

Extending PINATA, a Chrome-based Adaptive User Interface Tool

Christian Ortega

christianortega@tamu.edu

Department of Computer Science and Engineering
Texas A&M University

Tejas Bhalerao, Amy Hurst, Foad Hamidi

{ tejasb1, amyhurst, foadhamidi } @umbc.edu

Information Systems Department
University of Maryland, Baltimore County

I. INTRODUCTION

Abstract- The Internet has become an essential tool for communication, entertainment, and employment. Unfortunately, web-based interfaces can be inaccessible or difficult to navigate for users who experience difficulty in accurately controlling pointing devices such as computer mice due to limited hand motion. We have developed PINATA (Pointing Interaction Notifications and AdapTations), a Chrome-based plugin that adapts a browser's content to assist users with pointing problems and helps them become more aware of their pointing difficulties by providing feedback via customizable notifications. PINATA was first developed in 2015 and initial work focused on assessment and visualization. Over the summer I extended these features, but focused on supporting a new adaptation. Oftentimes hyperlinks to websites can be difficult to target and click due its text's size, color, and/or boldness. To combat this issue, I developed a new adaptation that detects when the user is experiencing difficulty in clicking on a hyperlink, manipulates its text to make it stand out, then offers a hotkey on the keyboard that can be pressed to take the user to the desired link. PINATA is a work in progress that helps users who face pointing problems by allowing them to navigate the Internet with ease. In the future, the new adaptation should be evaluated to learn of its usefulness to users by conducting user studies. PINATA could also be extended in several ways: customization of the newest adaptation, redesign of the notification manager so that all clickable elements can be easily pressed, and added touch-screen functionality.

Accessing the Internet can be challenging for people who experience difficulty in accurately controlling a computer mouse due limited hand mobility. This difficulty can impose a significant barrier as the Internet is an essential tool for many people. To combat this issue, PINATA (Pointing Interaction Notification and AdapTation) was developed. PINATA is a Google Chrome-based plugin in the form of a Chrome Extension that adapts the browser interface and provides feedback via customizable notifications in order to help the user become more aware of their pointing difficulties. This is a continuing research project that was first developed in 2015 that focused on assessment and visualization techniques. Before my work on PINATA, the plugin could only adapt the browser interface by zooming in the webpage the user was on. My summer work focused on the development of a new adaptation that serves the purpose of making the navigation between webpages easier for the user and incorporates the keyboard. Additionally, I developed two data-related features (timeout and toggle-recording) and a PIANATA user interface reset feature.

II. RELATED WORK

In initial stages of PINATA development, user studies were conducted that studied pointing behaviors in a technical perspective [1] and the user experiences of using a PINATA prototype among various groups of people [2][3]. Incorporating the findings that include the need

This research was funded by Computing Research Association - Women (CRA-W) as a part of the Distributed Research Experiences and Undergraduates (DREU) program. Additionally, this research is funded under a grant from the Department of Education, NIDRR grant number H133A130057.

for control, privacy, and accuracy, the PINATA prototype I started with was developed as an independent study with documentation that explains the codebase, Chrome Extension functionality, and other developer details [4].

III. DEVELOPMENT OF A NEW ADAPTATION

Navigating between webpages can be difficult because of the size and placement of buttons, hyperlinks, and other web elements. To solve the problem of clicking on difficult targets, I developed a new adaptation that incorporates the keyboard in order to make the physical clicking of a mouse over a hyperlink unnecessary. This development focused only on hyperlinks and was tested on the home pages of Wikipedia and Craigslist. I was able to develop this code by adding/modifying code in the listener file (written in JavaScript), which records data and carries out the original zoom adaptation. The mouse data needed to support my adaptation and other functions is recorded and extracted using techniques developed and explain in previous work [4].

This adaptation uses the concept of a *slip* to trigger the adaptation sequence. A *slip* is a type of pointing problem that a user can experience that occurs when a user drags the cursor when the intention is to click. Based on preferences set in the Notification Manager (Figure I) by the user, a notification will appear at the top of the browser when the user has hit their preferred slip-to-click ratio. In Figure I, for example, the preference is set so that the Bar + notification type will appear once the user has slipped more than they have clicked by 2%.

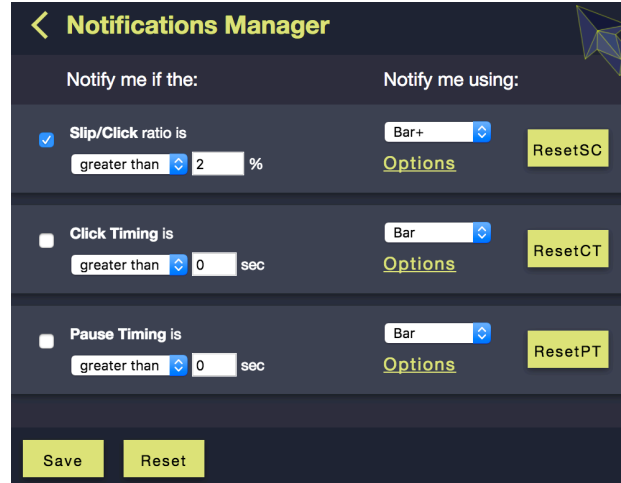


Figure I: PINATA Notification Manager allows end-users to easily customize notification and adaptation settings.

