

# *Butterfleye*: Supporting the Development of Accessible Web Applications for Users with Severe Motor-Impairment

Alastair Chetcuti  
Malta Information Technology Agency  
Gattard House, National Road  
Blata l-Bajda, HMR 9010  
[alastair.b.chetcuti@gov.mt](mailto:alastair.b.chetcuti@gov.mt)

Chris Porter  
Faculty of Information and Communication Technology  
University of Malta  
Msida, MSD 2080  
[chris.porter@um.edu.mt](mailto:chris.porter@um.edu.mt)

**Various accessibility standards and guidelines exist, targeting different disabilities. Nonetheless persons suffering from Severe Motor Disabilities (SMD) are generally excluded from development efforts, mainly because of a lack in accessibility regulations, standards and developer support. This work presents *Butterfleye*, a novel developer-centric tool that facilitates the development of accessible gaze-driven web applications for SMD users. *Butterfleye* relies and builds upon a widely-adopted open-source front-end framework to incentivise frictionless developer adoption. Low cost eye-tracking devices are also examined to lower barriers for end-user adoption. We present an open-source library developed iteratively over a series of user-centric studies and report initial evidence of, and observations on, its effectiveness with SMD users.**

*Accessibility, Severe Motor-Impairment, Human Computer Interaction, Web Development, Eye-Tracking*

## 1. INTRODUCTION

Web accessibility is the means by which adoption barriers are reduced or removed altogether however existing accessibility standards do not cater for all user demographics (Henry 2005). Severely motor-impaired sufferers form part of one such demographic for whom research, tools, standardisation and regulatory efforts leave much to be desired. Architects are in many countries bound by law to design buildings that are not only accessible to visually impaired users, but also to people with mobility impairments. This metaphor lends well to digital construction sites since the physical body is still required to navigate online – facilitated through the use of various input devices. Impairment levels range to several degrees of severity, whereby existing support is highly skewed towards visually impaired users (e.g., screen readers) and users with limited motor ability (e.g., sticky keys). The extent of support decreases as the level of ability decreases. A gap in both research and practice exists at the lower-end of the spectrum, highlighted by a lack of developer tools, practical guidelines and official regulations. It is therefore logical to suggest that developers are not incentivised to cater for SMD users. This disincentive is augmented further by the size of this demographic, which crudely put, has little commercial viability.

Existing accessibility standards and guidelines do not assume the potential of eye tracking devices, especially now that low-cost consumer grade eye-tracking devices are gaining popularity (e.g., *Tobii EyeX* and *EyeTribe*). A balance between entry costs (i.e., end-user set-up and training), usability (i.e., accuracy, speed and fatigue) and developer support (i.e., costs and incentives for adoption) is necessary.

## 2. BACKGROUND

Various health conditions can lead to severe motor-impairment, which can in turn lead to loss of limb function and communication abilities (e.g., muscular dystrophy, stroke and amyotrophic lateral sclerosis (ALS)). Such conditions reduce the user's ability to interact with people and machines while the effectiveness of traditional assistive technologies (e.g., mouth stick) diminishes with the severity or stage of the condition. Eye tracking systems have been shown to be effective with patients suffering from these conditions (Murata et al. 2012), however most regulatory and technological efforts largely focus on visually impaired users. Popular front-end frameworks such as Twitter's *Bootstrap* and Zurb's *Foundation* adhere to and support the A11Y project – “a community driven effort to make web accessibility easier” (The Accessibility Project

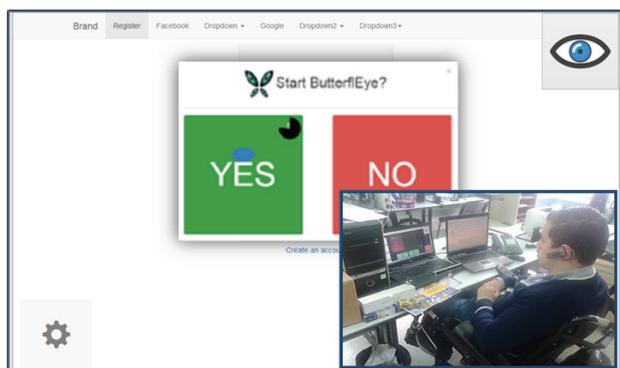


Figure 1: SMD participant operating Butterfleye in-situ

2016). Building web applications with accessibility at the core may require some compromise in terms of visual design and appeal, however these compromises may grow with the extent of disability being supported. To the best of our knowledge there are no developer tools that support the development of eye-gaze compatible web interfaces.

