A Preliminary Analysis of Android Educational Game Accessibility

JESSE J. MARTINEZ, JAMES FOGARTY, and JON E. FROEHLICH, Paul G. Allen School of Computer Science & Engineering, University of Washington, United States

Android educational games are powerful learning tools but small, moving targets and graphic rendering implementations pose accessibility challenges to people with upper-body motor impairments. In this poster, we present findings from a qualitative accessibility evaluation of 30 popular Android educational games, identify and reflect on accessibility barriers, and provide preliminary design recommendations.

CCS Concepts: • **Human-centered computing → Accessibility design and evaluation methods**.

Additional Key Words and Phrases: accessibility, educational games, Android

ACM Reference Format:
Jesse J. Martinez, James Fogarty, and Jon E. Froehlich. 2021. A Preliminary Analysis of Android Educational Game Accessibility. In *The 23rd International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS '21), October 18–22, 2021, Virtual Event, USA.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3441852.3476532

1 INTRODUCTION

Mobile educational games are powerful learning tools, engaging children with gameplay while supporting exploration of a range of educational topics. As the global pandemic again highlights inequities in access to technology-mediated learning, it is important to ensure that such learning tools are available and accessible to all children.

Educational games present additional accessibility challenges beyond traditional apps: they have interactive elements that move and animate, their interfaces are highly stylized, and their unique interaction models can require unusual forms of engagement. Although prior work has contributed to understanding [10] and repairing [13] common mobile accessibility issues, it is unclear how research in traditional apps extends to educational games. To explore accessibility of games or apps, prior work has often employed manual reviews with a qualitative codebook [3, 7, 9–11]. We build on this prior work with a qualitative assessment of 30 popular Android educational games, evaluating their accessibility and identifying accessibility issues both in their game design and in their implementation.

We specifically focus on educational game accessibility for people with upper-body motor impairments. Because this is a diverse population, we further refine our scope by focusing on three facets of interaction that have been identified in prior work as important for accessibility for this population: (1) Interactive elements are large and easy to target [6]; (2) Interaction can be preformed by users with varying levels of dexterity [11]; and (3) Interaction is compatible with assistive technology, such as adaptive switches [8, 11].

This poster makes two research contributions: (1) findings from an initial qualitative assessment of the accessibility of 30 popular educational games on Android, examining each based on Switch Access, Google's Accessibility Scanner, and additional criteria drawn from the literature; and (2) a reflection on findings and opportunities to address inaccessibility in educational games, including means of repairing currently inaccessible games and recommendations for designers and developers of educational games to avoid re-creating existing accessibility issues.

2 BACKGROUND: ANDROID SWITCH ACCESS

Adaptive switches are a common form of assistive technology that can support interaction without the need to use a touchscreen or a pointing device. Android's native switch interface, Switch Access [1], provides two main modes of interaction: *linear scanning* and *point scanning*. With *linear scanning*, the device focuses on discrete elements of the interface and supports direct interaction with the currently-focused element. Importantly, for an element to be included in the linear scan, the developer must ensure that element is exposed in the application's view hierarchy and has the "focusable" attribute. With *point scanning*, the device first displays a line moving across the screen vertically, followed by a line that moves horizontally. The user stops each line at the desired x or y coordinate to specify a point at which to interact with the screen. Although point scanning is functional in all contexts (i.e., it does not depend on app developer implementation), it can be slower and less precise, and can require a higher level of dexterity than linear scanning.

3 SURVEY OF ANDROID EDUCATIONAL GAMES

To explore the accessibility of Android educational games, we conducted an accessibility assessment of 30 free educational games in the "Kids" section of the Google Play Store. We compiled the top 15 games in the "Educational Games" list for each age group (i.e., "Up to 5", "6-8", "9-12"), gathered on February 2, 2021. After removing duplicates and paid games from the list, we analyzed 30 games, including spelling games, math games, creative/art games, and educational video players. Our full coded dataset is available in our supplementary materials.

Each game was explored for up to 60 minutes, aiming to to explore and evaluate all distinct game modes within the game. We developed a procedure for assessing various accessibility issues that violated our previously-stated system accessibility principles. To explore how games worked with or failed to support common adaptive technology used for interaction with mobile devices, we first used Switch Access to explore what elements were focusable through linear scanning. To identify additional accessibility issues, we then used the Google Accessibility Scanner [2], a runtime tool that identifies common issues such as small targets, missing labels, or poor contrast. Finally, informed by prior work [7, 8, 11] and an initial exploration of educational games, we identified two additional potential challenges for accessibility: the presence of time-constrained tasks and the presence of moving targets. We defined each of these terms in order to consistently evaluate whether they existed in each app.

In sum, our assessment consisted of the following four components: (1) exploration of the app using Switch Access in linear scanning mode; (2) evaluation of the unique game screens using Google's Accessibility Scanner; (3) evaluation of whether the game has moving targets; and (4) evaluation of whether the game has time-constrained tasks.

