists of pre-produced video snippets [10] representing 'spirits of the past' at the location of their former living (compare Fig. 1). Technically, the developed AR system relies on GPS and image recognition to trigger content, using no dedicated markers, but reference images found in the physical environment. This technological vision of a natural seamless interaction is at the same time a challenge for designing the right affordances for users, as there are no technically given 'calls to action' for finding the right spots. While there are no standards, UI designers need to be included in the design process, to come up with innovative ideas. Therefore, an interdisciplinary user interaction design process is needed.

2.1 Augmented Reality User Interaction in SPIRIT

While several common AR interaction styles are not necessary, such as manipulating 3D objects, there are other challenges for users to be addressed in a user interface, like orientation in the environment and interacting with spirit videos.

In the SPIRIT scenario, the main AR user interaction consists of pointing the camera towards interesting surroundings, anticipating that (markerless) image recognition leads to a trigger for content. Users need to steady their mobile device until the recognition algorithm has successfully captured 'reality', which can then be overlaid with

information, videos or other media assets. In practice, a user searches the whole area for suitable visual backdrops that could serve as image triggers, with the help of a UI that supports the right anticipation. On success, ghost-like characters as semi-transparent video deliver story content in form of dialogues, while users keep the camera focus on the triggered backdrop also accessing limited dialogue functions. In summary and besides this basic interaction loop, user actions include:

- walking between specific outdoor locations relevant for the told story,
- searching visual backdrops as image triggers (finding interesting story locations),
- tapping on buttons in a conventional GUI-like interface,
- tilting the mobile device up and down, changing between augmented story and a general GUI.

Further, we develop and evaluate novel input styles using different sensors, such as turning around like in a panoramic image, turning at specific angles up to 360°, or pitching and shaking the mobile device. We also test setting a timer for automatic displays without a trigger image, for example as several step-by-step hints and instructions, helping the user to find the right locations and backdrops.

2.2 User Interface Design Prototyping Versus User Interface Development

For the above-mentioned interactions, the mobile application needs to be equipped with various UI elements offering meta-information and hints for the user. As there is not a straight-forward standard solution, several optional creative design ideas are developed. In a cycle of design and formative evaluation, these alternatives need to be tested long before final implementation. In traditional interactive media (such as for desktop, web and touch applications using point-and-click or tap interaction), so-called wireframes are created with basic tools [9] as simple as for example MS-PowerPoint. The result can be experienced clicking linearly through ideal progressions of use, mostly at the desktop. This approach has been used at first, however it is limited for experiencing location-based AR issues. These include the difference between lighting and readability conditions indoors vs. outdoors (see Fig. 2), the handling of the device including holding its weight and reach space for touch interaction, as well as issues concerning the specificity of the outdoor location, far away from office space. To fill this gap, we developed MockAR, an application to be used for low-fidelity prototyping of UI design alternatives.



Fig. 2. Mockup UI evaluation indoors (left) and outdoors (right).

3 User Interface Wire Framing for AR Technology

An increasing number of wireframing tools for non-programmers exist, suggesting that creative design and programming tasks are distributed to different persons in a team [3, 5, 7]. However, these lack the possibility to test the above-mentioned issues of mobile AR, while dedicated AR tools are programming libraries [8, 11]. The WozARd [1] prototyping tool for non-programmers provides AR, however, it is limited by its Wizard of Oz concept relying on a supervisor. In short, we did not find suitable tools enabling non-programmers to quickly test and evaluate AR user interaction ideas, which motivated the development of MockAR.

3.1 MockAR Details

MockAR engages non-programmers in the UI design process of AR interactions, making this task accessible to a wider group of creators used to wireframing. MockAR emulates AR effects by displaying a linear order of semi-transparent images in front of camera-captured video. It makes full use of available sensors of mobile devices as input options to advance the image sequence. The image succession is determined by alphabetical order of image file names (0–9, A–Z). Transition between two images is set by an interaction keyword (e.g. *shake*, *pitch*, *turnL*, *turnR*, *tap*) contained in the last six letters of the file name. In SPIRIT, we used MockAR with interactions like tapping on screen (default), shaking, pitching by 90°, turning the device left and right (Fig. 3), and setting a timer. A *six digit number* sets timed image transitions in milliseconds. Technical skills needed for using MockAR are image editing, changing filenames, saving files on an Android device and installing and starting the Android app on a mobile device. Therefore we assume MockAR is easy to use for non-programming user interaction designers.



Fig. 3. Example user interactions from left to right: tap, shake, pitch, turn left & turn right

Utilizing devices' sensor data, we use basic Android Software Development Kit [2] functions to access device coordinates, such as of the accelerometer for detecting movements in landscape mode, as well as the camera2 package for capturing video used as AR background overlaid by a FrameLayout displaying transparent PNG images. Images are kept in a specific folder on the device's SD-card. After reading file names and finding interaction keywords, a listener on specified device sensors or the timer in milliseconds is set. Figure 3 shows user interactions, and Fig. 4 user interaction keywords in the last six characters of the image file name. MockAR displays the transparent PNG image files from left to right.

