Integrating WAI-ARIA suite through browser games' components

Wolgan Ens Quepfert, Wesley Ferreira de Ferreira, Jean Felipe Patikowski Cheiran
Universidade Federal do Pampa
Alegrete, Brasil
Email: ens.wolgan@gmail.com, wferreira531@gmail.com, jeancheiran@unipampa.edu.br

***Abstract*—Digital games are widely present in our lives since 1980s, impacting on human economy and development. Games played through internet have even stronger influence, once they cross demographic boundaries that commonly segregate players. The emerging Web 2.0 and new web technologies have brought opportunities to play digital games directly on web browsers, but some technologies lack suitable accessibility support. WAI-ARIA is an accessibility specification for multimedia applications that could improve web games accessibility, but it seems to be unexplored. This papers aims to present the integration of some WAI-ARIA specification elements into an accessible, educational browser game. Accessibility testing is performed through a screening technique by using screen reader and other assistive technologies. Our main findings include the use of WAI-ARIA live regions in order to report critical content change to blind players and the use of hidden labels to provide additional information through screen reader. The final discussion of non WAI-ARIA solutions supports that the specification is more appropriate and causes less side-effects on game mechanics.**

Keywords-browser games; accessibility; WAI-ARIA.

I. INTRODUCTION

Digital games are present in modern life - accessed from dedicated game consoles to multipurpose mobile devices - and are very representative in the world market. According to ESA [1], a total of 29.1 billion dollars were spent with video game content in 2017 in USA. Beyond the economic side, games represent an important resource in development of human abilities [2], being the learning stimulus an essential aspect in educational games proposition [3]. Games and learning walk side by side during the most of childhood, and games and play collaborate on development of children abilities and competences through exploratory activities [4].

Games also have an important social role by allowing players to keep connected to their friends and parents to socialize with their children [1]. The possibility to find other players by means of online games allows people to cross demographic boundaries (e.g. age, gender, and ethnicity) [2]. The popularization of Internet and the emergence of technologies as Adobe Flash and Adobe Shockwave make possible the construction of games that run directly in web browsers. Those technologies need plug-ins to support an improved multimedia environment that became popular during Web 1.0 and Web 2.0 transition time [5]. After, the consolidation of Web 2.0 and the necessity for more dynamic Web applications, called Rich Internet Applications (RIA), fostered the improvement of native browser technologies - as Hypertext Markup Language version 5 (HTML5) and Cascading Style Sheets level 2 and 3 (CSS2 and CSS3) - by avoiding installation of additional software and favoring development of Web games. Among these new resources, tag elements focused on content semantics, improved visual effects, offline storage and 2D and 3D graphics creation stand out.

In this context, concerning with Web content accessibility which started in a overall perspective with the Web Content Accessibility Guidelines (WCAG) [7] has reached RIA (including Web browser games), leading to the proposition of new specifications as the Accessible Rich Internet Applications Suite (WAI-ARIA) [8] and creating a transition movement: less accessible technologies (as Adobe Flash) were replaced by more flexible, compatible technologies (as HTML, CSS and JavaScript) that favors people with disabilities [6].

Even though many browser games are currently implemented in the HTML + CSS + JavaScript set, studies which describe the integration process of WAI-ARIA into games or present an global overview of accessible web games conceived with the specification in mind were not found. So, this paper aims to present the integration of WAI-ARIA specification into components of an educational browser game for mathematics teaching. Complementary, it is performed a search in scientific literature on WAI-ARIA application in game production.

This paper is organized as follows: at section II, an overview on accessibility for web games and related technologies are presented; at section III, the search process for related work and the papers found are described; at section IV, methods adopted in this research and the chosen game are presented; at section V, accessibility issues and respective solutions are discussed; finally, at section VI, main findings are highlighted and future work are established.

II. BACKGROUND

Browser games can be implemented by using different technologies [15]. One common component of web browser games is the runtime environment that may be: (1) installed in the browser to run a particular technology, or (2) the web browser itself.

