9-2019

Examining augmented virtuality impairment simulation for mobile app accessibility design

Tsu Wei, Kenny (ZHU Shuwei, Kenny) CHOO
Singapore Management University, kennychoo@smu.edu.sg

Rajesh Krishna BALAN

Rajesh Krishna BALAN
Singapore Management University, rajesh@smu.edu.sg

Follow this and additional works at: https://ink.library.smu.edu.sg/sis_research

Part of the Programming Languages and Compilers Commons, and the Software Engineering Commons

Examining Augmented Virtuality Impairment Simulation for Mobile App Accessibility Design

Kenny Tsu Wei Choo
Singapore Management University
kennychoo@smu.edu.sg

Rajesh Krishna Balan
Singapore Management University
rajesh@smu.edu.sg

Youngki Lee
Seoul National University
youngkilee@snu.ac.kr

ABSTRACT

With mobile apps rapidly permeating all aspects of daily living with use by all segments of the population, it is crucial to support the evaluation of app usability for specific impaired users to improve app accessibility. In this work, we examine the effects of using our *augmented virtuality* impairment simulation system–*Empath-D*–to support experienced designer-developers to redesign a mockup of commonly used mobile application for cataract-impaired users, comparing this with existing tools that aid designing for accessibility. We show that the use of augmented virtuality for assessing usability supports enhanced usability challenge identification, finding more defects and doing so more accurately than with existing methods. Through our user interviews, we also show that augmented virtuality impairment simulation supports realistic interaction and evaluation to provide a concrete understanding over the usability challenges that impaired users face, and complements the existing guidelines-based approaches meant for general accessibility.

CCS CONCEPTS

• **Human-centered computing → Systems and tools for interaction design**; **Ubiquitous and mobile computing systems and tools**; **Accessibility design and evaluation methods**; **Accessibility systems and tools**; *Ubiquitous and mobile computing design and evaluation methods.*

KEYWORDS

accessibility; empathetic design; mobile app design; augmented virtuality; virtual reality

1 INTRODUCTION

As we well know, mobile apps are a primary source of interaction for a large percentage of the population. These apps are usually developed for use by a younger and generally able-bodied population. However, the global elderly population is increasing and is expected to reach 16.7% by 2050 [18]. How usable would current mobile apps be when used by an elderly population with diminished eyesight, hearing, and other senses?

In this paper, we present the results of a study designed to answer that question as it relates to a *visual disability* of cataracts and from a mobile app developer's perspective. In particular, how can we provide an effective development environment that allows a mobile app developer to refactor the user interface components of their app to make it more accessible when used by an elderly population.

To provide this perspective, we used *Empath-D* [21], our system that was developed and demonstrated at MobiSys 2018 as it provides a novel Virtual Reality (VR) environment where developers can experience how their mobile apps would operate when used by individuals with various visual physical ailments such as cataracts and glaucoma–both common ailments amongst an elderly population. *Empath-D* has many novel elements such as 1) its use of virtual reality allows many different visual and auditory ailments to be experienced, and 2) it uses a real phone device held by the user to capture actual touchscreen events and these events are replayed in the VR world on a virtual phone that is running the actual app in an Android emulator. Thus the user can run the actual app in the VR environment and interact with it naturally–providing an *augmented virtuality* experience [27]. Figure 1 provides a visual description of how *Empath-D* works.

We conducted a rigorous design study to understand how developers could use a solution such as *Empath-D* to build
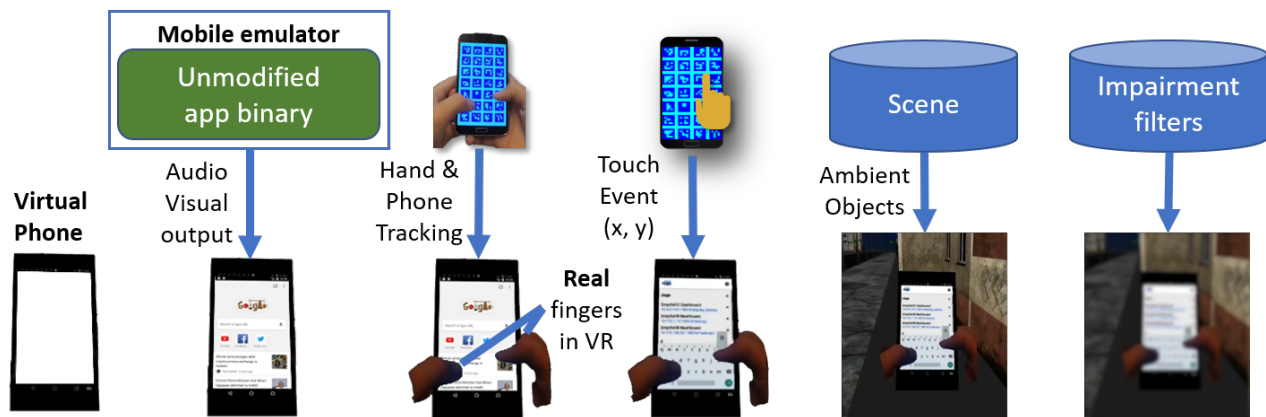
Figure 1: Composing the augmented virtuality view in *Empath-D*

apps that were more accessible to a disadvantaged population. In particular, without any significant loss of generality, we focused on an elderly population suffering from cataracts. To conduct this study, we first recruited 4 elderly (>60 years old) participants suffering from cataracts to understand the usability challenges they faced when using a popular mobile app–in this case, we used the *Instagram* app on Android. We then recruited 10 app developers, between the ages of 21 and 31, to modify a mockup of Instagram to be more accessible. We compared their performance with and without *Empath-D* and evaluated their output against both accessibility guidelines and subjective evaluation by our 4 elderly participants.

Our work yields insights on a number of issues involved in designing mobile apps with an augmented virtuality impairment simulator interface like *Empath-D* and over existing tools:

- How do existing accessibility checking and inspection tools perform in a design task?
- What are the usability challenges that an augmented virtuality impairment simulation identify that existing solutions cannot?
- What are the challenges with using augmented virtuality simulation for design?
- Is an augmented virtuality impairment simulator more usable for design?

Overall, our analysis of these questions contributes to a fundamental body of knowledge on the use of augmented virtuality to design for impairments that is novel and has not been adequately addressed.

