RePlay: Touchscreen Interaction Substitution Method for Accessible Gaming

Dragan Ahmetovic
dragan.ahmetovic@unimi.it
Università degli Studi di Milano

Daniele Riboli
daniele.riboli@studenti.unimi.it
Università degli Studi di Milano

Cristian Bernareggi
cristian.bernareggi@unimi.it
Università degli Studi di Milano

Sergio Mascetti
sergio.mascetti@unimi.it
Università degli Studi di Milano

ABSTRACT

We present RePlay, an interaction substitution method designed to support people with upper extremity motor impairments while interacting with mobile apps. RePlay overcomes key limitations of existing approaches, allowing users to quickly access all app interface elements, even those not visible to the OS (a common situation especially with games). This is achieved through personalized mapping of interface elements to alternative inputs, such as external switches or non-verbal vocal sounds, hence adapting to users with diverse abilities. RePlay was implemented as an Android Accessibility Service, running without any changes at OS level.

An evaluation conducted with ten participants with upper extremity motor impairments shows that RePlay can provide access to mobile games that would otherwise be inaccessible. Still, reaction time using RePlay is slower than direct touchscreen interaction by users without motor impairments, and therefore speed compensation might be needed. Both external switches and non-verbal voice input achieve comparable reaction times, while their combination is slower and more cognitively demanding. Prolonged voice input may also cause fatigue and should be used parsimoniously.

CCS CONCEPTS

• **Human-centered computing → User interface design**; **Accessibility systems and tools**; *Empirical studies in accessibility*; *Touch screens*; *Sound-based input / output*.

KEYWORDS

Upper extremity motor impairments; interaction substitution.

ACM Reference Format:
Dragan Ahmetovic, Daniele Riboli, Cristian Bernareggi, and Sergio Mascetti. 2021. RePlay: Touchscreen Interaction Substitution Method for Accessible Gaming. In *Proceedings of the 23rd International Conference on Mobile Human-Computer Interaction (MobileHCI '21), September 27-October 1, 2021, Toulouse & Virtual, France.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3447526.3472044

MobileHCI '21, September 27-October 1, 2021, Toulouse & Virtual, France

ACM ISBN 978-1-4503-8328-8/21/09…$15.00
https://doi.org/10.1145/3447526.3472044

1 INTRODUCTION

People with Upper Extremity Motor Impairments (UEMI) access mobile devices using Assistive Technologies (ATs) that replace interactions inaccessible to them with different, accessible ones. Direct touchscreen access is commonly replaced with sequential traversal of User Interface (UI) elements, until the target element is reached. This approach is slower, and therefore not suited for apps with time-constrained interaction, such as games [19]. Furthermore, when apps are developed disregarding accessibility guidelines [53] or using third party toolkits, such as cross-platform game engines, UI elements are often not visible to the OS [25, 48], and therefore cannot be traversed sequentially. As a result, many mobile apps, in particular games, are still inaccessible to people with UEMI [13].

To address this issue, we propose *RePlay* (*Re*placement interface for *Play*ing), an *interaction substitution method* that enables people with UEMI to use many currently inaccessible mobile apps, including games. *RePlay* can access all UI elements of an app, including those developed using inaccessible third-party toolkits, or without conforming to accessibility guidelines. During configuration, UI elements are manually labelled on an app screenshot. They can then be mapped to a different interface and triggered without sequential exploration, hence supporting time-constrained interaction. Various alternative interfaces can be used. Thus, *RePlay* is highly personalizable to heterogeneous user needs and changes in user abilities. For example, in Super Mario Run [9], instead of tapping the touchscreen to have Mario jump, a user who cannot perform precise hand movements can press an external button, while a user who cannot move hands can use voice input (*e.g.*, uttering "A").

We implemented *RePlay* as an Android Accessibility Service [2] (AAS), without changes at OS level, which makes it practical for accessible gaming. It is published on Google Play Store[1], and its source code is available online[2]. *RePlay* currently supports three input interfaces: external switches, a novel input method based on custom, non-verbal vocal sounds (*e.g.*, vowels or mouth clicks), and their combination. Supported touchscreen interactions include taps, prolonged taps and swipes, but others can be easily added. We performed an empirical evaluation of the interaction accuracy and reaction time using *RePlay*. For this, we developed a prototype game, and tested it with 10 participants with UEMI, and 10 users without UEMI. Participants with UEMI also tested *RePlay* with 3 popular mobile games, assessing system usability and task load.

[1] https://play.google.com/store/apps/details?id=com.carlo.a_cube
[2] https://github.com/A-CubeTest/A-Cube

All participants with UEMI were able to play the tested games with *RePlay*, and could not otherwise. They showed enthusiasm, found the system usable, and reported that they would use it frequently. Comparable reaction times were achieved with external buttons and voice input, while combined input was significantly slower. Such effect can be attributed to a heavier mental load for the combined input, measured with NASA Task Load Index (TLX) [35]. All three input methods were accurate for taps, while for prolonged taps the voice input was more physically demanding and error prone, which in part also impacts the combined input modality. Usage data also shows longer reaction times for participants with UEMI using *RePlay* than users without UEMI accessing the device through touch screen. Thus, in addition to *RePlay*, third-party tools for changing game speed [10], might be needed for inclusive and enjoyable gaming. This is consistent with prior literature suggesting speed customization for accessible gaming [39].

2 RELATED WORK

UEMI are caused by a number of different conditions, such as movement disorders (*e.g.*, cerebral palsy [56]), injuries [50] or stroke [63]. Thus, people with UEMI have heterogeneous abilities: some cannot move one or both hands, while others have difficulties in performing fine movements. UEMI can also be associated to speech impairment (*e.g.*, cerebral palsy [21], anarthria [42]). Other causes of UEMI are chronic diseases such as diabetes [57], neurodegenerative conditions (*e.g.*, parkinsons [54]) or aging [26], which are also characterized by gradual impairment worsening.

To access computers or mobile devices, people with UEMI use hardware [4] or software [47] ATs specifically tailored to each user. Such ATs may reduce the required dexterity, strength [36] or interaction complexity [71]. Furthermore, ATs for people with UEMI need to adapt not only to their heterogeneous abilities, but also to the changing needs of each person [22, 40].

2.1 Mobile accessibility for People with UEMI

Mobile devices can provide ubiquitous access to information, and act as a single interface for many daily tasks such as navigation, personal organization and smart home control [52]. Thus, their adoption by people with disabilities, including those with UEMI, is increasing [51]. However, mobile devices are accessed through touch screen interaction, which is challenging for people with UEMI [41] because it depends on precise finger movements [14].

ATs supporting mobile device access for people with UEMI can be hardware devices (*e.g.*, external buttons), software or their combination. Software solutions are usually implemented as accessibility services (ASs) [2], background apps that replace inaccessible interactions with simpler, accessible ones. These ASs provide access to UI elements through alternative pointing mechanisms, such as gaze or head tracking [46, 68], verbal instructions [44, 70], and sequential exploration with touchscreen gestures or external switches [5]. Thus, the user can select and activate any UI element detected by the OS, but the interaction is slower than direct touchscreen access [19]. For example, simulating a single tap on a UI element requires the user to first focus the target UI element by pressing several times on a physical button and then to enter the tap gesture by pressing a different physical button.

Issues may also arise when app developers do not follow accessibility guidelines [53], in which case UI elements may not be correctly accessed by the OS. In such cases, prior works propose *interaction proxies* [69], third-party add-ons specifically developed for each app to repair accessibility issues, exposing previously inaccessible UI elements to the OS. For apps developed using cross platform development toolkits (*e.g.*, *Unity* [34]), which is a common situation for games, such approach may not work because UI elements created by these toolkits may not even be visible to the OS [48]. In these cases, app developers need to re-define native components to enable accessible interaction, which requires additional effort and time, and hence it is rarely done [25]. As a result, most ASs are ineffective for such apps, as they cannot access UI elements at all.

Alternative pointing mechanisms that do not rely on the UI elements exposed to the OS, such as grid-based voice control on iOS devices [30], make it possible to aim and interact with any part of the screen. However, reaching and activating UI elements with these ASs is even slower. For example, using voice control, multiple verbal instructions are required to pinpoint a specific area of the screen, which makes this approach inappropriate for time-constrained interactions that are common in many mobile games.