In the first case, for example, there are Adobe Flash player, Adobe Shockwave player, Oracle Java Runtime, Unity Web player, Microsoft Silverlight e Novell Moonlight [15]. Adobe Flash is a technology featured popularity and easiness in game production by people with no expert programming knowledge. Because the multimedia nature of content created in Adobe Flash, there was flexibility to present web content in a unique way [6]. The object oriented programming and the event-driven development also contributed to a great collection of browser games in last decades. However, the production of

content that runs on Adobe Flash player has been decreasing and becoming deprecated, since native browser technologies were improved to better support multimedia content, some devices (e.g. Apple smart phones) do not support runtime environment plug-ins, and assistive technologies used by people with disabilities are not fully compatible with non-native technologies [6].

In the second case, HTML versions, CSS levels and specifications of JavaScript programming languages are in continuous improvement. Web consortia and organizations also ensure the implementation of these new specifications into different web browsers. The guarantee of flexibility and support has boosted several web browser game engines to be developed in the last years[1]. Some examples of popular web games developed with HTML5 are agar.io[2] and Cookie Clicker[3].

This transition from installed runtime environments to native resources of browsers is very propitious for the evolution of accessibility in browser games, once accessibility recommendations for web (e.g. [7]) and additional resources for accessibility improvement on RIA (e.g. [8]) might also be applied in game production.

Specially, WAI-ARIA [8] is a technical specification that allows improving accessibility of rich web content by means of the definition of roles, states and properties in HTML code, providing enhanced semantics to application components. Considering the complexity of game components and mechanics, this explicit specification of elements semantics allows assistive technology tools to expose more accurate and complete information to players. However, there are few examples about how this specification can be successfully integrated to a so particular kind of software: games.

III. RELATED WORK

In order to evaluate the application of WAI-ARIA specification in production of browser game, we performed a scientific literature search that aims to answer the following question: which browser games have been produced in integration with WAI-ARIA specification?

So, we applied the search string (“wai-aria” AND “game”) in ACM Digital Library and Scopus databases over the following fields: title, abstract and complete text. Moreover, we applied an equivalent search string in previous editions of Brazilian Symposium on Computer Games and Digital Entertainment (SBGames) proceedings by using advanced search tools at Google to look for those terms in all HTML and PDF available files.

The results of these searches were filtered according the following inclusion criteria:

- Papers that describe the integration of WAI-ARIA in browser games;
- Papers from Computing field;
- Papers in English or Portuguese;
- Papers published since 2008 (publication year of the first WAI-ARIA draft version).

Considering the small number of related works (three), a snowballing technique [9] was included in the search process. Basically, this technique consists of finding other works based on one (or more) seed works, the references in the seeds and the citations to the seeds. Citations to the papers found after the first filtering stage were also searched in ACM Digital Library and Scopus databases, and in the proceedings of SBGames.

Figure 1 presents an overview of the research process and the results in each database, by specifying the original number of works, the number of works resulting from the filtering based on the inclusion criteria, and the final number of related works after the snowballing method. Table I lists three works resulting from the search process.

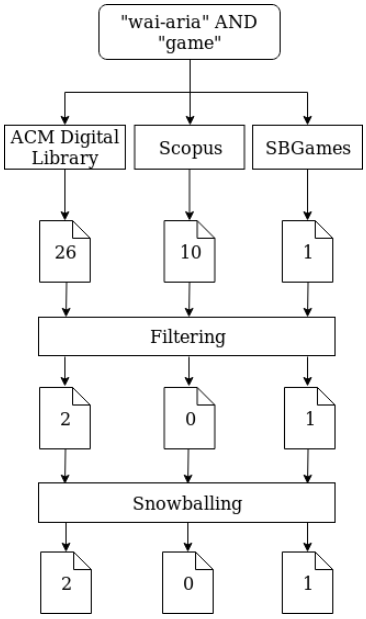

Figure 1. Related work search process

TABLE I. RELATED WORKS FINAL LIST

Paper title	Authors	Year
An inclusive educational game usable via screen reader on a touch-screen [10]	Leporini e Palmucci	2017
A Mobile Educational Game Accessible to All, Including Screen Reading Users on a Touch-Screen Device [11]	Leporini e Palmucci	2017
Uso de protótipo em papel no design de um jogo educacional acessível [12]	Domingues et al.	2014