2 BACKGROUND

Mobile Application Accessibility

While much work has been done in the area of accessibility for the web [9, 12, 22, 32], considerably less work has been done on mobile app accessibility. With web content increasingly being accessed from mobile devices, the Worldwide Web Consortium Web Accessibility Initiative (W3C WAI) recognised the convergence and released a note to help map the Web Content Accessibility Guidelines (WCAG) [9] for mobile devices [31]. Beyond web accessibility, past work has characterised mobile application accessibility in areas such as mobile health apps [28], for the blind [10], or to develop frameworks for analysis at scale [33].

While the WCAG has been in place since 1999 (WCAG 1.0 [11], and subsequent updates released to its current version of WCAG 2.1), large percentages of web sites are still inaccessible to users with disabilities [13, 23]. Lazar et al. [22] provide some insights on the challenges of implementing accessibility on webpages from developers' perspectives. The most pertinent perspectives to our work, were web developers responses that they *lack training, had inadequate software tools, and were confused by the accessibility guidelines.* Our work developing *Empath-D* [21] and validation in design here aim to address these. Our focus, is on *tools that support accessibility design for mobile apps.*

To support developers to consider accessibility in their designs, Apple [4] and Google [16] both released mobile accessibility guidelines, which are considerably simpler than the WCAG. They also developed tools–Google Accessibility Scanner (GAS) [17] and Apple Accessibility Inspector [3]– that examine apps for accessibility on a screen-by-screen basis. We incorporated Google Accessibility Scanner in our baseline condition as it provides the most direct assessment of Android apps for accessibility considerations.

Virtual Impairment Simulation for Design

Virtual impairment simulation as an approach to support iterative design may offer significant advantages by offering authentic first-person perspectives. However, this needs to be considered in the frame of the adaptability of disabled users. Silverman et al. [35] note that many impairment simulations often emphasise the initial challenges of becoming disabled. This may lead to overcompensation for the abilities of disabled users in designs, which may not sit well with users who have adapted around their disabilities. Our vision in developing *Empath-D* [21] is to provide a flexible system that may be calibrated to suit the desired design objectives for target user abilities. The detailed validation studies in this paper demonstrate the viability of *Empath-D* as an approach to target user abilities (in particular, visual abilities). This provides a customisable platform by which future work in the research community may implement impairment simulations that are accurate to the *adapted abilities* of impaired users and examine the design outcomes.

Virtual impairment simulation for design has also been employed in multiple settings. In one of the earlier works on the subject, Higuchi et al. [19] proposed a tool to simulate the visual capabilities of the elderly on digital scans of photos of control panels. Mankoff et al. [25] developed a tool to simulate visual and motor impairments for any desktop application (including web interfaces). More recently, a head mounted display with visual impairment simulation was used in a virtual 3D model of a city to understand the issues involved in wayfinding and target location [39]. Ates et al. [6] implemented an augmented reality see-through approach–using stereoscopic cameras mounted on an Oculus VR–to simulate impairments to inspect mobile applications. While their [6, 39] work is the closest to our tool *Empath-D* [21] that is used in our studies–our goal was to examine the usability challenges that may be identified and improved in mobile apps using an augmented virtuality approach in performing actual mobile application design.

Augmented Virtuality Impairment Simulation

We used our augmented virtuality system, *Empath-D* [21] to conduct our studies to observe and investigate the effects of using augmented virtuality impairment simulation in designing mobile apps.

Empath-D is an augmented virtuality system composed of a VR display mounted with a depth camera, a dummy phone, with a mobile emulator running on a computer. The software runs off the same computer and tightly orchestrates the different pieces of hardware such as to provide sufficiently real-time interaction performance [21] for most purposes (237.7 msec end-to-end touch latency). To experience *Empath-D,* an unmodified app binary–the designed

app–is run in the emulator (See Figure 1). The RGBD images from the depth camera and the fiduciary marker array on the dummy phone are processed to support hand and phone tracking. Touch events on the dummy phone are also forwarded and processed. Finally, the scene and impairments are rendered–with the corresponding tracking–onto the VR display using the graphics engine.

To use Empath-D, the designer-developers simply need to 1) provide an unmodified app apk to be tested to the Empath-D simulator, 2) configure the impairment conditions and severity to test, and 3) wear a head-mounted VR device and use the application in the simulated environment to identify accessibility problems.

3 STUDY METHOD

We designed and conducted two studies to examine if *Empath-D* is a useful tool to design mobile applications for impairment-specific accessibility. In the studies, we focused on 1) cataract-impaired users, one of the most common vision problems that the elderly experience and 2) a social media application–Instagram–an archetype that encapsulates many common interactions in mobile apps.

In the first study (S1), we recruited cataract-impaired users to examine the usability challenges that they faced under a few everyday use cases of Instagram. In the second study (S2), we recruited experienced designer-developers and got them to redesign a mock-up of Instagram using *Empath-D* and compared that against Google Accessibility Scanner.

Study 1: Cataract Impaired Users

Participants. We recruited 4 (all female), ages 65-71 cataract-impaired users (self-reported to be verified by doctors) to identify the usability challenges that they experience in using Instagram. We used strict participant selection criteria to reduce the effects of other visual impairments on the study condition (cataracts). In particular, all participants were selected for mild-moderate cataracts in *both* eyes, with minimal or no other visual impairments (e.g., low-degree myopia, no glaucoma, no age-related macular degeneration). With cataracts being a predominantly age-related disease [14], all participants unavoidably also had presbyopia–the loss of elasticity in the lens of the eye, which causes issues with near focus.

To confirm that our selection criteria was effective and that the usability challenges identified by participants were based on a similar level of cataract impairment, we asked participants to complete the CatQuest-9SF questionnaire [24]. The CatQuest-9SF is a questionnaire originally developed to measure pre- and post-cataract surgery outcomes. It was constructed in Swedish (with an English translated version)

but has been translated and examined (under Rasch analysis) for Malay, Chinese, and Italian populations, and has consistently shown good psychometric properties [1, 37].

Method. Participants came in for a 1 hour session, where we adopted a master-apprentice frame to explore their use of Instagram on a Samsung Galaxy S7. To identify the usability issues they faced with Instagram, we used an unmodified version with default settings and no accessibility options enabled. They were asked to 1) use the phone with or without corrective glasses as they would normally (in line with the notion of everyday living from [24]), and 2) hold their phones at a normal distance of 25-30cm from their eyes, and not compensate for visual problems (e.g., holding phone very close to see). We focused the participants on three particular use scenarios (T1-T3, See Table 1) that are common to Instagram, which cover a wide range of use common to many mobile applications (e.g., reading text, viewing pictures, swipes, touch, text input). Video recordings were made of the session, and coded for the analysis.