2.2 Accessible Gaming for People with UEMI

Limited availability of gaming options for youth with disabilities, including those with UEMI [13], has contributed for a long time to their exclusion among their peers [65]. To improve game accessibility for people with UEMI, an ongoing effort by researchers [45] and nonprofit projects [4] is to promote the development of accessible games [67] and ATs to make existing games accessible [17]. Accessible games for people with UEMI [65] are specifically designed [66] and often present simplified interaction, commonly using a single binary input activated through external buttons [5]. Voice input has also been proposed, demonstrating similar performance to manual interaction on participants without disabilities [59]. This approach, however, depends on the recognition of pre-defined vocal sounds, which may not be easy to reproduce for users with speech impairments. While guidelines for the creation of accessible games for people with UEMI exist [39], this population is rarely considered by game developers due to development costs [15, 25]. Specialized accessible games also do not support inclusiveness because existing, popular games remain inaccessible.

ATs for accessing existing games have been proposed as hardware peripherals for computer and consoles (*e.g.*, Microsoft Xbox Adaptive Controller [7]). However, for mobile games, accessibility is still limited due to two main reasons. First, as observed above, many mobile games are developed with cross-platform developing tools, and therefore may include UI elements which are not visible by the OS and ASs. Second, even when UI elements are visible by ASs, existing ASs are still unsuitable for games that require time-constrained interaction because they rely on sequential access, which prior works report to be time consuming [55]. This is a problem also for existing alternative pointing mechanisms (*e.g.*, external joystick, gaze tracking, “vocal joystick” [20]), which were analyzed in previous works and reported to be potentially either too slow or too innacurate for game interaction [19].

3 *REPLAY*

RePlay supports input either with specialized hardware (*e.g.*, external buttons) or though software (*e.g.*, voice recognition). Hence, it is highly personalizable to different needs of people with UEMI, also adapting to changes in user abilities. Differently from previous approaches [59], *RePlay* voice input is trained on the user's own voice. This may be particularly relevant when UEMI is associated to a speech impairment (*e.g.*, in cerebral palsy [21] or anarthria [42]).

3.1 Solution design

RePlay is designed to enable people with UEMI to access games. This requires to address three main challenges:

C1. To interact with UI elements that are not visible to the OS.

C2. To allow the user to quickly interact with the UI elements.

C3. To adapt to heterogeneous and changing user abilities.

To address these challenges, during the design phase we introduced two concepts: *interaction event* and *triggering action*. An *interaction event* is a touchscreen gesture (*e.g.*, a tap at specific coordinates) received by a target app. A *triggering action*, instead, is a user action (*e.g.*, tapping on the top left touchscreen corner) that triggers a corresponding interaction event. In non-mediated interaction (*i.e.*, default touchscreen interaction), the two concepts are strongly coupled: when the user taps in the top left touchscreen corner (triggering action), a tap gesture is generated on the $\langle 0, 0 \rangle$ UI coordinates (interaction event). This is depicted in Figure 1(a). This interaction is quick, but not accessible to many users with UEMI.

Existing ATs for people with UEMI decouple interaction events and triggering actions, introducing alternative triggering actions, easier to perform for the users. Such triggering actions usually entail sequential exploration and activation of UI elements detected by the OS, which makes them ineffective for UI elements that are not visible to the OS. As one interaction event is translated to a series of triggering actions, existing ATs are also unsuitable for time-constrained apps such as games. For example, with *Switch Access*[3], the user can traverse UI elements by pressing an external physical button until the target element is reached. Then, a different external button can be used to activate the target element (see Figure 1(b)).

Our solution addresses C1 by defining interaction events in terms of screen coordinates. Thus, unlike existing solutions, *RePlay* is independent from the underlying UI structure. a similar approach for identifying UI elements, unrelated to accessibility tasks, has been proposed for automatizing desktop interaction sequences [64].

To address C2, we define triggering actions that users can do quickly. Indeed, in *RePlay*, depicted in Figure 1(c), a single triggering action (*e.g.*, pressing an external button) immediately activates the associated interaction event. This is in contrast to existing solutions that require a combination of actions to trigger an event.

To address C3, triggering actions, interaction events, and their mapping is defined for each app, through an initial configuration phase performed by the user or a caregiver. The possibility to define personalized game configurations for each user, and to modify these configurations as needed, enables *RePlay* to adapt to user heterogeneous abilities and their changes through time (**C3**).

[3] an AAS, included in the Android Accessibility Suite [5]

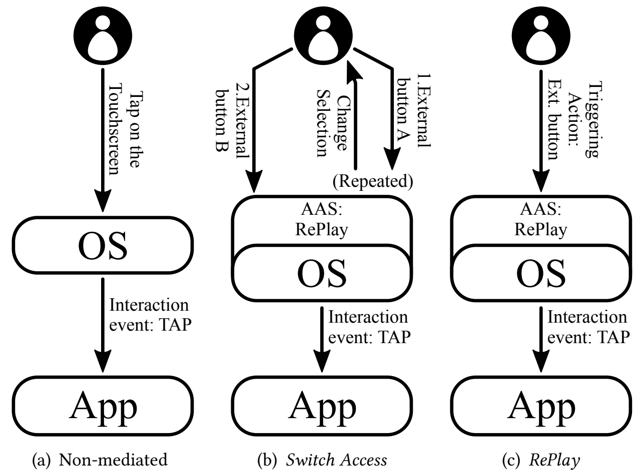


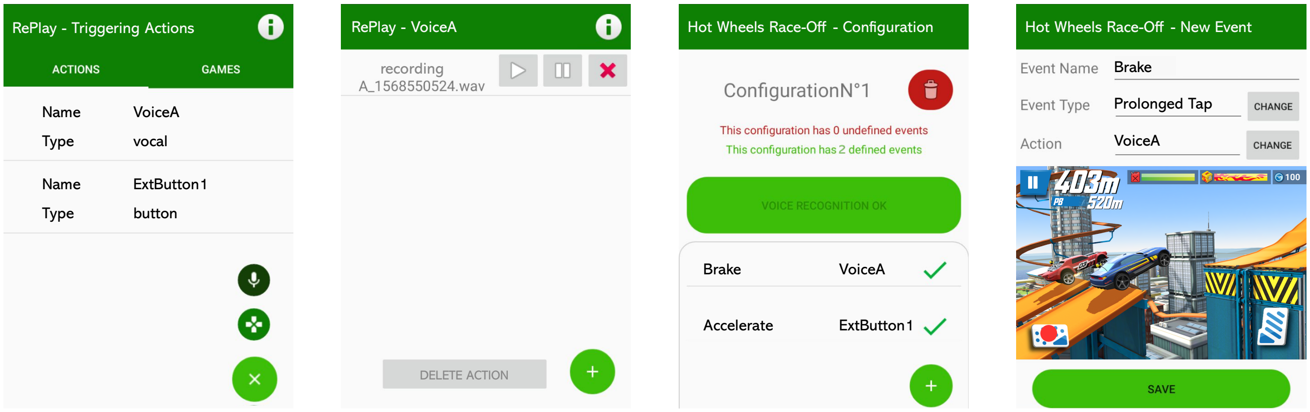
Figure 1: Non-mediated interaction, existing AAS, *RePlay*.

3.2 The *RePlay* System

The *RePlay* system is implemented for Android and consists of two components: a configuration app used to map triggering actions to interaction events for target games, and an AAS that runs in background and applies interaction substitutions when these games are played. By implementing *RePlay* interaction substitutions as an AAS [2] it is possible to generate *interaction events*, hence emulating touchscreen interaction.

Before using the *RePlay* AAS, the user needs to configure it for a target app (see Figure 2). The configuration is divided in two steps, and can be set up by users with UEMI on their own, using existing ATs such as switch access [5], or with the support of a caregiver. First, the user specifies a set of triggering actions to use for the target app (see Figure 2(a)). Currently, two types of triggering actions are supported: external switches (such as buttons or sip-and-puff interfaces), and voice input. For external switches, the user is instructed to connect them with the device and to identify the target button by pressing it. For voice input, the user records audio files containing the desired sound (see Figure 2(b)), which are then used to train the sound recognition model (described in the following section).

In the second step, the user selects the target app from those installed and creates its configuration (see Figure 2(c)). Then, the user loads a screenshot of the target app and defines the interaction events by selecting the position where an interaction event should happen, the interaction type (*e.g.*, tap) and the triggering action. In the example shown in Figure 2(d), the user associates a tap in the lower-left part of the screen (the red dot in the screenshot) with the triggering action called "VoiceA". Currently *RePlay* supports three interaction event types: instantaneous taps, prolonged taps (*i.e.*, taps that last as long as the user is pressing the switch or pronouncing the voice input), and swipes. Other, more complex gestures (*e.g.*, pinch, pan, double tap), or sequences of gestures could be easily added.