4 FINDINGS

4.1 Switch Exploration

Of the 30 games surveyed, only three (i.e., "Math Tests", "Kahoot Quizzes", "PBS Video") had focusable elements that could be meaningfully interacted with through linear scanning.

Of the remaining games, 25 exposed a view hierarchy consisting entirely of a single, screen-sized target. Despite containing the entire interface for the game, these elements did not expose any interactive elements within that hierarchy. These "canvas-style" elements all behaved similarly: they allowed the screen to update rapidly with custom-drawn images and animations without needing to create a new Android View for each object. As a result, these games tended to have rich visual styles, but were entirely inaccessible to linear scanning.

The final two games (i.e., "BrainPOP Video", "Letter School") were a hybrid of the previously-stated forms: they were mostly dominated by a large canvas-style element that did not expose any of its components, but included a few additional Android views that could be directly interacted with. Despite the additional views providing some scaffolding, these games were still almost completely inaccessible when using linear scanning.

Fig. 1. Annotated screenshots of an inaccessible game interface ("Mental Math"). The left image shows the expected focusable elements: a unique target for each of the number keys, for each input slot, and for the pause button. The right shows the actual focusable elements: a single target that covers the entire screen, blocking any interaction with the underlying interactive elements.

4.2 Accessibility Scanner

Although 25/30 games consisted entirely of a single view element and were therefore completely inaccessible to linear scanning, the Accessibility Scanner did not identify accessibility issues in these games, except in cases where the canvas-style view was missing a content description. This lack of identified accessibility issues can be attributed to the Scanner's reliance on the Android view hierarchy to inform its evaluation: in the case of these apps, the only focusable element "visible" to the Scanner is the canvas-style element itself, with no regard for what is being drawn on it. However, given that "what is being drawn" is the entire game interface, any accessibility issues in the games will be missed by the Scanner.

In the five remaining games, the Accessibility Scanner did identify several key accessibility failings, such as small targets, unlabeled elements, and low-contrast text. Consistent with prior work in mobile app accessibility [10], these interfaces were mostly accessible, but contained a few small issues that can make navigating the application difficult.

4.3 Moving Targets

Eight of the 30 games contained moving targets. In some of these games, the target's motion was the primary source of difficulty for a given challenge. For example, "Pooza - Educational Puzzles" gives players bubbles to pop after successfully completing a puzzle. However, other games combined moving targets with other learning challenges. For example, "Fun Clock" combines a time-telling challenge with a dexterity challenge, requiring players to tap bouncing balls with numbers in order to make the clock show the correct time.

This combination of dexterity challenges with learning challenges creates a unique accessibility barrier: people who want to engage with learning content are not able to do so if they are blocked from engagement by an inaccessible dexterity challenge posed by a moving target.

4.4 Time Constraints

Seven of the 30 games incorporated time constraints. Within those, four involved "quick-time events" (game challenges that require an immediate reaction) and three imposed larger-scale "time limits" on certain tasks.

These time constraints serve a variety of purposes within the games. In "Mental Math", a time limit was imposed to encourage rapid mental math and create a sense of urgency in a "battle" with a monster. In "ABC Spelling - Spell & Phonics", a quick-time event to pop the correctly labeled balloon before it left the screen added an element of fast-paced excitement to a spelling challenge.

In this case, all time constraints were entangled with the learning challenges in each game. If a person wants to engage with the learning content of these games, they are also forced to engage with the dexterity challenges.

5 DISCUSSION AND FUTURE WORK

A key finding is that many popular Android educational games are almost entirely inaccessible using Android Switch Access with linear scanning. Future work can explore whether this generalizes across a larger corpus of Android educational games, but the presence of such a severe barrier in 30 of the most popular games is in itself a significant issue.

Although point scanning could provide a possible workaround for a lack of focusable elements, the presence of moving targets and time constraints makes point scanning much more difficult. For example, because a point scan is not an instantaneous action, a moving target requires a person to predict where the target *will be* upon completion of the scan, which can be an unnecessarily difficult challenge. Given the wide range of abilities of gamers, the various methods used to interact with these games, and the wide range of requirements of these games, ensuring that all interaction channels are properly accessible is crucial to making games accessible to everyone.

In order to make the wealth of existing, inaccessible apps more accessible to more people, future work could explore the space of runtime repair and enhancement for educational game interfaces. Prior work has demonstrated the potential for automated runtime tools to identify and enhance interactive elements [5] and even repair accessibility barriers in standard mobile applications [12, 13]. The unique design of educational games presents new challenges for runtime repair beyond those seen in standard mobile applications, including accounting for moving targets, identifying unexposed interactive elements, and disentangling desirable game challenges (e.g., learning tasks) from inaccessible game challenges (e.g., gratuitous dexterity challenges).