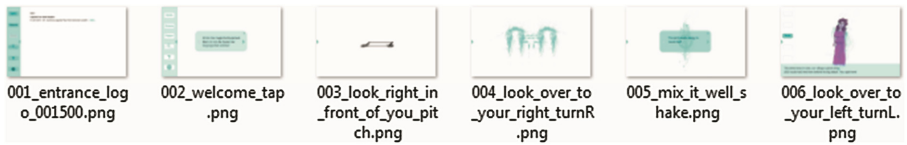


Fig. 4. Image file names with specifying user interaction keywords

3.2 Proof of Concept

Figure 5 shows an example of how MockAR has been successfully used to evaluate creative ideas. Designers came up with the idea to conceive the app generally as metaphorical ‘magic equipment’, asking a user to apply ‘energy’ in order to visualize a spirit. The question then was how ‘applying energy’ can be performed. One idea was to let the user shake the tablet (Fig. 5, upper row from left to right). Alternative ideas included using two thumbs to swipe at the same time, or, just standing still and waiting while pointing at a backdrop (Fig. 5, lower row from left to right).

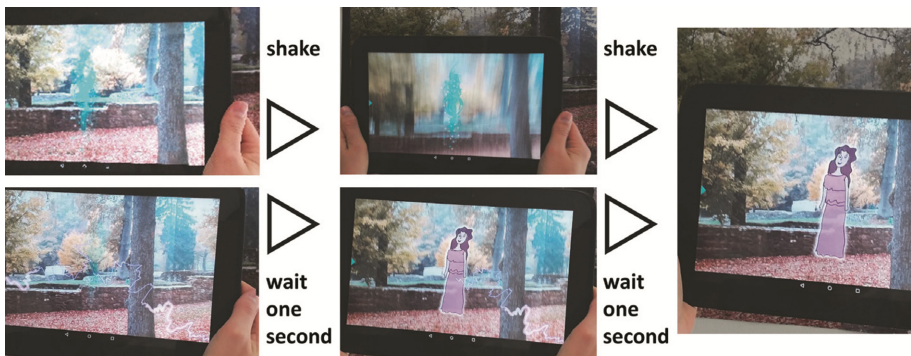


Fig. 5. Two alternative actions to evoke the visualisation of spirits, tested with MockAR

A non-programming media artist quickly built the alternatives by creating illustrations and including the according keywords for interaction into the filenames of an ordered image sequence. In fact, this way we found that shaking the device in this situation has been experienced as unnatural and difficult by test users. With the help of MockAR, such ideas could be tested and filtered in an early phase of development, when implementation of the more complex interactions was yet to be done. Also other quick prototyping of UI element positions and the sequencing and duration of video snippets saved time in the collaboration with the programmers.

MockAR was also tested in a student project involving four non-programmers in AR application design. Due to the simplified approach, they were able to build alternative design interactions within 20 min each. After building a series of designs, MockARs users requested further functionalities regarding non-linear playback and gesture input, which will be set on the development agenda.

4 Conclusion

MockAR has been developed as a side endeavor that became necessary within the project SPIRIT, while designing for a complex and novel user interaction in location-based AR with handheld devices. As recently, this area of development appears to prosper as a new field for media designers, easy-to-use prototyping tools for non-programmers are needed. MockAR established an iterative and interdisciplinary design and communication process within the project, which proved to be more difficult when designers had to wait for implementation by the AR developers.

Acknowledgements. This work has been funded (in part) by the Federal Ministry of Education and Research (BMBF) in Germany (03FH035PA3). We thank all project members for their support.

References

1. Alce, G.: Prototyping methods for augmented reality interaction. *ACM Siggraph Comput. Graph.* **39**(1), 17–22 (2015)
2. Android SDK, sensors overview in android software development (2016). http://developer.android.com/guide/topics/sensors/sensors_overview.html
3. Aurasma development kit (2016). <https://www.aurasma.com/>
4. Bai, Z., Blackwell, A.F.: Analytic review of usability evaluation in ISMAR. *Interact. Comput.* **24**(6), 450–460 (2012)
5. Balsamiq. Rapid, effective and fun wire framing software. Balsamiq (2016). <https://balsamiq.com/>
6. Billinghurst, M., Grasset, R., Looser, J.: Designing augmented reality interfaces. In: *ACM SIGGRAPH Computer Graphics - Learning through computer-generated visualization*. vol. 39, no. 1, February (2005)
7. Layar, augmented reality, interactive print (2016). <https://www.layar.com/>
8. MacIntyre, B., et al.: DART: a toolkit for rapid design exploration of augmented reality experiences. In: *Proceedings of the 17th annual ACM Symposium on User Interface Software and Technology*. ACM (2004)
9. Snyder, C.: *Paper Prototyping: The Fast and Easy Way to Design and Refine User Interfaces*. Morgan Kaufmann Publishers, Boston (2003)
10. Spierling, U., Kampa, A.: Structuring location-aware interactive narratives for mobile augmented reality. In: Mitchell, A., Fernández-Vara, C., Thue, D. (eds.) *ICIDS 2014. LNCS*, vol. 8832, pp. 196–203. Springer, Heidelberg (2014)
11. VUFORIA SDK. Vuforia play mode for unity (2016). <https://developer.vuforia.com/library/articles/Training/Vuforia-Play-Mode-for-Unity>