Table 1: Common use scenarios in Instagram framed as design tasks for Study 2

Task	Description
1	Like and bookmark a post
2	Read a post and post a comment
3	Send a message to a friend about where you are

Study 2: Designers using Augmented Virtuality Impairment Simulation

Participants. We recruited 10 (5 female) experienced HTML-CSS-JS designer-developers, ages 21-31 (mean 24.4) years. The participants were selected to not have any pre-existing uncorrected visual impairments–for example, myopia corrected by spectacles was allowed, but colour blindness was not. To ensure participants had the ability to perform the study task, we administered a coding test for HTML-CSS-JS, finding that all participants had the requisite skills (9 of 10 scored similarly, with only one participant demonstrating greater coding ability). The participants were divided into two conditions. In the first condition, C1, participants were given *Empath-D* (see Table 2 for the hardware used to run Empath-D), and in the second, C2, they were given *Google Accessibility Scanner* [17], a diagnostic tool for Android mobile app accessibility. To further ensure that participants in each group were unbiased in ability, we administered the Need For Cognition [8] scale that reflects an individual's inclination towards effortful cognitive abilities (in this case, redesign). We did not find significant differences in the mean

scores between the two groups of users (C1: 10.6 (SD = 5.9) and C2: 13.8 (SD = 7.8))).

Method. Participants came in for a full day study and were asked to perform the role of a mobile app designer, and redesign a mockup of Instagram for cataract impaired users to support the same three use scenarios described in S1 and for the same reasons (see Table 1). This allows us to match the usability challenges identified by cataract impaired users to the usability challenges addressed by designer-developers. For each task, participants were given 100 mins to redesign and develop the mockup, rolling over from each session to finally end with a working prototype. Participants were free to choose their process of working with the tools provided. They could modify the designs as they pleased (e.g., reposition, resize, recolour, remove), with only one limiting condition: they should not take away functions from the user interface. For instance, if the bookmark button was the sole means by which to bookmark a post, users cannot remove the bookmark button without implementing bookmarking in a different manner.

Table 2: Hardware used to run Empath-D.

Component	Device Used
VR display	Samsung Galaxy S7
RGBD camera	Intel RealSense SR300
Computer	CPU: Intel Core i7-3720QM (4 cores, 2.6 GHz)
	RAM: 16 GB DDR3
	GPU: NVIDIA GeForce GT 640M
Physical IO smartphone	Samsung Galaxy S7

We developed the mockup of Instagram using HTML-CSS-JS, and set it up to be compiled using Cordova [2] into an Android apk file. Only two popular JavaScript-based APIs were utilised to minimise the learning that was required of participants. *jQuery Mobile* was used to provide a mobile-like experience, yet having all the code in a single HTML file to simplify the organisation of the code. *jQuery* was provided to support simpler coding of the UI logic. This served as the baseline application mockup (see Figure 2) that all participants started with.

We deliberately chose HTML-CSS-JS as the underlying means to develop this app as it provides three distinct advantages: 1) users can focus on aspects of design, as compared to native-app coding, 2) they may easily inspect and simulate changes using browser-based mobile emulation (e.g., in Google Chrome), and 3) the HTML-CSS-JS can be run through existing accessibility checkers for WCAG 2.0 compliance.

Participants were briefed on the WCAG 2.0 [9], and given a link to an abstract version of the WCAG 2.0 geared towards
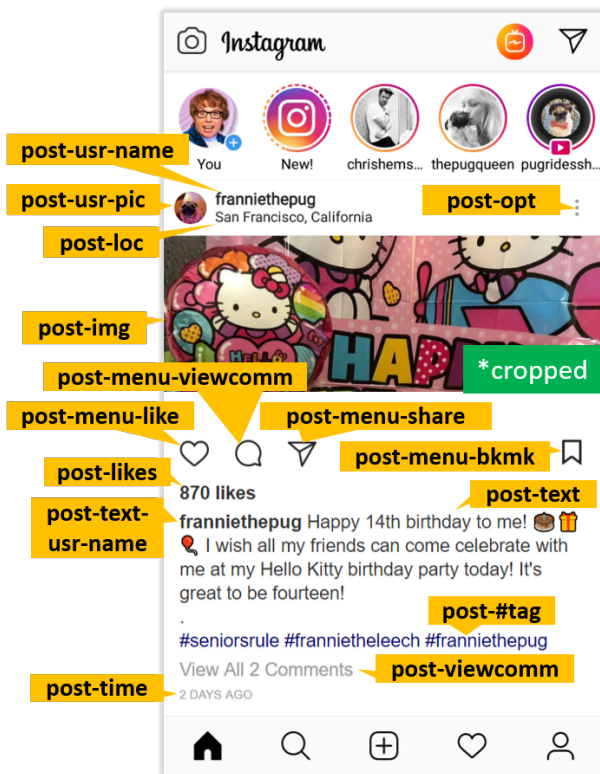
Figure 2: First page of the mockup baseline design used for S2 with example labels of UI elements used in the analysis. *Image source: Instagram, © franniethepug*

older users (WCAG 2.0-Elderly) [5]. The WCAG 2.0-Elderly was provided as a guiding document since it provides the closest match to our desired elderly demographic of which 78.6% suffer from cataracts [20]. Participants were also given a description of cataracts, and its underlying symptoms [36]. They also had unfettered access to the Internet to support seeking information on any aspect of the task. Lastly, to help participants to lower the challenges of implementation and focus them on design, we provided them with: 1) cheatsheets for HTML-CSS-JS, and 2) assistance from the experimenters on implementation (e.g., how to adjust HTML elements, code a JS function, or image editing) but not design issues. The support provided was meant to reduce the impact of individual differences in coding ability, which may negatively affect participants with weaker coding ability by distracting them from focusing on design.

All participants performed the study using a 13-inch Macbook, installed with tools common to web development (e.g., Sublime Text [38] / Atom, with code completion / syntax highlighting features). They were also free to install any tools that they preferred to use, though no participants felt the need to and were comfortable with the tools given. An

external monitor was connected to the computer and was recorded using a video camera to capture holistic data of all that the user was doing in the design task (e.g., searching for information, focusing on an element for redesign). A separate video camera in parallel was also used to capture the interactions that a user had with *Empath-D* or Google Accessibility Scanner. We adopted a think-aloud protocol, getting participants to verbalise their thoughts as they performed the design task. At the end of each task, participants were asked to fill in the NASA Task Load Index (NASA-TLX) with reference to the whole design task and System Usability Scale (SUS) with reference to the system they used (C1: *Empath-D* or C2: Google Accessibility Scanner).

