Cyclops: Designing an Eye-Controlled Instrument for Accessibility and Flexible Use

William Payne
New York University
370 Jay St, Rm 323
Brooklyn, NY 11201
william.payne@nyu.edu

Ann Paradiso
Microsoft Research
14820 NE 36th St
Redmond, WA 98052
annpar@microsoft.com

Shaun K. Kane
University of Colorado Boulder
430 UCB
Boulder, CO 80309
shaun.kane@colorado.edu

ABSTRACT

The Cyclops is an eye-gaze controlled instrument designed for live performance and improvisation. It is primarily motivated by a need for expressive musical instruments that are more easily accessible to people who rely on eye trackers for computer access, such as those with amyotrophic lateral sclerosis (ALS). At its current implementation, the Cyclops contains a synthesizer and sequencer, and provides the ability to easily create and automate musical parameters and effects through recording eye-gaze gestures on a two-dimensional canvas. In this paper, we frame our prototype in the context of previous eye-controlled instruments, and we discuss how we designed the Cyclops to make gaze-controlled music making as fun, accessible, and seamless as possible despite notable interaction challenges like latency, inaccuracy, and "Midas Touch."[1]

Author Keywords

Accessibility, Eye Gaze, Human Music Interaction, ALS, Motor Neuron Disease

CCS Concepts

•**Human-centered computing → Accessibility technologies;** User interface design; •**Applied computing →** *Sound and music computing;*

1. INTRODUCTION

A seemingly crucial element of instrument learning is the development of precision, speed, and muscle memory. For people with limited dexterity and motor skills, there are not always obvious paths forward to becoming expressive musicians. New Interfaces for Musical Expression can lower barriers to meaningful music-making given that inputs are not constrained by acoustic realities, while audio outputs and thoughtful mappings [10] can result in incredibly expressive tools. The most famous case of a technology restoring expression is the Theremin at the hands of Clara Rockmore, a musician who gave up the violin due to tendinitis, but the NIME community has demonstrated an active commitment to empowering people of diverse abilities [1, 3, 13, 17]. Still, few developments to date have supported musicians with the most severe motor impairments caused by injuries such as a high-level spinal fracture or diseases such as Amyotrophic Lateral Sclerosis (ALS), also known as Motor Neurone Disease (MND). The Cyclops is designed from the ground up to support musicians who only have control over their eyes and is thus accessible to those with any motor control.

[1]To see the Cyclops in use, a demo video is available `https://youtu.be/u8j7AapyQPY`.

NIME'20, July 21-25, 2020, Royal Birmingham Conservatoire, Birmingham City University, Birmingham, United Kingdom.

1.1 Eye Tracking

Prior work extensively describes the challenges in controlling user interfaces via eye gaze [8]. Functionally, the use of an eye tracker as the primary mode of input introduces two major constraints. The first limitation is *precision*: compared to pointing, targets must be made larger and must be spaced further apart, which significantly reduces the amount of input elements that can be displayed. The second limitation is *timing*: because eye gaze users typically select an on-screen target by holding their gaze over a specific region (known as dwelling), it is difficult or impossible to precisely control the timing or rhythm of inputs. Taken together, these limitations present significant barriers to music creation.

When designing eye-gaze systems, additional human and technological constraints must be considered. Eye tracker accuracy and precision varies widely across users, trackers, and lighting conditions [6]. Targeting tends to be more accurate near the center of the screen, and less accurate on the edges, especially at the bottom and bottom-right corners [6]. A person's head must be positioned carefully for the tracker to work. As Hornof notes [8], the interface used in Duet for Eyes [20] provides visual feedback so that users are aware when they are outside tracker range. People with ALS, an important population of eye-tracker users, lack voluntary control over most of their body. Communication is especially demanding and frustrating due to low typing speed and few opportunities for personal expression [11]. Finally, eye tracker users may use old hardware and software and may not be comfortable installing prototypes for fear of breaking systems that provide crucial access.

1.2 Eye-Gaze Controlled Music

Hornof writes the most comprehensive review of eye-controlled musical instruments and performances to date [8]. One of the earliest designs identified nine eye positions (apparently with some error) used to trigger sounds [18]. The recent availability and relative affordability of eye tracking hardware allows systems to capture eye positions at high sampling rates and feed the resulting x,y coordinates into real-time audio synthesis algorithms (e.g. [9]). Still, ongoing challenges including hardware inaccuracies and gaze interaction constraints, (e.g. "Midas Touch:" accidental target activation during exploration, eye fatigue during use, limits on rhythmic "eye tapping" [2]), have demanded novel design strategies. The EyeHarp addresses the "Midas Touch" Problem through a [24] radial design allowing a user to rest their eyes at the center and move outward triggering intended targets without accidentally crashing into targets along the way, while Netytar [5] addresses navigation complexity with an isomorphic interface which, its authors argue, reduces screen distance between targets and input error in comparison to the EyeHarp. EyeJam [15] removes dwell latency entirely through a Context Switching paradigm [15] in which triggering a note is done through crossing a "bridge" at the center of the screen after selecting a pitch above or below it. However, EyeJam limits the number of available inputs since identical pitches are duplicated above and below the bridge. While we focused entirely on gaze, recent systems explore multimodal inputs. Dueto is adjustable to player ability supporting switch and cooperative modes [23]. Finally, Focal, is a see-through, head-mounted display in which players select various effects with their eyes extending the capabilities of a single expression pedal [7].