(a) Triggering actions list (b) Voice recognition training (c) Game configuration screen (d) Event position and action

Figure 2: *RePlay* configuration screens

3.3 Voice Input Module

Voice input module allows the use of non-verbal vocal sounds, such as vowels or lip sounds (*e.g.*, mouth clicks), as triggering actions. We use personalized machine learning models [12] trained to recognize voice inputs specific to each user. This approach is more flexible than a pre-trained classifier, as it allows users to define their own sounds. This is particularly important for users with speech impairments that might have difficulties with pronouncing pre-determined sounds. To train a model, a user can create voice triggering action inputs using the *RePlay* configuration app (see Figure 2(b)). For each triggering action, multiple audio clips can be recorded, each containing several samples of the corresponding voice input sound (*e.g.*, letter "A"). Since each audio clip only contains samples of one voice input sound, the extracted audio frames are automatically labelled, assigning the sound's name as the label when a sound is detected, and "no sound" label otherwise.

The extracted audio frames are used to train a machine learning classifier [37], based on a multi-class Support Vector Machine [28] with a Radial Basis Function (RBF) kernel [62], and Mel Frequency Cepstrum Coefficients (MFCC) [29] as sound features. We use *libSVM* opensource library [27]. Regularization and RBF gamma hyperparameters are set to $C = 50$ and $\gamma = 0.9$ respectively, based on preliminary tests. To account for unbalanced training data, each sample is assigned a weight inversely proportional to the number of samples in the class. The sound is acquired at 44100Hz sampling rate for high quality recognition. To achieve responsiveness suitable for gaming, real time recognition is performed on 20ms frames with 50% overlap, resulting in only 10ms overhead. For each frame we compute the first 20 MFCCs on voice frequency band (300-3000Hz), using *TarsosDSP* library [58] with filterbank size set to 32. This configuration is robust for speaker recognition [60, 61] and in our preliminary tests it achieved a recognition accuracy of 0.98, for each frame, for two class vowel recognition. Since even short vocalizations last many frames (about 100ms [38]), moving average on 3 frames is used to minimize the impact of such errors.

4 USER EVALUATION

We conducted an empirical evaluation to assess the efficacy of the *RePlay* system for enabling accessible gaming for people with UEMI, considering different input modalities. We also assessed system usability and load through a questionnaire proposed to participants with UEMI, after testing *RePlay* with 3 popular mobile games. The following research questions were addressed:

Q1. Is *RePlay* effective in supporting accessible, accurate, and time critical gaming for people with UEMI?
Q2. How do different input modalities compare among themselves and to non-mediated touchscreen gaming?
Q3. To which extent is *RePlay* perceived usable and what is the perceived task load for different input modalities?

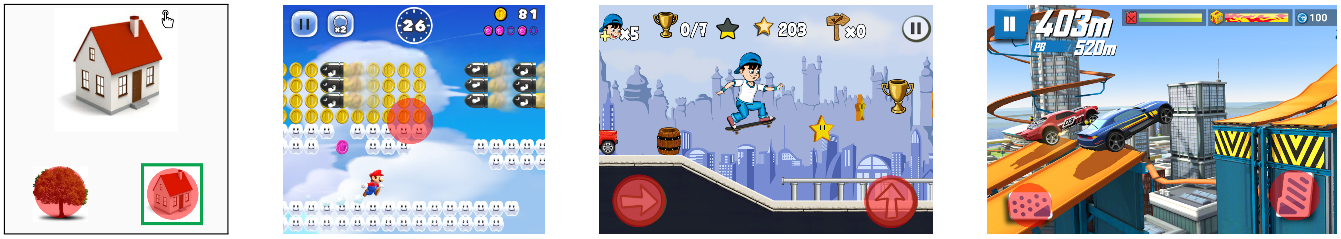
4.1 Apparatus

For experiments we used a Xiaomi Mi A2 smartphone with Android 9.0. The external button interface used was Blue2 Bluetooth Switch, with 2 70mm x 70mm buttons. It is designed for people with UEMI, to be used with Switch Access [5] and few mobile apps that support it [1]. For voice input, we used wireless headphones with a microphone close to participants' mouth for better accuracy.

To assess how different *RePlay* input modalities compare to touchscreen gaming by users without UEMI (**Q2**), we analyze gaming interaction accuracy and reaction times. To collect such data, we implemented a simple prototype game, as a web app[4], and we used it to remotely collect participants' usage data [11]:

Press the Button The game is organized in three levels of 12 rounds each. Each round a tree or house is shown on the top of the screen and the player has to select the corresponding button among the two shown at the bottom (see Figure 3(a)). In the first level an instantaneous tap is needed, while in the second and third levels, prolonged taps are required, of 2s and 3s respectively. Visual feedback shows if the button is correct and if it has been pressed for sufficient time. After completing a round, a new one starts after 1s.

[4] https://touch-game.netlify.app/



(a) Press the Button (b) Super Mario Run (c) Skater Kid (d) Hot-Wheels Race Off

Figure 3: Games used for the study. Red marks show tap positions

In addition, to assess whether *RePlay* is effective in supporting accessible gaming with existing games (**Q1**), we tested it with 3 among the most popular casual games[5] that are not accessible through existing mobile ATs. These games require time-constrained interaction through either immediate or prolonged taps:

> **Super Mario Run [9]** uses prolonged one tap interaction to control character's jumps. Longer interaction corresponding to longer and taller jumps (see Figure 3(b)).
> **Skater Kid [8]** has a slower interaction speed than Super Mario Run, but two instantaneous interaction events, one for accelerating and another for jumping (Figure 3(c)).
> **Hot-Wheels Race Off [3]** has an interaction speed comparable to Super Mario Run, and two prolonged taps, used to accelerate and brake (Figure 3(d)).

4.2 Participants

We recruited 10 participants with UEMI, all of age and without sensory or cognitive impairments. Most were male (8), consistently with demographic distribution of motor impairments [49]. They had a range of different conditions (see Table 1), four since birth, three between 5 and 10 years, and three for 5 years or less. Most used a smartphone, but only P2, who had the highest self-reported difficulty in performing precise touchscreen interactions, used an AT. Many had difficulties (measured on a Likert scale, where 1="low", 7="high") during prolonged interactions (P2, P4, P5, P8), and more with one arm than with the other (P1, P3, P5, P6, P7, P10).

PC and console games were popular. P2 and P4 play sport games, P8 first person shooters, and P1 and P7 logic and card games. P1, P4 and P5 found PC or console games with rapid interaction inaccessible, while games with many buttons or complex interactions were difficult for P1, P2 and P4. As reported by P2 and P10, mobile games, including those used for the study, are too complex. Only P6, who sometimes plays simple mobile sport games, was able to interact with Super Mario Run, since it has a single input.

We also collected interaction logs of 10 users without UEMI playing the "Press the Button" game with touchscreen, as a control group, in order to understand whether assisted interaction using *RePlay* ensures a similar gaming experience for participants with UEMI as touchscreen gaming for users without UEMI. For this, we made the game publicly available and we publicised it with our friends and co-workers, remotely collecting anonymous usage logs.

4.3 Study Design

The study design refers to reaction time and interaction accuracy evaluation with "Press the Button" game. Participants with UEMI chose a suitable input modality (independent variable) based on their ability and preference. Since some participants were not able to use all input modalities, within-groups testing was not possible and therefore between-groups design was chosen.

Possible choices for input modality were **Voice**, **Button** or **Combined** (one button and one voice input). Instead, all users without UEMI used **Non-mediated** touchscreen input (control condition). Specifically, four participants with UEMI used Button interaction (P4, P5, P6, P10). Those who had more difficulties in interacting manually than by using voice (P2, P7) selected Voice input. P1, who had difficulties articulating words, insisted to try voice input and therefore was also assigned this condition. Participants with similar ability in interacting with voice and manually (P3, P8, P9) selected Combined input. In particular, P8 had no difficulty in interacting manually or through voice input for short time, but could not withstand prolonged interaction with either. These settings were also used for interacting with the three existing games. In particular, for Super Mario Run, which only has one event, participants who selected Combined input used one of their input modalities (P3 & P8 voice, P9 button).