4 RESULTS

S1: Cataract Impaired Users

From the CatQuest-9SF, all participants were found to match our targeted mild to moderate level of impairments, with cataract impairment scores ranging from 2.111-2.889 (2 being "some difficulty" and 3 being "great difficulty") (See Figure 3).
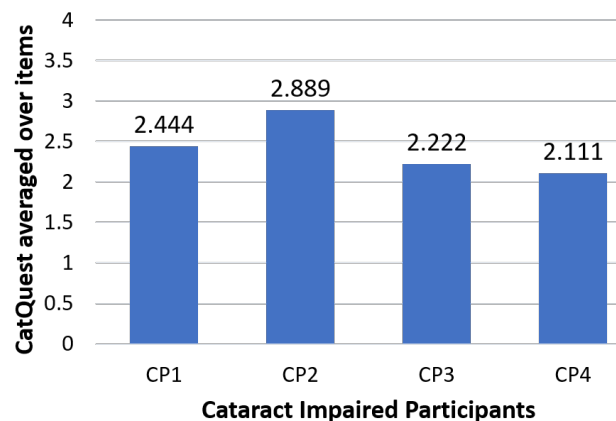
Figure 3: All participants had similar mild-moderate (between 2: "some difficulty" to 3: "great difficulty") levels of cataract impairments.

We analysed the observations in S1 to establish a base set of usability challenges that cataract impaired users face (See Cataract Impaired Users, Table 3), with respect to the core use tasks detailed in Table 1.

The challenges that they faced match the WCAG 2.0-Elderly guidelines in all challenges except *letter spacing*. One participant noted that bold text (e.g., *post-text-usr-name*) was harder to see than non-bold text (e.g., *post-text*), as the lower ratios of letter to stroke spacing caused letters to "clump together" (See Figure 2 for italicised examples). The converse problem also occurred, with low font weights, two users found text hard to read. All users noted that with larger

Table 3: Usability Challenges identified in C1 and C2, mapped to WCAG 2.0-Elderly and what Impaired Users identified.

Usability Challenges	WCAG 2.0 -Elderly	Cataract Impaired Users	C1 (*Empath-D*)	C2 (GAS)
Font size	✓	✓	✓	✓
Letter spacing		✓	✓	
Contrast	✓	✓	✓	✓
Image visibility	✓	✓	✓	✓
Icon visibility	✓	✓	✓	✓
Hit Target			✓	✓

font sizes, text became more readable. This demonstrates the need to *appropriately balance font size, font weight and letter spacing*–a design requirement that is often left to the judgement of designers. All participants also had problems reading low contrast grey text such as those of *post-viewcomm* and *post-time* (See Figure 2). For all participants, *contrast* was the biggest problem, often resulting in the inability to perceive the content at all. *post-viewcomm* and *post-time* are deliberately designed to be grey, such as to convey its relatively lower importance in the UI. However, with elderly users, this results in the loss of perceiving some information about the post. For *post-time*, elderly users cannot tell when the post was posted. For *post-viewcomm*, elderly users cannot perceive how much discussion is going on and consequently decide if to click and find out more.

S2: Identifying and Fixing Design Problems

We performed a detailed breakdown of the changes made to the base mock-up at the element level, and separated them into positive and negative categories for each condition (see Figure 4). We determined positive/negative changes by how they are in line with the WCAG 2.0 and data from cataract impaired users (S1), but ignore for the magnitude of changes and how they affect overall usability. For example, if a design guideline indicates that font sizes should be enlargeable/enlarged to support vision, and the user made a change to enlarge the font size, this counts as a positive change. However, if the user instead reduces the font size, this registers as a negative change. We further separated each positive and negative category into the key usability challenges (e.g., font size, letter spacing).

Accuracy and Coverage. In C1 (*Empath-D*), participants were able to more accurately (94.2%, 180 positive changes) identify usability challenges than without (C2: Google Accessibility Scanner; 85.6%, 160 positive changes). From Figure 4, we see that *Empath-D* supports positively identifying 19.4% more usability challenges across the different UI elements. Notably, *Empath-D* allowed participants to identify *letter spacing* problems, a usability challenge not picked up at all in

C2. This was because Google Accessibility Scanner did not report issues of letter spacing. Users in C1 however, could *directly observe* through *Empath-D* impairment simulation that words "clumped together" (P4), and made appropriate changes. Participants in C2 picked up 40% more contrast usability challenges than in C1, however, this is at the detriment of wrongly identifying 100% more contrast usability challenges across the different UI elements. This indicates that Google Accessibility Scanner (C2) is unreliable in identifying contrast usability challenges.

Examining for all UI elements where positive changes were identified, participants in C1 were able to uniquely identify usability challenges in 7 (15.2% of all elements) UI elements: *post-usr-pic*, *post-opt*, *post-img*, *msg1-back*, *msg1-newmsg*, *msg1-pic*, and *msg2-back*. This is pertinent particularly since images (*post-img*) (see Figure 2), are a central mechanic to deciding to further interact in Instagram, and that cataract impaired users (in S1) reported not being able to perceive the details in images, particularly those that are complex (e.g., colourful images with fine details that may obscure the focal subject of the image).

Magnitude of Changes. To better understand the effects of the changes made, we collated the UI elements into their functional categories and analysed the differences between C1 and C2 (see Table 4). Only letter spacing is definitively significant as users in C2 did not make letter-spacing changes at all. While no significant differences were found for the rest of the conditions, this is largely due to the nature of a design task. Take for instance the size of text. The only change that may be specified is that an element should meet a minimum size in order to be perceivable. However, users may increase the size of text much more than that, as they feel that it does not impact scrolling significantly.

Given that text is central to performing T1-T3, we see that in both C1 and C2, participants increased the font sizes, with C1 being larger across all UI functions. Letter spacing was modified only in C1, with spacing ranging from 1.5-1.83 pixels across the different UI functions.