The Bar+ notification includes a message asking the user if they'd like to adapt their interface and provides "Adapt my UI" and dismiss buttons. If the user chooses the adapt button, the next time the user hovers over an element associated with a hyperlink, a green border will appear around that element, indicating to the user that they can use a hotkey (the forward-slash key was used for testing) on the keyboard to proceed to that link rather than physically clicking on a mouse to go to the link. This makes navigating to intended webpages easier because it eliminates the need to be precise in clicking with the mouse. The way this adaptation works is that when the mouse is hovered over an HTML element (using a JavaScript event called *onmouseover*), the element's associated *target*, which is a property of an element, is stored in a variable. The target is then checked to see if an *href attribute*, a property associated with hyperlinks, is present in that element. If the href attribute is present, the HTML *style attribute* is manipulated by setting its border property equal to the hex code for lime-green, which effectively adds a lime-green border around that element. When the mouse hovers over an element, the hyperlink associated with that element is stored in a variable. This hyperlink is updated every time the mouse hovers

over a new element. If the adaptation is active, the JavaScript event called *onkeypress* will take the user to the stored link if the hotkey on the keyboard is pressed. This sequence of events is demonstrated in Figures II and III below.



Figure II: The user triggers a notification when a specified number of slips is detected. The user can then select to activate the adaptation or dismiss the notification.



Figure III: After pressing the “Adapt my UI” button, the web page adds a green border over any element with a hyperlink. This indicates that the user may press the forward-slash (hotkey) on the keyboard to proceed to the link.

IV. PINATA USER INTERFACE AND DATA FEATURES

To aid my understanding of HTML and JavaScript after completing online tutorials, I sought to understand the existing PINATA codebase before developing a new adaptation. To do this, I developed three data-related features: *timeout*, *toggle-recording*, and *reset-per-metric*. The development of all three features required an understanding of Chrome Storage, an API used by Chrome that handles the storage, retrieval, and tracking of data.

The timeout feature pauses the recording of data after a long period of user inactivity and resumes once the user is active again. Using a method that executes its contents after a specified amount of time called *setTimeout* provided by the jQuery library, I was able to create this timeout feature. Similar code was used to develop the toggle-recording feature, which is a button added to the home screen of PINATA that allows the user to turn data on/off at the user’s will in order to provide data privacy to the user. Adding the button to the PINATA interface helped me learn how to manipulate and add HTML elements which helped me add the buttons created for the reset-per-metric feature. This feature was developed for the Notification Manager (shown in Figure I). Before this feature was added, only a single reset button existed that reset all metrics in the Notification Manager to default values. My feature adds a button to each metric that only resets that specific metric’s values and does not affect any other metrics.

VI. FUTURE WORK

In the future, the Notification Manager will be redesigned so that all of its clickable elements can be more easily clicked by the user. User studies will be conducted to observe the effectiveness of the new adaptation and improve the adaptation accordingly. We would also like to look into the possibility of incorporating a Makey-Makey into PINATA, which is an invention kit that contains alligator clips and a circuit board with a USB port. The alligator clips can connect to any conductive material and turn that material into a controller. To suit PINATA’s purpose of assisting those with limited hand mobility, Makey-Makey could be used to turn any conductive object into a functioning computer mouse that can click with a physical tap of the object.

ACKNOWLEDGEMENTS

I would like to thank my DREU program mentor Dr. Amy Hurst, associate professor of Human-Centered Computing in the Information Systems Department at UMBC and my co-mentor Dr. Foad Hamidi, a Human-Computer Interaction postdoctoral research associate also in the Information Systems Department at UMBC. Additionally, I'd like to thank the Department of Education for the funding of PIANATA. Lastly, I'd like to thank the Distributed Research Experiences for Undergraduates (DREU) program that has provided me with this unique and rewarding research opportunity.

REFERENCES

[1] Amy Hurst, Scott E. Hudson, Jennifer Mankoff, and Shari Trewin. 2013. Distinguishing Users By Pointing Performance in Laboratory and Real-World Tasks. *ACM Trans. Access. Comput.* 5, 2, Article 5 (October 2013), 27 pages. DOI=<http://dx.doi.org/10.1145/2517039>

[2] Aqueasha Martin-Hammond, Abdullah Ali, Casey Means, Catherine Hornback, and Amy Hurst. 2016. Supporting awareness of pointing behavior among diverse groups. In *Proceedings of the 10th EAI International Conference on Pervasive Computing Technologies for Healthcare*(PervasiveHealth '16). ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), ICST, Brussels, Belgium, Belgium, 231-234.

[3] Aqueasha Martin-Hammond, Abdullah Ali, Catherine Hornback, and Amy K. Hurst. 2015. Understanding design considerations for adaptive user interfaces for accessible pointing with older and younger adults. In *Proceedings of the 12th Web for All Conference* (W4A '15). ACM, New York, NY, USA, , Article 19 , 10 pages. DOI: <http://dx.doi.org/10.1145/2745555.2746645>

[4] Tejas Bhalerao 2017, Independent Study Report: Enabling Pinata's users to gain more control over their pointing assessment and notifications.