As dependent variables, we considered a) the interaction reaction time, that is the time between visual stimulus an the activation of the relevant input, and b) the outcome for each round played. For the rounds completed correctly the outcome is "Successful". Instead, the errors were categorized in "Selection errors" if the participant selected the wrong input, and "Duration Errors" if the participant did not maintain the interaction for the sufficient amount of time. If the participant could not proceed due to fatigue, the remaining rounds in the level were all categorized as "Not Completed".

Statistical testing of reaction time differences between conditions was conducted using Kruskal-Wallis, a non-parametric test appropriate for assessing whether multiple samples originate from the same distribution. Post-hoc analysis used Dunn's Test with Benjamini-Hochberg adjustment [31]. Instead, for analyzing differences in round outcomes between conditions, Fisher's Exact Test (with Bonferroni correction in case of pairwise comparisons) was used because some outcomes had low frequency counts [23].

[5] bit.ly/androidRanks

Table 1: Participants' demographic information. A: touchscreen use; B: articulating words and vocalizing; 1: brief; 2: prolonged

ID	Sex	Age	Motor Impairment		Difficulty in				Smartphone		Games		*RePlay*
			Condition	Since	A1	A2	B1	B2	Use	AT	Pc/Console	Mobile	Interaction
P1	M	18-27	Anarthria	10y	3	4	6	6	Yes	No	Simple	No	Voice
P2	M	18-27	Duchenne dystrophy	8y	7	7	1	5	Yes	Google Asst.	Sport	No	Voice
P3	F	18-27	Hemiparesis	birth	2	2	2	2	No	-	No	No	Combined
P4	M	18-27	Pelizaeus-Merzbacher	birth	1	6	1	7	No	-	Race	No	Button
P5	F	18-27	Spastic tetraplegia	birth	1	6	1	7	Yes	No	No	No	Button
P6	M	58-67	Hemiplegia	5y	1	1	2	6	Yes	No	No	Sport (rare)	Button
P7	M	48-57	Spastic tetraplegia	birth	2	4	1	1	No	-	Logic	No	Voice
P8	M	48-57	Huntington's disease	10y	3	5	2	5	Yes	No	FPS	No	Combined
P9	M	38-47	Aneurysm	1y	2	3	3	3	Yes	No	No	No	Combined
P10	M	38-47	Hemiparesis	3y	2	5	2	4	Yes	No	No	No	Button

4.4 Protocol

The study, approved by our university's ethics committee, was performed at a place of participants' convenience, in an empty room to minimize distractions. After introducing our research, we acquired study consent and collected participants' demographic information (see Table 1). The games were configured based on the input modality selected by the participants (see above). Those who selected voice or combined input also recorded the sounds to use as triggering actions. The sounds were chosen to be easy to distinguish, and for the participant to vocalize repeatedly and continuously. All participants chose vowels or vowel-like sounds, for example (using international phonetic notation [16]) [a] (as in "hat" [hat]) and [e] (as in "may" [me:]). The app recorded the participants vocalizing each sound for about 10s and trained the machine learning model on the recorded audio tracks.

As the first task, participants played "Press the Button". The sequence of the presented objects was random and same for all the participants to minimize effects of order. Each round we logged the outcome and the reaction time (dependent variables). Afterwards, participants freely tested *RePlay* with the three existing games, for 5 minutes each. After each test, we collected responses to the NASA-TLX [35] questionnaire to measure the perceived task load for each game.

After the tests we also assessed the participants' feedback on the overall system usability (**Q3**) using the System Usability Scale (SUS) questionnaire [24]. We did not collect quotes as this would have resulted in under-representing participants with verbal communication difficulties. However, we did collect participants' comments. These were given during demographic questionnaire and related to their lack of prior gaming experiences (*e.g.*, P1: *"I never play mobile games because I need time to touch the screen precisely"*). In total, each study lasted between 30 minutes and one hour.

5 RESULTS

As a key result, we report that, with *RePlay*, all participants were able to play the selected games with their chosen input modalities. Additionally, we analyze reaction times and round outcomes based on the data collected with "Press the Button" game, comparing different input modalities and non-mediated interaction by users without UEMI. We then report subjective feedback results based on participants' responses to NASA-TLX and SUS questionnaires.

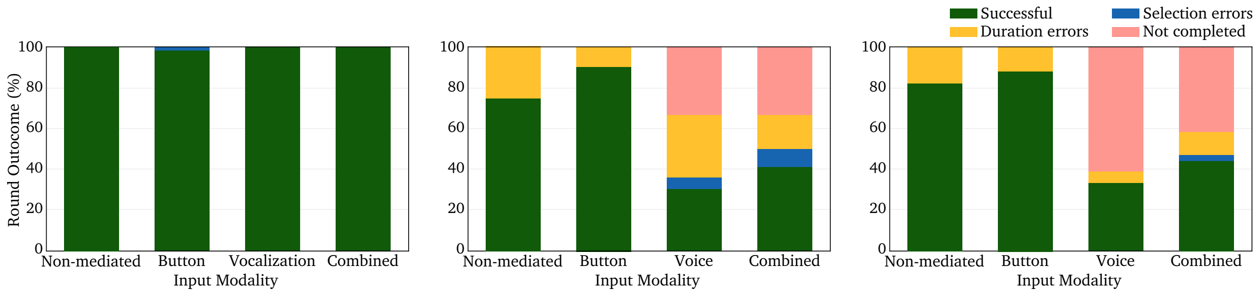
5.1 Interaction Accuracy

Interaction accuracy was evaluated as round outcome ratios for each input modality, separated by level to measure the effect of instantaneous and prolonged taps (see Figure 4). We collected $N = 720$ outcomes (20 users × 12 rounds × 3 levels). For instantaneous taps, all three input modalities and non-mediated input had near perfect scores (see Figure 4(a)). Only one participant with Button input performed one "Selection error" (1/48, 2.08%). There were no statistically significant differences between the conditions. Instead, for prolonged taps, there were stark differences across input modalities, in both second (see Figure 4(b)) and third (see Figure 4(c)) level ($p < .01$). In both cases, Non-mediated and Button input had significantly lower error rates than Voice and Combined input ($p < .01/4$). Specifically, in the second level, there were no "Selection errors" or "Not completed" rounds for Button and Non-mediated conditions. Instead, prolonged input resulted in 5/48 "Duration errors" (10.42%) for Button and 28/120 (23.3%) for Non-mediated condition. For Voice input, there were 2/36 "Selection errors" (5.56%), 11/36 "Duration errors" (30.56%) and 12/36 "Not Completed" rounds (33.33%). Only 11/36 rounds (30.56%) were "Successful". Combined input followed a similar trend: there were 3/36 "Selection errors" (8.33%), 6/36 "Duration errors" (16.67%), 12/36 "Not Completed" rounds (33.33%), and only 14/36 rounds were "Successful" (41.67%). In the third level, Button input had 6/48 while Non-mediated condition had 21 (17.5%) "Duration errors" (12.5%). Again, there were no "Selection errors" or "Not completed" rounds. Voice input had no "Selection errors", while "Duration errors" were 2 (5.56%). "Not Completed" rounds were 22 (61.11%), and "Successful" rounds were 12 (33.33%). For Combined input, there were 4 "Duration errors" (11.11%) and 1 "Selection error" (2.8%). "Not Completed" rounds were 15 (41.67%) and "Successful" rounds 16 (44.44%).

5.2 Reaction Time

Participants' reaction time was measured on all the completed rounds ($N = 659$), separated by Button ($n = 144$), Verbal ($n = 74$), Combined ($n = 81$) and Non-mediated ($n = 360$) input modality. Specifically, it was $1.72s \pm 0.99s$[6] for Button input, $1.68s \pm 0.83s$ for Voice input, and $2.92s \pm 1.59s$ for Combined input. Instead, users without UEMI had a reaction time of $1.08s \pm 0.74s$ (see Figure 5) using Non-mediated touchscreen access.

[6] We will use *Mean* ± *Standard Deviation* notation



(a) Instantaneous interaction (b) Prolonged interaction (2s) (c) Prolonged interaction (3s)

Figure 4: Interaction outcomes by level and input modality

Differences among modalities ware found to be statistically significant ($\chi^2(3, N = 659) = 190.74$, $p < .01$). Post-hoc analysis revealed that non-mediated input was faster than all *RePlay* input modalities ($p < .01$). Additionally, the combined input resulted slower than both button and voice input ($p < .01$), while the difference between button and voice input was not significant.