This was corroborated by the observational data captured during the experiments. In C2, users found it hard to understand and use the relevant WCAG 2.0 success criteria (1.4.4 - Resize text). While 1.4.4 suggests that "text can be resized without assistive technology up to 200 percent without loss of content or functionality", participants also considered the resulting penalty to interaction (having to scroll a lot more due to larger text/buttons). Conversely, participants in C1 had the ability to directly simulate the cataract impairment using *Empath-D*, verifying that the font sizes that they chose were sufficiently large such that they could be perceived, and yet minimise the penalty to interaction.

UI element	Pos: Font size	Pos: Letter spacing	Pos: Contrast	Pos: Image Visibility	Pos: Icon Visibility	Pos: Hit Target	Neg: Font size	Neg: Letter spacing	Neg: Contrast	Neg: Image Visibility	Neg: Icon Visibility	Neg: Hit Target
post-usr-pic				a								
post-usr-name	c	a	a									
post-loc	c	a	b				b		b			
post-opt					a							
post-img				a								
post-menu-like			c		c	c						
post-menu-viewcomm			b		c	c						
post-menu-share			b		c	c						
post-menu-bkmk			b		c	c			b			
post-likes	c	a	a						b			
post-text-usr-name	c	a							b			
post-text	c	a										
post-#tag	c	a							c			
post-viewcomm	c	a	c									
post-time	c		c						b			
comm-pic				c								
comm-name	c	a							c			
comm-text	c	a							c			
comm-#tags	c	a							c			
comm-time	c		c						b			
comm-usr-pic			c									
comm-input-placeholder	c								b			
comm-input-text	c								b			
comm-post	c		c									c
msg1-title	a								c			
msg1-back									a			
msg1-newmsg									a			
msg1-search	c											
msg1-suggestions	c											
msg1-pic				a								
msg1-name	c											
msg1-lastseen	c											
msg2-back				a								
msg2-pic				c								
msg2-name	c								b			
msg2-msg	c								b			
msg2-takepic			b		c				b			
msg2-input-placeholder	c											
msg2-input-text	c					c						a
msg2-input-box			b			c						c
msg2-post	c		c			c						b

Figure 4: Coverage matrix of the usability challenges positively/negatively addressed by designer-developers. a indicates challenges identified only by Empath-D (C1). b indicates challenges identified only by Google Accessibility Scanner (C2). c indicates challenges identified by *both* Empath-D and Google Accessibility Scanner. Blank cells indicate that no users identified challenges in either condition for that UI element.

Figure 5 shows the baseline design (left), an example design by P8 in C2 (centre), and P10 from C1 (right). P8 focused on supporting the design task T1, enlarging the two buttons of like and bookmark to 11 times larger a hit and visible area

than the baseline. This resulted in less content being able to be seen in one page view (pushing down *post-likes*). While the much larger buttons help users to accurately press the buttons, we find from S1 that the cataract impaired users reported not having problems with pressing the buttons due to their familiarity with using mobile apps such as WhatsApp or Facebook. Comparatively, P10's design made moderate icon size, font size, and letter spacing adjustments, retaining more content that may be observed in one page view.

Table 4: UI elements, grouped by key functions showing mean changes made by designer-developers.

UI Function	Cond -tion	Font Size (%)	Letter Spacing (px)	Cont -rast (%)	Visi -bility (%)	Hit Target (%)
button	1	58	1.83	18	98	118
	2	44	0	60	185	107
image-content	1	-	-	-	5	-
	2	-	-	-	0	-
image-profile	1	-	-	-	63	-
	2	-	-	-	16	-
text-content	1	52	1.5	-6	-	-
	2	31	0	-5	-	-
text-input	1	59	-	0	-	23
	2	34	-	0	-	20
text-navigation	1	28	-	0	-	-
	2	25	-	-1	-	-
text-status	1	43	1.63	31	-	-
	2	28	0	12	-	-
text-username	1	68	1.69	18	-	-
	2	40	0	8	-	-

Usability of Tools

Participants using *Empath-D* reported higher SUS scores of 76.3 (*Acceptable*) as compared to using Google Accessibility Scanner, having a score of 65.2 (*Marginal*) [7]. One contributing reason is that all users found that Google Accessibility Scanner can be unreliable. It reported suggestions in app designs despite its obvious irrelevance or was unable to flag issues with some UI elements despite similar UI elements being flagged elsewhere in the UI, e.g., *post-menu-share*) (See Figure 6).

However, 2 users (40%) indicated that overall Google Accessibility Scanner was useful for the in situ examination of the accessibility of mobile apps, providing a more concrete means (compared to WCAG guidelines) to identify potential problems that users may then map to cataract impairment.

All users in C1 on the other hand found that *Empath-D* provided a concrete means to immediately identify usability problems. One user noted that *Empath-D* enabled her to identify problems with letter-spacing, which she stated was not in the WCAG 2.0. However, this is inaccurate–the WCAG
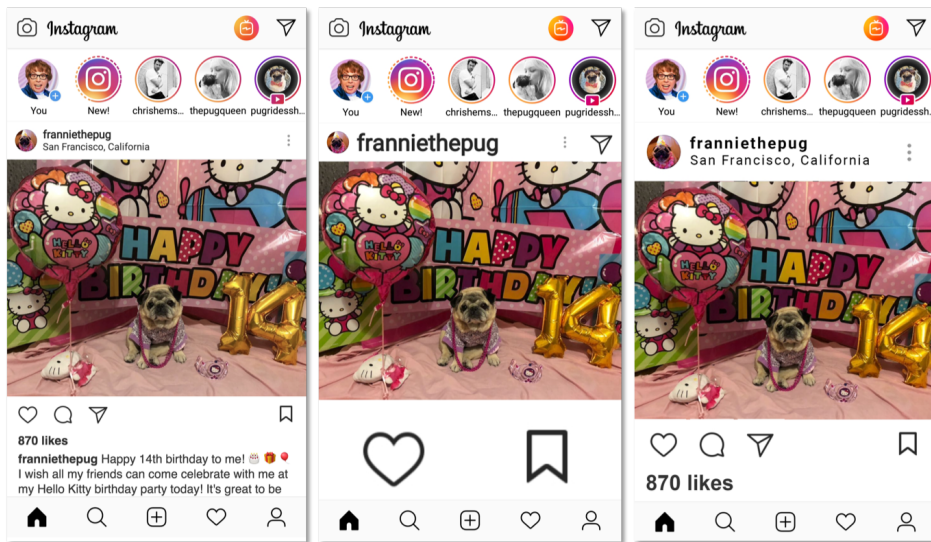
Figure 5: Balancing interaction design. Left: Unmodified base app. Centre: Design from P8 (C2: Google Accessibility Scanner) with very large like and bookmark buttons sacrificing content and interaction. Right: Design from P10 (C1: *Empath-D*) showing balance in design despite changes made to increase elements. *Image source: Instagram, © franniethepug*

2.0 does contain recommendations about adjusting letter spacing. We believe this reflects two issues in the WCAG 2.0: 1) that the WCAG 2.0 while comprehensive, is onerous for use, and 2) the difficulty in mapping the pathophysiology of impairments to specific guidelines. The interviews with users alluded to this. Users found the WCAG 2.0 to be extremely lengthy (4, 40%), and often found it hard to map guidelines relevant to cataract impairment (5, 50%). However, they noted that the WCAG 2.0-Elderly (and WCAG 2.0 itself) is a rich source of information to provide conceptual guidance on possible issues to focus on (8, 80%).