Leporini and Palmucci [10] present the prototype of a mobile accessible game developed in Cordova. This game is a multiple choice quiz about solar system, sky and earth. The authors evaluate touchscreen interaction related to blind people and run validation tests with two blind players. WAI-ARIA specification is mentioned as tool used in prototype development, but authors do not describe details of roles, states and properties integration. Also, the prototype is currently not available for testing. The authors refer to the same prototype [10] in another work [11], by presenting again details on design, mechanics and play testing that are quite similar to the previous paper. Still, there are no implementation details or game availability.

Domingues et al. [12] propose a point-and-click game called Cuidando Bem. The goal of this game is to support development of abilities and competencies of nursing technicians on patients safety protocols. The authors perform paper prototyping in an iterative, evolutionary

[1] Available at < https://html5gameengine.com >
[2] Available at < http://agar.io >
[3] Available at < http://orteil.dashnet.org/cookieclicker >

process by passing through several prototype improvement steps that are focused on creating playful learning experiences. The prototype is tested into four sessions in play computer style. After, the prototype is developed using web technologies - HTML and JavaScript languages -, and it is possible to find this version of the game at the university website of the authors[4]. This digital version of the game uses the WAI-ARIA specification, and it allows us to analyze the applied elements. Cuidando Bem game uses live regions in characters dialog with the player, labels to more detailed dialog options descriptions, and other modifiers to hide from traditional visual interface the information used exclusively by screen readers.

IV. METHODOLOGY

In this work, we applied WAI-ARIA specification in an educational browser game with focus on accessibility that has already been published in SBGames proceedings. The game Diversão com Números [13] aims to aid learning and teaching of mathematics through exercises related to some arithmetic operations: addition, subtraction and multiplication. The game is built in HTML5, CSS2 and JavaScript languages and uses ReactJS framework.

The goal of the game is to correctly answer the most questions within a minute, where each question answered correctly gives 10 points to the player. Figure 2 shows the main screen of the game being played. The components evaluated and presented in the scope of this paper are highlighted by white rectangles with indication (1, 2, 3).

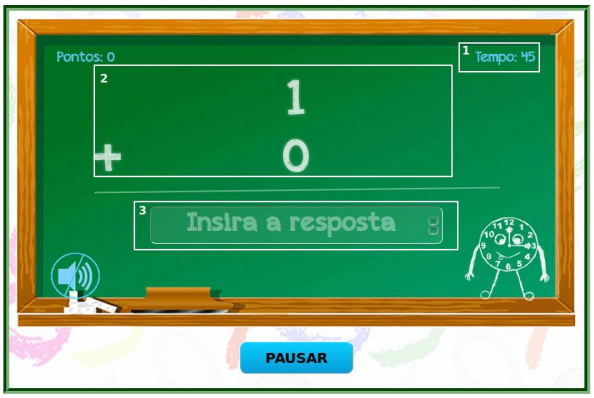

Figure 2. Diversão com Números main game screen [13].

Component 1 is a countdown timer, and it shows how much time is left for the player. It starts with 60 seconds and when it reaches 0 the game shows the game over screen and the player's score. Component 2 shows the numbers and operation of the calculation to be solved by the player. This is the central element of the game chosen for analysis, being critical from the point of view of accessibility and usability for players with disabilities. Finally, component 3 is the text field that receives the result of the calculation from the player. After inputting a number as response (the field only accepts numbers), the player must press ENTER key to confirm his attempt. If the answer is correct, a new calculation is displayed in component 2 and component 3 is cleared.

The WAI-ARIA specification integration process is divided into four main steps: (1) accessibility testing using screening technique [14], (2) identification of accessibility issues, (3) definition of solutions using WAI-ARIA specification, and (4) implementation of solutions.

The screening technique [14] allows preliminary testing to be performed before the participation of people with disabilities in real users testing. Its goal is to find the most basic and most impactful issues with regard to accessibility by means of testing with users with limitations and modes of interaction similar to those faced by people with disabilities.