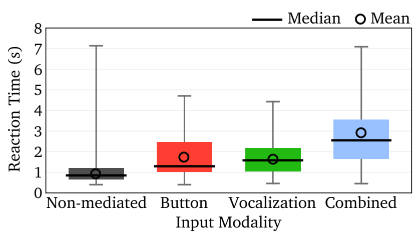


Figure 5: Reaction times for the first level by input modality

5.3 Task Load Evaluation

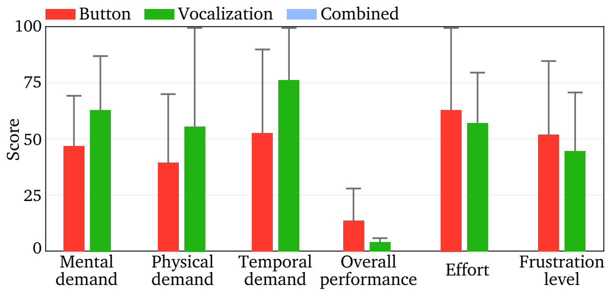
To assess the task load associated to using *RePlay* with different input modalities, we analyze participants' responses to the NASA-TLX questionnaire for each game. Due to a low number of participants for each condition, we did not perform statistical significance analysis on this data. Also, we highlight that P1 became tired and did not conduct tests with Skater Kid and Hot-Wheels Race Off.

5.3.1 Press the Button. Raw NASA-TLX [35] score for the game (see Figure 6(d)) was 35.63 (lower is better) for Button (25^{th} percentile range for gaming apps [32]), 47.78 for Voice and 56.67 for Combined input, (both 50^{th} percentile). Mental demand for Voice input (50 ± 21.79) was perceived to be lower than for other games with prolonged taps. However, Voice input was still considered more Physically (53.33 ± 20.21) and Temporally demanding (58.33 ± 7.64), and required higher Effort (61.67 ± 23.63) than Button input (31.25 ± 23.58, 28.75 ± 14.93, and 41.25 ± 10.31 respectively). Combined input was considered demanding Mentally (68.33 ± 16.07) and Temporally (68.33 ± 16.07), but had a lower perceived Effort (55 ± 18.03) with respect to Voice (61.67 ± 23.63). It also had a worse perceived Performance score (41.67 ± 20.82) with respect to Button (21.25 ± 21.75) and Voice input (18.33 ± 10.41). This is consistent with our findings from the reaction time data analysis which show how Combined input is slower than the other two input modalities.

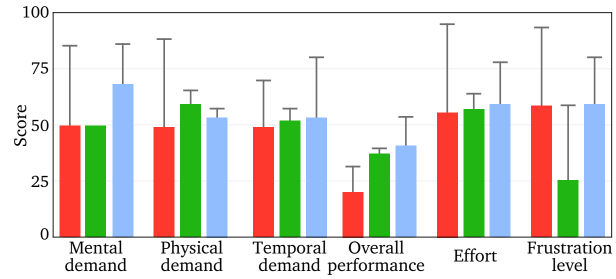
5.3.2 Super Mario Run. Since the game has one interaction event, we evaluated only Button and Voice input (see Figure 6(a)). Button input achieved a TLX score of 44.58, within 25^{th} percentile, while Voice input scored 50.28 (50^{th} percentile). Due to the use of prolonged taps, Voice interaction seemed to be harder. This was reflected by higher Mental (66.67 ± 20.82), Physical (55 ± 48.22) and Temporal (76.67 ± 23.09) demand scores with respect to Button input (46.25 ± 21.36, 38.75 ± 28.1 and 53.75 ± 32.5 respectively). However, the perceived Performance score (3.33 ± 2.89), Effort (60 ± 17.32) and Frustration level (40 ± 31.22) were better than for Button interaction (12.5 ± 14.43, 63.75 ± 34.73 and 52.5 ± 33.04 respectively).

5.3.3 Skater Kid. Button input reached a TLX score of 47.5, Voice input 47.08, and Combined input 56.11, all within 50^{th} percentile for gaming apps (see Figure 6(b)). Due to instantaneous taps and slower pace than in Super Mario Run, Voice input was less Mentally (50 ± 0) and Temporally (52.5 ± 3.54) demanding, and comparable to Button input (50 ± 33.91 and 48.75 ± 40.08 respectively). Voice input Performance was perceived to be worse (37.5 ± 3.54), but the Frustration level was also lower (25 ± 35.36) than in Super Mario Run. Instead, for Combined input, Mental (68.33 ± 16.07) and Temporal demand (53.33 ± 5.77) were perceived to be higher than for the other two modalities, but Physical demand was lower (53.33 ± 5.77) than for Voice input (60 ± 7.07). This confirms the findings of the Reaction Time analysis, which show that Combined input is slower than the other two modalities, but less physically demanding than Voice.

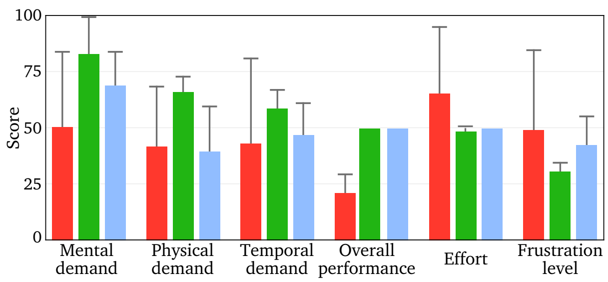
5.3.4 Hot-Wheels Race Off. Button input (see Figure 6(c)) had a TLX score of 44.17 (25^{th} percentile), while Voice and Combined input scored 55.42 and 48.89 respectively (50^{th} percentile). As in Skater Kid, due to prolonged taps, Voice input was perceived more difficult considering Mental (82.5 ± 24.75), Physical (65 ± 7.07) and Temporal demand (57.5 ± 10.61), than Button input (50 ± 33.91, 40 ± 28.58 and 43.75 ± 37.72 respectively). However, Combined input was not more Mentally (68.33 ± 16.07) or Temporally (45 ± 18.03) demanding than Voice, and it was also less Physically demanding than the other two modalities (38.33 ± 24.66). Perceived Performance was also worse for Voice (50 ± 0) and Combined input (50 ± 0) than for Button input (20 ± 8.16). However, the perceived Effort was also lower for Voice (47.5 ± 3.54) and Combined (50 ± 0) input than for Button (63.75 ± 33.01) input.



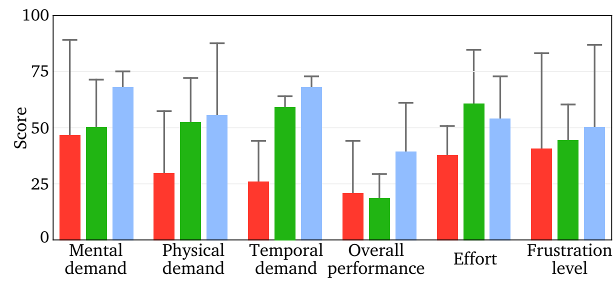
(a) Super Mario Run



(b) Skater Kid



(c) Hot-Wheels Race Off



(d) Press the Button

Figure 6: NASA TLX scores boxplots by question (full text of questions available in the original paper [35])

5.4 System Usability Scale Evaluation

We assess the perceived usability of the *RePlay* system, by input modality, using the SUS questionnaire (see Figure 7). As for NASA-TLX, we omit statistical significance analysis due to the low number of data points. Instead, we grade the overall score [18] and item-level scores [43], based on prior literature, in order to assess which specific items influence the overall score. In general, Button input yielded a higher SUS score (73.12, grade B) with respect to Voice (65.83) and Combined (66.67) input (both grade C).