No significant differences were found in the scores of the NASA-TLX.

Table 5: System Usability Score and NASA-TLX

Condition	SUS	NASA-TLX
1	76.33‡	11.64
2	65.17†	10.71

not Acceptable / *Marginal*† / *Acceptable*‡ [7]

Automated Accessibility Checking. We ran an automated accessibility checking tool *aChecker* [15] on the final app designs from S2 over WCAG 2.0 guidelines, and at the *AA* level of compliance. The W3C does not recommend *AAA* conformance as a general policy for entire sites as it is not possible to satisfy all the success criteria for some content. *aChecker* bins checks into 3 categories of *Known*, *Likely*, and *Potential* problems.
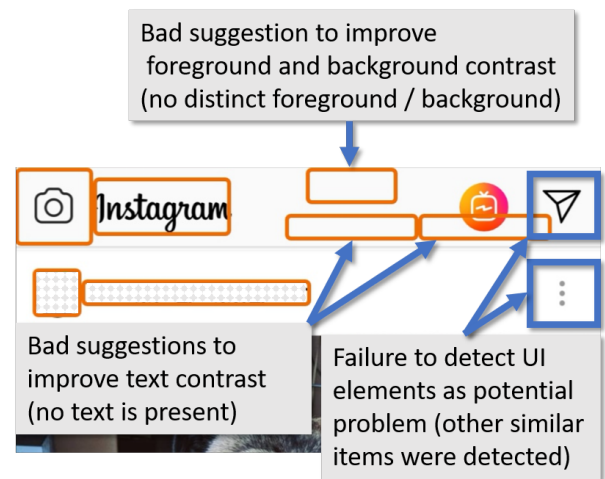

Figure 6: Scan from Google Accessibility Scanner showing obvious bad suggestions. Google Accessibility Scanner flags out potentially problematic areas with orange boxes.

aChecker flagged 40.5% more *Known* and 7.4% more *Potential* problems in C1 than in C2 (See Figure 7). No significant differences were found when comparing C1 and C2 in both *Known* and *Potential* conditions. From Figure 4, we know that C1 has better accuracy at identifying and fixing designs for cataract impaired users. We examined the issues that were flagged by *aChecker*, and found that many of the identified issues related to missing alt text labels for images. Such a requirement is only necessary to support the use of general

accessibility functions (e.g., talk back to read out what an element is). However, our focus is on designing specifically for an impairment, with no reliance on OS-provided accessibility support. The results from *aChecker* demonstrate the limitations of automated accessibility checking, which are inaccurate when used for designing specific impairments.
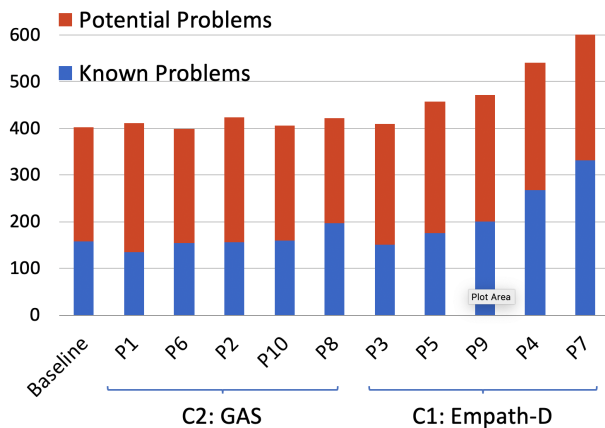

Figure 7: More known and potential accessibility problems identified in C1 than in C2 when aChecker is used.

5 DISCUSSION

The need for more and better tools to design mobile apps for accessibility. While much has been explored in designing *web content* for general accessibility, tools to support examining and designing mobile apps for accessibility largely remain an impoverished space. They primarily rely on the same basis that web content relies on–design guidelines–which have been previously suggested in previous studies to be insufficient to ensure web accessibility [12, 26, 29]. Our study utilising Google Accessibility Scanner reinforces this position, demonstrating its mismatch with user concerns as well as its potential for false positives. Users despite their best efforts, found it hard to map WCAG guidelines to cataract impairments, and even when they did, had to guess the effects of adjustments to the design.

While *Empath-D* allowed users to make adjustments to their designs concretely, it suffers from several problems. Firstly, the *attention* of users is required for them to examine usability problems. Without systematically examining the various components in a user interface, users may fixate on a particular issue, and miss out on identifying others. Secondly, users often want to find *several usability problems* at once when using *Empath-D*. However, being immersed in the VR, they are unable to make a note of all the identified issues. One potential solution to these two problems is to support annotation in the VR. A cursor overlay would allow users to

use digital styluses to circle and note down problematic aspects of the UI. These annotations (e.g., the circular strokes), may be used as a means to develop a heat map over areas that have been considered and remind users to examine other aspects of the UI.

Complementing augmented virtuality evaluation and guidelines. Even when users focus their attention to inspect a suspected problem area of the UI, they may not be able to conceptualise all problems related to that element. For instance, a user may observe that an icon button has insufficient stroke width to be adequately perceived, and adjusts accordingly. However, an alternate solution to this is to provide text labels, which may be a more appropriate solution for elderly users since it states its function, rather than relying on a user's memory. *Impairment-focused accessibility scanning* may be one potential solution. A filtered set of guidelines (possibly stemming from the WCAG), may be employed to scan the UI elements in the app to flag potential issues. This marries the benefits of *concrete experience* provided by *Empath-D* with the *completeness* provided by guidelines.