In this paper, due to resource constraints, two junior researchers with experience on digital accessibility have applied screening. NVDA screen reader was chosen for accessibility testing, because it is free to use and due to its popularity within blind community. Native virtual keyboard was used for accessibility testing that represents people with mouse limitations. Tests were performed on Windows 10 operating system, and the game was run on Google Chrome 67 and Mozilla Firefox 60 browsers.

Solutions were implemented using Sublime Text and Atom text editors. A GIT repository on Bitbucket platform was used to control changes and keep issue records. Heroku cloud application platform was used as a hosting for new WAI-ARIA versions of Diversão com Números.

V. RESULTS AND DISCUSSION

Screening tests with virtual keyboard do not showed game accessibility issues on Windows operating system. On the other hand, tests on Ubuntu revealed a major problem on visualizing some contents, since its virtual keyboard have absolute positioning on screen and, depending on the current magnification level of the web browser, it might cause occlusion of important game elements (e.g. “pausar” button at the bottom of the screen at Figure 2). Unfortunately, this kind of issue cannot be fixed by JavaScript programming or WAI-ARIA application, being inherent in the assistive technology.

Screening tests with NVDA screen reader do not revealed accessibility issues that might be considered a barrier (by impeding the component use) in the original version of Diversão com Números game. However, tests showed significant issues in efficiency of use (the interaction with some components is slower for screen reader users) and calculation reading (some characters are incorrectly detected by the screen reader), reducing the user experience of gaming for people with disabilities. So, children that depend on screen readers would play in disadvantage in comparison to other children. Details on these issues and their solutions are presented next.

The game's timer (component 1 at Figure 2) is an element whose content is updated every second. In the original version of the game, the player that uses a screen reader is not able to know the remaining time if he or she does not navigate to the component whenever he or she wants to get the information. This results in a time consuming action that could be used to solve calculations presented by the game. An alternative solution without WAI-ARIA specification would automatically and temporarily (every 10 seconds) move the browser focus from the current element to the countdown timer. However, this would demand that the game to memorize the last active element (so that it can return after reading the remaining time for the player) and would be vulnerable

[4] Available at < http://sistemas2.sead.ufscar.br/loa/CuidandoBem >

to errors (what would happen if the player triggers a focus change exactly when the focus is moved to the countdown timer?). The solution to these problems using WAI-ARIA is effective and does not turn the game vulnerable: an additional, invisible live region element of medium priority was created. This element is updated every 10 seconds with the remaining time and, due to WAI-ARIA specification, changes are immediately reported to screen reader without changing focus.

The calculation area (component 2 at Figure 2), as well as the timer, has its content dynamically updated: a new calculation is displayed each time the player answers correctly the previous one. So, the player would need to switch the focus between the answer text field and the calculation element after answering correctly to find out what the new numbers and operator. Although the fields are adjacent, this also causes a reduction in the efficiency of playing when compared to a player who does not use screen reader. In addition, the screen reader said "hyphen" instead of "minus" when accessing the minus symbol, since they are the same symbol. An alternative without WAI-ARIA to solve the issues would be analogous to the previous one, leading to an invisible element that would contain the word "minus" (not the symbol) between operands and causing the same previous problems by the focus exchange. The solution adopted when using WAI-ARIA, besides the creation of the live region, includes inserting the aria-label property containing the calculation written in words. In this way, the aria-label property assigns a descriptive text to the calculation that does not cause symbol ambiguity and prevents the creation of unnecessary invisible elements.

The input text field used for the calculation result (component 3 at Figure 2), differently from the other elements, did not present accessibility or efficiency issues. However, the adoption of solutions through WAI-ARIA specification in the previous components also prevented unnecessary modifications to this component.

VI. FINAL CONSIDERATIONS

Digital games play an important role in society, represent an expressive industry in world economy and affect the development of people. Therefore, the experience offered by games should be available to as many people as possible, guaranteeing that different kinds of players are not excluded. The evolution of native web technologies has supported more dynamic applications, including the building of games that run directly in web browser and replacing technologies that do not favor content accessibility. It is possible to notice the lack of studies which describe the integration of WAI-ARIA specification to browser games, reducing the support to game developers concerned with accessibility.