Considering single SUS items, starting from Question 1, we notice that for all input modalities the participants reported that they Would use frequently the system. For Button (4.25 ± 0.5) and Combined (4.33 ± 0.58) input, the grade is considered Good, while for Voice input (3.67 ± 0.58) it is Above Average. Voice (3.33 ± 1.15) and Combined (3.33 ± 1.15) input were perceived to be Unnecessarily complex (Question 2) Indeed, both were worse than the average. Instead, Button input (1.75 ± 0.5) achieved a Good score, indicating low complexity. All input modalities were considered Above average in terms of Ease of use (Question 3). Specifically, Button input result is 4 ± 0 while Voice and Combined input scored 3.67 ± 0.58 and 3.67 ± 1.53 respectively. However, for all three modalities participants felt that they would Need technical support (Question 4) to set it up (Button 2.5 ± 1, Voice 3.33 ± 0.58, Combined 2 ± 0). Both items scored Under average for all input modalities. This was actually expected since *RePlay* requires an initial configuration, which was conducted with the support of an assistant. Regarding Question 5, participants felt that Voice input was not as Well integrated (3.67 ± 2.31, Above average) as Button (4.25 ± 0.5) and Combined (4.33 ± 0.58) input (both Good),

Considering Question 6, Combined interaction was deemed Too much inconsistent (2.67 ± 1.15, Under average) with respect to Button (1.75 ± 0.5, Good) and Voice (2 ± 1.73, Above average) input. Nonetheless, participants also considered the system Quick to learn (Question 7) for Button input (4.25 ± 0.5, Good), Above average for Combined input (4 ± 0) and slightly Under average for Voice input (3.67 ± 1.53). All three modalities were not considered Cumbersome to use (Question 8). Button (1.75 ± 0.5) and Combined (2 ± 0) input ranked Above average, while Voice input (1.67 ± 0.58) was graded as Good. Participants Felt confident (Question 9) in interacting with the system through Voice input (4.67 ± 0.58, Good), but not as much with Button (3.25 ± 1.5) and Combined input (3.67 ± 0.58) which were both Under average. However, they also felt that they Needed to learn a lot (Question 10) to use the system (Button 3 ± 1.41, Voice 2.67 ± 1.53, Combined 3.33 ± 1.15).

6 DISCUSSION AND LIMITATIONS

The key result of our research is that all the participants, using the *RePlay* system, were able to play with the selected games. This was not previously possible for them, neither using non-mediated touchscreen interaction or existing ATs. Each participant was able to select their preferred input modality among voice, button and combined input, and to access the game with the selected modality, hence confirming that *RePlay* is capable of adapting to heterogeneous needs and preferences of different users. Even *P*1, who selected voice input despite having a speech impairment, was able to access all game elements, thanks to the personalized voice recognition model, and only had problems to vocalize sounds during prolonged interactions, as other participants.

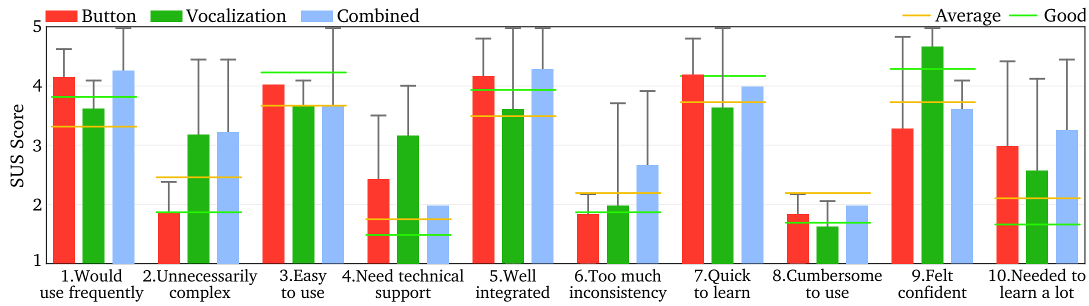


Figure 7: SUS boxplots by question (full text in the original paper [24], odd questions are better higher, even questions lower).

6.1 System Usability and Task Load

Participants were thrilled to be able to play mobile games, in particular using voice. Some enjoyed so much using the system that they asked a demonstration of different input modalities. Actually, we observed that the excitement procured some fatigue to the participants which may have contributed to the high rate of "Not Completed" rounds in the vocal and combined interaction with the system. As a future work, we will consider a longitudinal study also to offset the impact of this effect on the interaction.

The system was found to be usable considering all three input modalities. While Button interaction appears to be overall more usable than the other two, each modality had scores higher than the baseline for some items and lower for others. Thus, there was no clear consensus in terms of preferences for all SUS questions. Indeed, participants often stated that they would use the system frequently for all three modalities. Nonetheless, we also discovered that combining vocal and manual interaction significantly increases reaction time. We attribute this effect to the higher cognitive load associated to the combined interaction, measured with NASA-TLX. It will be interesting to study, as a future work, whether this effect can be mitigated with longer training. The perceived task load in playing the considered games with *RePlay* is within the average for gaming apps [32] and consistent among the four tested games. For button input the task load is actually perceived within the lowest 25%. These results show that, with *RePlay*, the task load associated to gaming experience for users with UEMI is actually comparable to the task load perceived by the general population in using games.

6.2 Implications for Design

For users without disabilities, voice and manual input were shown to achieve comparable reaction times in gaming [59]. We confirm this for participants with UEMI, who achieved comparable reaction times using button and voice interaction. However, despite using *RePlay*, participants with UEMI had overall slower reaction times than users without UEMI. Thus, *RePlay* may not be sufficient for people with UEMI to achieve similar gaming performance, and therefore playability, as users without disabilities. A design implication of this finding is that game speed adaptation should be provided to ensure accessible mobile gaming for people with UEMI. For specifically developed games, this functionality can be integrated within the game itself, as previous development guidelines note [39]. Instead, using *RePlay*, it is possible to apply existing solutions [10] that allow to change execution speed of mobile apps.

In terms of interaction accuracy, considering instantaneous taps, button, vocal, and combined input, people with UEMI achieve similar results to non-mediated touchscreen input by users without UEMI. For prolonged taps, button input by people with UEMI maintains similar performance as non-mediated input by users without disabilities. However, we detect a greater number of duration errors and not completed rounds for voice input, and to a lesser degree for combined input, due to the fatigue caused by prolonged vocal interaction for people with UEMI. These errors are greater for 3s interaction than for 2s interaction, showing that the fatigue accrues with the greater duration. Therefore, prolonged vocal interactions needs to be limited in number and duration. A possible design guideline to address this problem could be producing prolonged interactions with instantaneous triggering actions. One solution could be to have one triggering action that initiates the prolonged tap, and another one that stops it. In some case this implies to double the number of triggering actions required to play a game, however there are case in which the same triggering action can be associated to two or more interaction events.

SUS evaluation highlighted that button interaction was considered less complex and more consistent than the other input modalities. Instead, Voice input improved user confidence with the system and was perceived to be less cumbersome to use. This further supports our belief that user preferences and needs are heterogeneous. Therefore, we argue that a high level of personalization is required in order to better adapt to different games and diverse user needs and abilities. Based on these results, we suggest that accessible mobile gaming for people with UEMI should suport multimodal input and diversified interfaces such as external peripherals, voice input, body movements or visually detected facial gestures.

6.3 System limitations

While *RePlay* enables people with UEMI to use many otherwise inaccessible mobile apps, it needs to be configured first. In our study, the setup was done by a supervisor (not by the user themselves) and took about 5 minutes per user. However, in real world usage, the operation can be even faster. Indeed, each user can specify triggering actions only once and re-use them for different apps. Similarly, interaction events could be specified only once for a game, by the developers or through crowdsourcing [33], and afterwards they could be shared among users. So, to configure a new game, the user would only need to specify the association between the interaction events and the triggering actions, which takes few seconds.

Another intrinsic limitation of *RePlay* is that it can guarantee accessibility only to those games where interaction events occur in pre-defined positions. Accessible games are therefore those with a virtual controller or with onscreen buttons (like those shown in Figures 3(d) and 3(c)). Also, many games allow the user to interact through gestures that can be done anywhere on the screen (like Super Mario Run, see Figure 3(b)); these games are also accessible with *RePlay*. While these games are among the most popular ones[5], there are other games in which the user needs to interact with objects whose position changes dynamically (*e.g.*, "Fruit Ninja" [6]) and these games are not accessible with *RePlay*.

Finally, the current version of *RePlay* supports limited interaction events and triggering actions. Specifically, three types of interactions events are available: instantaneous taps, prolonged taps, and swipes. Similarly, only two types of triggering actions: external button and voice input. Both limitations will be addressed as future work, increasing possible touchscreen interaction events and adding other possible triggering actions.

7 CONCLUSIONS AND FUTURE WORK

We present a novel approach to enable people with UEMI to use mobile apps, and in particular those that require time-constrained interactions, such as games. The design, based on the mapping between triggering actions and interaction events, allows each user to personalize the interaction based on their unique abilities. The proposed approach does not require to be aware of the apps' UI structure, hence overcoming a common technical problem that causes existing ATs to be ineffective with most games.