### 3. METHODOLOGY

Following a piloting phase, two user groups were recruited (i.e., 16 able-bodied users and 5 SMD users) in order to iteratively evaluate and develop *Butterfleye*. The SMD group included people with spinal-cord injuries, muscular dystrophy and ALS. All participants had a basic level of ICT proficiency and were able to communicate verbally. Experiments were performed in two different environments: lab-based and in-situ (see Figure 2). Each session consisted of a briefing stage, eye-tracker calibration, system-familiarisation and finally the actual data collection phase via a set of predefined tasks: (1) initialise *Butterfleye* by dwelling on a transparent icon at the top of the page, (2) navigate to a specific page using a drop-down menu and read back the first sentence, and finally (3) send a message using the “contact us” page. Both groups were given the same set of tasks and their performance measured using the Task Completion/Success Rate (TCR) and Time on Task (ToT) metrics. System Usability Scale (SUS), Net Promoter Score (NPS) as well as semi-structured interviews, audio/video recordings and gaze replays were used to substantiate performance observations.

### 4. BUTTERFLEYE’S ARCHITECTURE

*Butterfleye* is a client-side script that dynamically re-renders a Bootstrap-based web application to afford eye-tracking interaction, adding a transparent overlay with larger action buttons, dwell timers, an auto-complete keyboard, auditory feedback and

cursor management (including sticky cursor and dynamic opacity for reading activities). The last iteration focused on front-end developers, resulting in additional configuration options. To integrate (or retro-fit) *Butterfleye* a JQuery plugin (*butterfleye.js*) needs to be included before it is initialised with the following code:

```
1 $(document).butterfleye(  
2 {  
3     dwellTime: 1000,  
4     highContrast: true,  
5     ...  
6 });
```

Listing 1: Initialising plugin with optional parameters

A consumer grade eye tracker (developed by *The Eye Tribe*) was used for testing purposes, selling at a fraction of the cost required for research level eye tracking equipment. This decision was also based on a good balance between cost and accuracy. This eye tracker performs noticeably well compared to other systems, including *Tobii Pro X2-60* and *Tobii EyeX*. *EyeTribe’s* tracker comes with a sampling rate of 60Hz, operating at an average accuracy of 0.5 to 1 degrees in terms of visual angle, resulting in an on-screen error of 0.5 to 1cm assuming the user is calibrated for operation at a distance of 60cm (The EyeTribe 2014).

### 5. OBSERVATIONS AND WAY FORWARD

A tool to support the development of accessible websites for people with severe motor disabilities was developed and evaluated empirically using HCI techniques. *Butterfleye* builds upon and extends an open-source industry-ready front-end framework to afford eye-gaze based interaction in web applications. Initial empirical evidence has shown that SMD users were generally able to complete the given tasks in reasonable time-frames however we acknowledge that this technology requires further long-term field tests to evaluate its feasibility in unsupervised settings. Workload associated with tracker calibration needs to be assessed and further reduced (e.g., rig design and auto-calibration when error rates rise). Based on these initial results and experiences, we plan to (1) extend experiments to include benchmarking data on (a) available assistive technologies (e.g., *Tobii Dynavox’s* Gaze Selection) and (b) alternative eye-trackers (low to high-end), (2) collaborate with Malta’s Foundation for Technology Accessibility (FITA) to run a large scale pilot study on an actual e-government service and (3) obtain an open-source license and publish libraries on GitHub for public consumption and development (e.g., targeting additional front-end frameworks).

## REFERENCES

- Henry, S. L. (2005), 'Introduction to web accessibility', Available from <https://www.w3.org/WAI/intro/accessibility.php>. (14 March 2016).
- Murata, A., Hayashi, K., Moriwaka, M. and Hayami, T. (2012), Study on character input methods using eye-gaze input interface, *in* 'Proceedings of SICE Annual Conference (SICE), 2012', IEEE, pp. 1402–1407.
- The Accessibility Project (2016), The a11y project: A community-driven effort to make web accessibility easier, Technical report. (14 March 2016).
- The EyeTribe (2014), 'Eye tracking 101', Available from <http://dev.theeyetribe.com/general/>. (14 March 2016).