In this paper, we describe the application of WAI-ARIA specification in some components of an educational game that aims to support the mathematics teaching for children. The specification aims to improve game accessibility and to enhance the user experience for players that use screen readers. Accessibility barriers were not found in testing by using screening technique [14] (potentially because the chosen game had been created with accessibility in mind), but some efficiency issues related to screen reader use have emerged. Adopted solutions that included WAI-ARIA elements have consumed little time and have required little effort (this could be related to game simplicity), but they have expressively contributed to maximize efficiency. The used WAI-ARIA attributes that enhanced the usable accessibility of the game are aria-live (creating live regions) and aria-label (creating additional descriptions that are visible to assistive technologies).

Future work includes to cover more assistive technologies (e.g. additional screen readers) while testing, to create a full report of WAI-ARIA integration to more than one browser game, and to include expert blind evaluators to improve testing.

ACKNOWLEDGMENT

Authors would like to thank UNIPAMPA for supporting this research by providing academic scholarships PDA 2018 and AGP 2018. We also thank the assistance of colleagues of PampaGD group who collaborate with insights in the development of this work.

REFERENCES

[1] ESA. "Essential Facts about the Computer and Video Game Industry: sales, demographic, and usage data," 2018. Available at: http://www.theesa.com/wp-content/uploads/2018/05/EF2018_FINAL.pdf

[2] M. Griffths, "The educational benefits of videogames," Education and Health. Vol. 20, No. 3, 2002. pp. 47-51.

[3] R. Savi and V. Ulbricht, "Jogos Digitais Educacionais: benefícios e desafios," Revistas Novas Tecnologias em Educação. V. 6. Nº 2. Dezembro. 2008. pp. 1-10.

[4] U. Ritterfeld, M. Cody and P. Vorderer, "Serious games: Mechanisms and effects," Routledge. 2009.

[5] U. Naik and D. Shivalingaiah, "Comparative Study of Web 1.0, Web 2.0 and Web 3.0," Int. Convention on Automation of Libraries in Education and Research Inst. India, Allahabad. 2008.

[6] WebAIM, "Creating Accessible Flash Content," 2013. Available at: https://webaim.org/techniques/flash/

[7] W3C, "Web Content Accessibility Guidelines (WCAG) 2.0," 2008. Available at: https://www.w3.org/TR/WCAG20/

[8] W3C, "Accessible Rich Internet Applications (WAI-ARIA) 1.1," 2017. Avilable at: https://www.w3.org/TR/wai-aria/

[9] C. Wohlin, "Guidelines for Snowballing in Systematic Literature Studies and a Replication in Software Engineering," Proc. of Int. Conference on Evaluation and Assessment in Software Engineering. United Kigdom, London.2014.

[10] B. Leporini and E. Palmucci, "An inclusive educational game usable via screen reader on a touch-screen," ACM SIGACCESS Accessibility and Computing. 119. USA, New York: ACM, 2017.

[11] B. Leporini and E. Palmucci, "A Mobile Educational Game Accessible to All, Including Screen Reading Users on a Touch-Screen Device," World Conference on Mobile and Contextual Learning, Greece, Larnaca. 2017.

[12] A. Domingues, A. Guimarães, J. Esteves, M. Lotufo, A. Silva, A. , D. Beder, S. Zem-Mascarenhas and J. Otsuka, "Uso de protótipo em papel no design de um jogo educacional acessível," Simpósio Brasileiro de Jogos e Entretenimento Digital. Brazil, Porto Alegre. 2018.

[13] P. Cordeiro, C. Conrad and J. Cheiran, "Redesigning towards accessibility: from a Facebook trivia game to an educational, accessible web game," Simpósio Brasileiro de Jogos e Entretenimento Digital. Brazil, Curitiba. 2017.

[14] S. Henry, "Accessibility in User-Centered Design: Screening Techniques," 2007. Available at: http://www.uiaccess.com/accessucd/screening.html

[15] J. Vanhatupa, "On the Development of Browser Games – Current Technologies and the Future," Int. J. of Computer Information Systems and Industrial Management Applications. Vol. 5. 2013.