The proposed approach can also be useful to people without disabilities. Indeed some mobile games allow the use of an external controller to provide an easier interaction for all. However, many games do not provide this possibility and the proposed approach can overcome this limitation. This is particularly relevant in the perspective of distributing the solution to the general public: indeed it is easier to sustain a project that is aimed at a larger market [11].

The *RePlay* system implements the proposed approach, showing its technical feasibility and making it possible to experimentally evaluate the system with people with UEMI. Its evaluation shows that the system is capable of providing access to existing mobile games, without the need to develop solutions for each game specifically, a result that cannot be achieved with other existing solutions.

The experimental evaluation also highlights some limitations. Indeed, the reaction times of participants with UEMI using *RePlay* are slower than those of users without disabilities through non mediated touchscreen interaction. To address this issue, existing third-party apps capable of adapting game speed [10] can be used. Prolonged vocal actions were also found to have low accuracy and cause fatigue. This problem can be addressed by substituting prolonged interaction events (*e.g.*, prolonged tap) with a starting and an ending instantaneous event (*e.g.*, start and end tap).

In addition to the currently implemented interaction events, we aim to increase possible interactions. Thus, we will add additional gestures, like pinch and pan, as well as gestures with multiple fingers, which are even harder to perform for many users with UEMI. Furthermore, we will implement personalized "macros", that is combinations of gestures that the users themselves can record [55].

Similarly, we will investigate additional triggering actions, such as body movements (*e.g.*, head gestures, eyes blinking), acquired with the device camera, through inertial sensors (*e.g.*, head mounted accelerometer), or with neural control interfaces. Furthermore, we aim to improve the existing voice input modality by researching better recognition models to further improve voice recognition accuracy. We are also investigating the applicability of the voice recognition approach for speech therapy, and gesture recognition for physical rehabilitation.

To better understand how the use of different interaction modalities evolves over time, as future work we will collect remote usage data from end-users over long periods. This is possible because the app has already been published on Google Play Store[1]. The analysis of this data will provide insights on how the solution is adopted by end-users. Along with the data collected in this study, further analysis of remote usage data will help us improve specific interaction modalities according to user needs. For example, our study shows that, for voice and combined interaction modalities, the perceived complexity is high, while perceived confidence is low for button and combined interactions. Based on this feedback, we will work on streamlining the voice training procedure and providing additional visual feedback for button interactions.

ACKNOWLEDGMENTS

We would like to thank our study participants from *Spazio Vita* of the *Niguarda Hospital* in Milan, Italy.

REFERENCES

[1] 2013. Jane Farrall Consulting list of switch-accessible iOS app. http://www.janefarrall.com/html/resources/Switch_Accessible_Apps_for_iPad.pdf
[2] 2019. Android Accessibility Services Guide. https://developer.android.com/guide/topics/ui/accessibility/service
[3] 2019. Hot Wheels Race-off. https://www.hutch.io/our-games/hot-wheels-race-off/
[4] 2021. The AbleGamers Charity. https://ablegamers.org/
[5] 2021. Android Accessibility Suite (including Switch Access). https://play.google.com/store/apps/details?id=com.google.android.marvin.talkback
[6] 2021. Fruit Ninja on Android Play Store. https://play.google.com/store/apps/details?id=com.halfbrick.fruitninjafree
[7] 2021. Microsoft Xbox Adaptive Controller. https://www.xbox.com/it-IT/xbox-one/accessories/controllers/xbox-adaptive-controller
[8] 2021. Skater Kid on Android Play Store. https://play.google.com/store/apps/details?id=com.renderedideas.skaterkid
[9] 2021. Super Mario Run on Android Play Store. https://play.google.com/store/apps/details?id=com.nintendo.zara
[10] 2021. X8 speeder, games speed hack for android. https://x8speeder.com/
[11] Dragan Ahmetovic, Cristian Bernareggi, Mattia Ducci, Andrea Gerino, and Sergio Mascetti. 2021. Remote Usage Data Collection and Analysis for Mobile Accessibility Applications. In *International Conference on Pervasive Computing and Communications Workshops - Mobile and Pervasive Assistive Technologies*. IEEE.
[12] Dragan Ahmetovic, Daisuke Sato, Uran Oh, Tatsuya Ishihara, Kris Kitani, and Chieko Asakawa. 2020. ReCog: Supporting Blind People in Recognizing Personal Objects. In *Human Factors in Computing Systems*. ACM.
[13] Meryl Alper. 2014. *Digital Youth with Disabilities*. MIT Press.
[14] Lisa Anthony, YooJin Kim, and Leah Findlater. 2013. Analyzing user-generated youtube videos to understand touchscreen use by people with motor impairments. In *Conference on Human Factors in Computing Systems*. ACM.
[15] Dominique Archambault, Thomas Gaudy, Klaus Miesenberger, Stéphane Natkin, and Rolland Ossmann. 2008. Towards generalised accessibility of computer games. In *International Conference on Technologies for E-Learning and Digital Entertainment*. Springer.
[16] International Phonetic Association, International Phonetic Association Staff, et al. 1999. *Handbook of the International Phonetic Association: A guide to the use of the International Phonetic Alphabet*. Cambridge University Press.
[17] Matthew T Atkinson, Sabahattin Gucukoglu, Colin HC Machin, and Adrian E Lawrence. 2006. Making the mainstream accessible: redefining the game. In *ACM SIGGRAPH symposium on Videogames*. ACM.