Simulation fidelity and range of testing. *Empath-D* needs to be further developed to support greater sensing capabilities that match or even supersede those provided by existing smartphone capabilities. The sensing capabilities of the physical smartphone should be appropriately represented in the VR in a seamless fashion. For instance, 1) location in the virtual environment can be mapped to the virtual coordinates in the environment and reflected in the virtual phone, and 2) virtual cameras show frames of the virtual environment on the virtual smartphone. These are but two examples of enhancements that support existing geolocation and camera applications respectively. By providing capabilities that may not be commonly available on existing hardware, and by providing realistic environmental simulations such as the home or the streets, *Empath-D* becomes a more powerful tool for interaction design. For example, by developing a virtual depth camera into *Empath-D*, one may test the interaction concepts in potentially dangerous environments such as the streets, which may present greater danger to users with disabilities. Naturally, the use of *Empath-D* also goes beyond designing for users with disabilities–absent impairment simulation–*Empath-D* still provides naturalistic interaction in a VR. This opens many opportunities for research that deals with novel interaction concepts with mobile and wearable devices.

Towards a Library of Impairments. Impairments are diverse in type, severity, and presentation. The studies in this paper are a first step in establishing a *library of impairments*, focusing in particular on cataract visual impairments. The envisioned library of impairments would allow designers to choose a representative group of users with disabilities to

design for. This in particular would help the issues of a lack of training and confusion with general accessibility guidelines as highlighted by Lazar et al. [22]. However, research involving users with accessibility needs are often fraught with difficulty (e.g., representing users in experimentation due to diversity, low availability) [30, 34]. We call on the accessibility research community to support this vision to develop a library of impairments.

6 CONCLUSION

In this paper, we studied the effects of using *Empath-D* , an *augmented virtuality* impairment simulation system, to support experienced designer-developers to design a mobile application for cataract-impaired users. Our studies with 4 cataract-impaired elderly and 10 experienced mobile developers show that with the aid of augmented virtuality and WCAG 2.0 guidelines, the developers were better able to identify the usability challenges of cataract-impaired users and made more positive changes to the app designs as compared with using Google Accessibility Scanner with WCAG 2.0 guidelines. The developers also noted that *Empath-D* is more usable than Google Accessibility Scanner. *Empath-D* helps them to immediately and uniquely identify usability problems (i.e., letter spacing) and see the effect of changes made, while the accessibility scanner often points out problems irrelevant to the target users and can be unreliable for many design components of the target application.

ACKNOWLEDGMENTS

REFERENCES

[1] Tassha Hilda Adnan, Mokhlisoh Mohamed Apandi, Haireen Kamaruddin, Mohamad Aziz Salowi, Kian Boon Law, Jamaiyah Haniff, and Pik Pin Goh. 2018. Catquest-9SF questionnaire: validation of Malay and Chinese-language versions using Rasch analysis. *Health and Quality of Life Outcomes* 16 (Jan. 2018). https://doi.org/10.1186/s12955-017-0833-3

[2] Apache Software Foundation. 2018. Apache Cordova. https://cordova.apache.org/

[3] Apple. 2015. Apple Accessibility Inspector. https://developer.apple.com/library/archive/documentation/Accessibility/Conceptual/AccessibilityMacOSX/OSXAXTestingApps.html

[4] Apple. 2018. iOS - Human Interface Guidelines for Accessibility. https://developer.apple.com/design/human-interface-guidelines/ios/app-architecture/accessibility/

[5] Andrew Arch and Shadi Abou-Zara. 2010. Developing Websites for Older People: How Web Content Accessibility Guidelines (WCAG) 2.0 Applies | Web Accessibility Initiative (WAI) | W3C. https://www.w3.org/WAI/older-users/developing.html

[6] Halim Cagri Ates, Alexander Fiannaca, and Eelke Folmer. 2015. Immersive Simulation of Visual Impairments Using a Wearable See-through Display. In *Proceedings of the Ninth International Conference on Tangible, Embedded, and Embodied Interaction (TEI '15)*. ACM, New York,

NY, USA, 225–228. https://doi.org/10.1145/2677199.2680551

[7] Aaron Bangor, Philip Kortum, and James Miller. 2009. Determining what individual SUS scores mean: Adding an adjective rating scale. *Journal of usability studies* 4, 3 (2009), 114–123.

[8] John T. Cacioppo and Richard E. Petty. 1982. The need for cognition. *Journal of Personality and Social Psychology; Journal of Personality and Social Psychology* 42, 1 (1982), 116. http://psycnet.apa.org/journals/psp/42/1/116/

[9] Ben Caldwell, Michael Cooper, Loretta Guarino Reid, and Gregg Vanderheiden. 2008. Web Content Accessibility Guidelines (WCAG) 2.0. https://www.w3.org/TR/WCAG20/

[10] Michael Crystian Nepomuceno Carvalho, Felipe Silva Dias, Aline Grazielle Silva Reis, and André Pimenta Freire. 2018. Accessibility and Usability Problems Encountered on Websites and Applications in Mobile Devices by Blind and Normal-vision Users. In *Proceedings of the 33rd Annual ACM Symposium on Applied Computing (SAC '18)*. ACM, New York, NY, USA, 2022–2029. https://doi.org/10.1145/3167132.3167349

[11] Wendy Chisholm, Gregg Vanderheiden, and Ian Jacobs. 1999. Web Content Accessibility Guidelines 1.0. https://www.w3.org/TR/1999/WAI-WEBCONTENT-19990505/

[12] Raphael Clegg-Vinell, Christopher Bailey, and Voula Gkatzidou. 2014. Investigating the appropriateness and relevance of mobile web accessibility guidelines. In *Proceedings of the 11th Web for All Conference*. ACM, 38.