However, it is also important to ensure that future educational games are designed and implemented for accessibility from the start. Building upon existing accessible game design recommendations [4], we present two design recommendations for designers and developers of Android educational games:

(1) **Separate Dexterity and Learning Challenges.** Although dexterity challenges can be valuable, especially in early-learner apps, they can block players from being able to engage with other content in the game. When possible, do not require players to complete dexterity challenges in order to engage with other core learning content of a game. Consider implementing means for a player to disable or circumvent these challenges, such as an option to disable certain quick-time events or time limits.

(2) **Implement a Meaningful, Focusable View Hierarchy.** When possible, do not solely rely on canvas-like elements to render interactive game elements. If not possible to avoid using canvas-like elements, consider augmenting them with focusable views to serve as an accessible alternative for direct interaction with the interface.

ACKNOWLEDGMENTS

This research was partially supported by NSF #1702751 and UW CREATE (Center for Research and Education on Accessible Technology and Experiences)

REFERENCES

[1] 2021. About Switch Access for Android - Android Accessibility Help. https://support.google.com/accessibility/android/answer/6122836?hl=en

[2] 2021. Accessibility Scanner - Apps on Google Play. https://play.google.com/store/apps/details?id=com.google.android.apps.accessibility.auditor&hl=en&gl=US

[3] Anabela Araújo Rodrigues. 2016. Understanding Gesture Demands of Touchscreen Games to Accommodate Unconventional Gamers. (2016), 118.

[4] Kevin Bierre, Jonathan Chetwynd, Barrie Ellis, D Michelle, Stephanie Ludi, and Thomas Westin. 2021. Game Not Over: Accessibility Issues in Video Games. (June 2021).

[5] Morgan Dixon and James Fogarty. 2010. Prefab: implementing advanced behaviors using pixel-based reverse engineering of interface structure. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '10)*. Association for Computing Machinery, New York, NY, USA, 1525–1534. https://doi.org/10.1145/1753326.1753554

[6] Tiago Guerreiro, Hugo Nicolau, Joaquim Jorge, and Daniel Gonçalves. 2010. Towards accessible touch interfaces. In *Proceedings of the 12th international ACM SIGACCESS conference on Computers and accessibility (ASSETS '10)*. Association for Computing Machinery, New York, NY, USA, 19–26. https://doi.org/10.1145/1878803.1878809

[7] Yoojin Kim, Nita Sutreja, Jon Froehlich, and Leah Findlater. 2013. Surveying the accessibility of touchscreen games for persons with motor impairments: a preliminary analysis. In *Proceedings of the 15th International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS '13)*. Association for Computing Machinery, New York, NY, USA, 1–2. https://doi.org/10.1145/2513383.2513416

[8] John R. Porter and Julie A. Kientz. 2013. An empirical study of issues and barriers to mainstream video game accessibility. In *Proceedings of the 15th International ACM SIGACCESS Conference on Computers and Accessibility*. ACM, Bellevue Washington, 1–8. https://doi.org/10.1145/2513383.2513444

[9] Anne Spencer Ross, Xiaoyi Zhang, James Fogarty, and Jacob O. Wobbrock. 2018. Examining Image-Based Button Labeling for Accessibility in Android Apps through Large-Scale Analysis. In *Proceedings of the 20th International ACM SIGACCESS Conference on Computers and Accessibility*. ACM, Galway Ireland, 119–130. https://doi.org/10.1145/3234695.3236364

[10] Anne Spencer Ross, Xiaoyi Zhang, James Fogarty, and Jacob O. Wobbrock. 2020. An Epidemiology-inspired Large-scale Analysis of Android App Accessibility. *ACM Transactions on Accessible Computing* 13, 1 (April 2020), 1–36. https://doi.org/10.1145/3348797

[11] Bei Yuan, Eelke Folmer, and Frederick C. Harris. 2011. Game accessibility: a survey. *Universal Access in the Information Society* 10, 1 (March 2011), 81–100. https://doi.org/10.1007/s10209-010-0189-5

[12] Xiaoyi Zhang, Lilian de Greef, Amanda Swearngin, Samuel White, Kyle Murray, Lisa Yu, Qi Shan, Jeffrey Nichols, Jason Wu, Chris Fleizach, Aaron Everitt, and Jeffrey P Bigham. 2021. Screen Recognition: Creating Accessibility Metadata for Mobile Applications from Pixels. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems* (Yokohama, Japan) *(CHI '21)*. Association for Computing Machinery, New York, NY, USA, Article 275, 15 pages. https://doi.org/10.1145/3411764.3445186

[13] Xiaoyi Zhang, Anne Spencer Ross, Anat Caspi, James Fogarty, and Jacob O. Wobbrock. 2017. Interaction Proxies for Runtime Repair and Enhancement of Mobile Application Accessibility. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. ACM, Denver Colorado USA, 6024–6037. https://doi.org/10.1145/3025453.3025846