[18] Aaron Bangor, Philip Kortum, and James Miller. 2009. Determining what individual SUS scores mean: Adding an adjective rating scale. *Journal of usability studies* (2009).
[19] Kevin Bierre, Jonathan Chetwynd, Barrie Ellis, D Michelle Hinn, Stephanie Ludi, and Thomas Westin. 2005. Game not over: Accessibility issues in video games. In *International Conference on Universal Access in Human-Computer Interaction*.
[20] Jeff A Bilmes, Xiao Li, Jonathan Malkin, Kelley Kilanski, Richard Wright, Katrin Kirchhoff, Amarnag Subramanya, Susumu Harada, James A Landay, Patricia Dowden, et al. 2005. The Vocal Joystick: A voice-based human-computer interface for individuals with motor impairments. In *Conference on Human Language Technology and Empirical Methods in Natural Language Processing*. ACL.
[21] Dorothy VM Bishop, B Byers Brown, and J Robson. 1990. The relationship between phoneme discrimination, speech production, and language comprehension in cerebral-palsied individuals. *Journal of Speech, Language, and Hearing Research* (1990).
[22] Pradipta Biswas and Patrick Langdon. 2012. Developing multimodal adaptation algorithm for mobility impaired users by evaluating their hand strength. *International Journal of Human-Computer Interaction* (2012).
[23] Keith M Bower. 2003. When to use FisherâĂŹs exact test. In *Six Sigma Forum Magazine*. American Society for Quality.
[24] John Brooke et al. 1996. SUS-A quick and dirty usability scale. *Usability evaluation in industry* (1996).
[25] Niccolò Cantù, Mattia Ducci, Dragan Ahmetovic, Cristian Bernareggi, and Sergio Mascetti. 2018. MathMelodies 2: a Mobile Assistive Application for People with Visual Impairments Developed with React Native. In *International ACM SIGACCESS Conference on Computers and Accessibility*. ACM.
[26] Eli Carmeli, Hagar Patish, and Raymond Coleman. 2003. The aging hand. *The Journals of Gerontology Series A: Biological Sciences and Medical Sciences* (2003).
[27] Chih-Chung Chang and Chih-Jen Lin. 2011. LIBSVM: A library for support vector machines. *ACM transactions on intelligent systems and technology (TIST)* (2011).
[28] Philip Clarkson and Pedro J Moreno. 1999. On the use of support vector machines for phonetic classification. In *International Conference on Acoustics, Speech, and Signal Processing*. IEEE.
[29] Steven Davis and Paul Mermelstein. 1980. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE transactions on acoustics, speech, and signal processing* (1980).
[30] Daniel Derksen-Staats. 2019. Accessibility features in iOS. In *Developing Accessible iOS Apps*. Springer.
[31] Alexis Dinno. 2015. Nonparametric pairwise multiple comparisons in independent groups using Dunn's test. *The Stata Journal* (2015).
[32] Rebecca A Grier. 2015. How high is high? A meta-analysis of NASA-TLX global workload scores. In *Human Factors and Ergonomics Society Annual Meeting*. SAGE.
[33] Anhong Guo, Xiang'Anthony' Chen, Haoran Qi, Samuel White, Suman Ghosh, Chieko Asakawa, and Jeffrey P Bigham. 2016. Vizlens: A robust and interactive screen reader for interfaces in the real world. In *Annual Symposium on User Interface Software and Technology*. ACM.
[34] John K Haas. 2014. A history of the Unity game engine. (2014).
[35] Sandra G Hart. 2006. NASA-task load index (NASA-TLX); 20 years later. In *Proceedings of the human factors and ergonomics society annual meeting*. Sage.
[36] Kimberly Hreha and Lauren Snowdon. 2011. We all can call: Enhancing accessible cell phone usage for clients with spinal cord injury. *Assistive Technology®* (2011).
[37] Chadawan Ittichaichareon, Siwat Suksri, and Thaweesak Yingthawornsuk. 2012. Speech recognition using MFCC. In *International Conference on Computer Graphics, Simulation and Modeling*. PSRC.
[38] Ewa Jacewicz, Robert A Fox, and Joseph Salmons. 2007. Vowel duration in three American English dialects. *American Speech* (2007).
[39] Angel Jaramillo-Alcázar, Luis Salvador-Ullauri, and Sergio Luján-Mora. 2017. A mobile serious games assessment tool for people with motor impairments. In *International Conference on Education Technology and Computers*.
[40] Joaquim A Jorge. 2001. Adaptive tools for the elderly: new devices to cope with age-induced cognitive disabilities. In *EC/NSF workshop on Universal accessibility of ubiquitous computing: providing for the elderly*. ACM.
[41] Shaun K Kane, Chandrika Jayant, Jacob O Wobbrock, and Richard E Ladner. 2009. Freedom to roam: a study of mobile device adoption and accessibility for people with visual and motor disabilities. In *International ACM SIGACCESS conference on Computers and accessibility*. ACM.
[42] AndréRoch Lecours and François Lhermitte. 1976. The "pure form" of the phonetic disintegration syndrome (pure anarthria); anatomo-clinical report of a historical case. *Brain and Language* (1976).
[43] James R Lewis and Jeff Sauro. 2018. Item benchmarks for the system usability scale. *Journal of Usability Studies* (2018).
[44] Toby Jia-Jun Li, Amos Azaria, and Brad A Myers. 2017. SUGILITE: creating multimodal smartphone automation by demonstration. In *Conference on human factors in computing systems*. ACM.
[45] Sebastián Aced López, Fulvio Corno, and Luigi De Russis. 2015. Playable one-switch video games for children with severe motor disabilities based on GNomon. In *International Conference on Intelligent Technologies for Interactive Entertainment*. IEEE.
[46] Cristina Manresa-Yee, Maria Francesca Roig-Maimó, and Javier Varona. 2019. Mobile accessibility: natural user interface for motion-impaired users. *Universal Access in the Information Society* (2019).
[47] Awais Mansoor, Wamiq M Ahmed, Ala Samarapungavan, John Cirillo, David Schwarte, J Paul Robinson, and Bradley S Duerstock. 2010. AccessScope project: Accessible light microscope for users with upper limb mobility or visual impairments. *Disability and Rehabilitation: Assistive Technology* (2010).
[48] Sergio Mascetti, Mattia Ducci, Niccoló Cantù, Paolo Pecis, and Dragan Ahmetovic. 2020. Developing Accessible Mobile Applications with Cross-Platform Development Frameworks. *arXiv preprint arXiv:2005.06875* (2020).
[49] Ivy N Miller and Alice Cronin-Golomb. 2010. Gender differences in Parkinson's disease: clinical characteristics and cognition. *Movement disorders* (2010).
[50] AB Mink Van Der Molen, AM Ettema, and SER Hovius. 2003. Outcome of hand trauma: the hand injury severity scoring system (HISS) and subsequent impairment and disability. *Journal of hand Surgery* (2003).
[51] John T Morris, Mark W Sweatman, and Michael L Jones. 2017. Smartphone Use and Activities by People with Disabilities: 2015-2016 Survey. (2017).
[52] Maia Naftali and Leah Findlater. 2014. Accessibility in context: understanding the truly mobile experience of smartphone users with motor impairments. In *International ACM SIGACCESS conference on Computers & accessibility*. ACM.
[53] Hugo Nicolau, Tiago Guerreiro, Joaquim Jorge, and Daniel Gonçalves. 2014. Mobile touchscreen user interfaces: bridging the gap between motor-impaired and able-bodied users. *Universal Access in the Information Society* (2014).
[54] Lori Quinn, Monica Busse, and Vanina Dal Bello-Haas. 2013. Management of upper extremity dysfunction in people with Parkinson disease and Huntington disease: facilitating outcomes across the disease lifespan. *Journal of Hand Therapy* (2013).
[55] André Rodrigues. 2015. Breaking Barriers with Assistive Macros. In *International ACM SIGACCESS Conference on Computers & Accessibility*. ACM.
[56] Remo N Russo, Pawel P Skuza, Myriam Sandelance, and Peter Flett. 2019. Upper limb impairments, process skills, and outcome in children with unilateral cerebral palsy. *Developmental Medicine & Child Neurology* (2019).
[57] KM Shah, BR Clark, JB McGill, and MJ Mueller. 2015. Upper extremity impairments, pain and disability in patients with diabetes mellitus. *Physiotherapy* (2015).
[58] Joren Six, Olmo Cornelis, and Marc Leman. 2014. TarsosDSP, a real-time audio processing framework in Java. In *Audio Engineering Society Conference: 53rd International Conference: Semantic Audio*. Audio Engineering Society.
[59] Adam J Sporka, Sri H Kurniawan, Murni Mahmud, and Pavel Slavík. 2006. Non-speech input and speech recognition for real-time control of computer games. In *International ACM SIGACCESS conference on Computers and accessibility*. ACM.
[60] Vibha Tiwari. 2010. MFCC and its applications in speaker recognition. *International journal on emerging technologies* (2010).
[61] George A Tsihrintzis, Dionisios N Sotiropoulos, and Lakhmi C Jain. 2019. Machine Learning Paradigms: Advances in Data Analytics. In *Machine Learning Paradigms*. Springer.
[62] Jean-Philippe Vert, Koji Tsuda, and Bernhard Schölkopf. 2004. A primer on kernel methods. *Kernel methods in computational biology* (2004).
[63] Michelle L Woodbury, Craig A Velozo, Lorie G Richards, and Pamela W Duncan. 2013. Rasch analysis staging methodology to classify upper extremity movement impairment after stroke. *Archives of physical medicine and rehabilitation* (2013).
[64] Tom Yeh, Tsung-Hsiang Chang, and Robert C Miller. 2009. Sikuli: using GUI screenshots for search and automation. In *Annual ACM symposium on User interface software and technology*. ACM.
[65] Bei Yuan, Eelke Folmer, and Frederick C Harris. 2011. Game accessibility: a survey. *Universal Access in the Information Society* (2011).
[66] Nurul Hidayah Zain, Azizah Jaafar, and Fariza Hanis Abdul Razak. 2015. A user-centered design: methodological tools to design and develop computer games for motor-impaired users. In *International Conference on Computing and Informatics*. IEEE.
[67] Nurul Hidayah Mat Zain, Azizah Jaafar, and Fariza Hanis Abdul Razak. 2016. Enjoyable game design: validation of motor-impaired user GameFlow model. *International Journal of Computer Theory and Engineering* (2016).
[68] Xiaoyi Zhang, Harish Kulkarni, and Meredith Ringel Morris. 2017. Smartphone-based gaze gesture communication for people with motor disabilities. In *Conference on Human Factors in Computing Systems*. ACM.
[69] 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 *Conference on Human Factors in Computing Systems*. ACM.
[70] Yu Zhong, TV Raman, Casey Burkhardt, Fadi Biadsy, and Jeffrey P Bigham. 2014. JustSpeak: enabling universal voice control on Android. In *Web for All Conference*. ACM.
[71] Yu Zhong, Astrid Weber, Casey Burkhardt, Phil Weaver, and Jeffrey P Bigham. 2015. Enhancing Android accessibility for users with hand tremor by reducing fine pointing and steady tapping. In *Web for All Conference*. ACM.