[13] Amanda Davis. 2017. Most Developers Fail to Design Websites Accessible to People With Disabilities. http://theinstitute.ieee.org/ieee-roundup/blogs/blog/most-developers-fail-to-design-websites-accessible-to-people-with-disabilities

[14] P. J. Foster, T. Y. Wong, D. Machin, G. J. Johnson, and S. K. L. Seah. 2003. Risk factors for nuclear, cortical and posterior subcapsular cataracts in the Chinese population of Singapore: the Tanjong Pagar Survey. *British Journal of Ophthalmology* 87, 9 (Sept. 2003), 1112–1120. https://doi.org/10.1136/bjo.87.9.1112

[15] Greg Gay and Cindy Qi Li. 2010. AChecker: Open, Interactive, Customizable, Web Accessibility Checking. In *Proceedings of the 2010 International Cross Disciplinary Conference on Web Accessibility (W4A) (W4A '10)*. ACM, New York, NY, USA, 23:1–23:2. https://doi.org/10.1145/1805986.1806019

[16] Google. 2018. Android Accessibility Guidelines. https://developer.android.com/guide/topics/ui/accessibility/

[17] Google. 2018. Google Accessibility Scanner. https://play.google.com/store/apps/details?id=com.google.android.apps.accessibility.auditor

[18] Wan He, Daniel Goodkind, and Paul Kowal. 2016. *An Aging World: 2015. International Population Reports*. Technical Report. United States Census Bureau. 175 pages. https://www.census.gov/content/dam/Census/library/publications/2016/demo/p95-16-1.pdf

[19] K. Higuchi, Y. Sakaguchi, K. Sugiyama, and T. Nakano. 1999. Simulating the human vision of elderly for designing control panels. In *1999 IEEE International Conference on Systems, Man, and Cybernetics, 1999. IEEE SMC '99 Conference Proceedings*, Vol. 5. 703–708 vol.5. https://doi.org/10.1109/ICSMC.1999.815637

[20] T. Ho, N. M. Law, L. G. Goh, and T. Yoong. 1997. Eye diseases in the elderly in Singapore. *Singapore medical journal* 38, 4 (1997), 149–155.

[21] Wonjung Kim, Kenny Tsu Wei Choo, Youngki Lee, Archan Misra, and Rajesh Krishna Balan. 2018. Empath-D: VR-based Empathetic App Design for Accessibility. In *Proceedings of the 16th Annual International Conference on Mobile Systems, Applications, and Services (MobiSys '18)*. ACM, New York, NY, USA.

[22] Jonathan Lazar, Alfreda Dudley-Sponaugle, and Kisha-Dawn Greenidge. 2004. Improving web accessibility: a study of webmaster perceptions. *Computers in human behavior* 20, 2 (2004), 269–288.

[23] Rui Lopes, Daniel Gomes, and Luís Carriço. 2010. Web Not for All: A Large Scale Study of Web Accessibility. In *Proceedings of the 2010 International Cross Disciplinary Conference on Web Accessibility (W4A) (W4A '10)*. ACM, New York, NY, USA, 10:1–10:4. https://doi.org/10.1145/1805986.1806001

[24] Mats Lundstrom and Konrad Pesudovs. 2009. Catquest-9SF patient outcomes questionnaire: Nine-item short-form Rasch-scaled revision of the Catquest questionnaire. *Journal of Cataract & Refractive Surgery* 35, 3 (March 2009), 504–513. https://doi.org/10.1016/j.jcrs.2008.11.038

[25] J. Mankoff, H. Fait, and R. Juang. 2005. Evaluating accessibility by simulating the experiences of users with vision or motor impairments. *IBM Systems Journal* 44, 3 (2005), 505–517. https://doi.org/10.1147/sj.443.0505

[26] Jennifer Mankoff, Holly Fait, and Tu Tran. 2005. Is your web page accessible?: a comparative study of methods for assessing web page accessibility for the blind. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM, 41–50.

[27] Paul Milgram and Fumio Kishino. 1994. A taxonomy of mixed reality visual displays. *IEICE TRANSACTIONS on Information and Systems* 77, 12 (1994), 1321–1329.

[28] Lauren R. Milne, Cynthia L. Bennett, and Richard E. Ladner. 2014. The Accessibility of Mobile Health Sensors for Blind Users. (Dec. 2014). http://scholarworks.calstate.edu/handle/10211.3/133384

[29] Scott Milne, Anna Dickinson, Alex Carmichael, David Sloan, Roos Eisma, and Peter Gregor. 2005. Are guidelines enough? An introduction to designing Web sites accessible to older people. *IBM systems journal* 44, 3 (2005), 557–571.

[30] A. F. Newell, P. Gregor, M. Morgan, G. Pullin, and C. Macaulay. 2011. User-Sensitive Inclusive Design. *Universal Access in the Information Society* 10, 3 (2011), 235–243. https://doi.org/10.1007/s10209-010-0203-y

[31] Kim Patch, Jeanne Spellman, and Kathy Wahlbin. 2015. Mobile Accessibility: How WCAG 2.0 and Other W3C/WAI Guidelines Apply to

Mobile. https://www.w3.org/TR/mobile-accessibility-mapping/

[32] Christopher Power, Andre Freire, Helen Petrie, and David Swallow. 2012. Guidelines Are Only Half of the Story: Accessibility Problems Encountered by Blind Users on the Web. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '12)*. ACM, New York, NY, USA, 433–442. https://doi.org/10.1145/2207676.2207736

[33] Anne Spencer Ross, Xiaoyi Zhang, James Fogarty, and Jacob O. Wobbrock. 2017. Epidemiology as a Framework for Large-Scale Mobile Application Accessibility Assessment. ACM Press, 2–11. https://doi.org/10.1145/3132525.3132547

[34] Andrew Sears and Vicki Hanson. 2011. Representing Users in Accessibility Research. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '11)*. ACM, New York, NY, USA, 2235–2238. https://doi.org/10.1145/1978942.1979268

[35] Arielle M. Silverman, Jason D. Gwinn, and Leaf Van Boven. 2015. Stumbling in Their Shoes: Disability Simulations Reduce Judged Capabilities of Disabled People. *Social Psychological and Personality Science* 6, 4 (May 2015), 464–471. https://doi.org/10.1177/1948550614559650

[36] Singapore National Eye Centre, General Cataract & Comprehensive Ophthalmology Department. 2018. Cataract. https://www.snec.com.sg/patient-care/conditions-treatments/cataract

[37] Eirini Skiadaresi, Giuseppe Ravalico, Silvio Polizzi, Mats Lundstrom, Miguel Gonzalez-Andrades, and Colm McAlinden. 2016. The Italian Catquest-9SF cataract questionnaire: translation, validation and application. *Eye and Vision* 3, 1 (April 2016), 12. https://doi.org/10.1186/s40662-016-0043-9

[38] Sublime Text. [n. d.]. Sublime Text 3. https://www.sublimetext.com/

[39] Jani Vayrynen, Ashley Colley, and Jonna Hakkila. 2016. Head Mounted Display Design Tool for Simulating Visual Disabilities. In *Proceedings of the 15th International Conference on Mobile and Ubiquitous Multimedia (MUM '16)*. ACM, New York, NY, USA, 69–73. https://doi.org/10.1145/3012709.3012714