In a few cases, eye-controlled instruments have supported the musical expression of players with disabilities. Duet for Eyes was a performance bringing together users with severe motor impairments, AAC experts, and others and largely taking advantage of off-the-shelf technologies using gaze to trigger pre-selected audio files [20]. The EyeHarp [24] involved an informal evaluation with people with cerebral palsy, and Ableton Live Adapted [8] was a collaboration between two people, one with ALS, to develop an eye-controlled Ableton Live interface overlay.

2. DEVELOPMENT

The Cyclops builds on and differs from existing work. Due to a primary motivation to make musical expression more accessible, it uses a commercially available eye-tracker with free drivers, and it is developed entirely in Universal Windows Platform (UWP) with the goal of eventual release on the Windows Store. Furthermore, the design ethos of the Cyclops can be summarized as "low floors and high ceilings" [19] resulting in a multi-screen design with independent yet interconnected modules supporting scaffolded growth and takes inspiration from the Dato DUO [4], a two-player synth designed for kids.

2.1 Hardware and Requirements

Our system is tested with the Tobii Eye Tracker 4c and the Tobii PcEye Mini (both can be found under $200) [22]. Tobii eye trackers use a combination of near-infrared light, camera capture, and onboard image processors, and are quite small mounting to the bottom of a monitor. When sold as an assistive technology for communication (AAC), Tobii eye trackers are very expensive, usually over $1000 [21]. The bulk of this cost is the software which converts text and symbols to speech, and provides other applications to enhance computer access for eye tracker users. The Tobii 4c is also sold as a gaming device without expensive AAC software. The tradeoff is that the free drivers provide limited customization and only six-point targeting for calibration. It also conceals how raw capture data of eye position is filtered and reduced into the gaze point shown on screen limiting any alteration of the underlying capture algorithms.

2.2 UWP/Gaze Interaction Library

Many, if not most, who rely on eye trackers use Windows due to the operating system's built-in support for eye control. Thus, the main advantage of UWP as a development platform is its compatibility with existing tracker-enabled devices. (We successfully installed the Cyclops on a Tobii Dynavox AAC device.) Simplifying development of eye gaze-enabled applications, the Gaze Interaction Library is available freely to Windows developers as part of the open-source Windows Community Toolkit. With only three lines of code, the Gaze Interaction Library adds dwell-based activation to common controls, and the API supports significant customization. Each control contains "Enter," "Fixation," "Dwell," and "Exit" properties enabling the developer to specify the timing and events that occur when a user looks at or away from a control. (Dwell, unlike fixation, is intended to indicate comprehension of the control's purpose.) Furthermore, the library supports customization over the appearance of the gaze cursor, a visual representation of where the tracker detects the eyes, and access to its pixel position.

2.3 Audio Synthesis

While UWP supports some audio output through the AudioGraph API including file playback, routing, and a few built-in effects, UWP is not intended for building sophisticated audio or music applications and does not contain an audio synthesis engine. We wanted the Cyclops to produce high-quality yet comprehensible sounds and not to emulate acoustic instruments with pre-recorded files or MIDI synths. Synthesizers are expressive instruments that do not necessarily require fine motor skills or precise timing, yet enjoyment of them is often the result of experimentation and happening upon unique settings. This is made possible through synthesis.

One feature of UWP is the ability to embed a web application rendered with Microsoft Edge, thus allowing for the use of the Web Audio API. The interface of the Cyclops is built in C# and XAML, the synthesizer is written in Javascript and runs completely offline, and the two communicate with each other using built-in UWP messaging functions. The audio components are built on Tone.js [14], a Web Audio framework for creating interactive music in the browser. The audio implementation includes:

- Polyphonic Synthesizer
  - Four oscillators
  - Amplitude and filter envelopes
  - Effects: bitcrusher, chorus, delay, low pass filter
  - Effects automation
- Sequencer
  - 1–16 steps
  - Accent/on/off
  - Separate gate sequencer from pitch sequence

3. INTERFACE DESIGN

We began with two primary design goals:

1. Low barrier of entry (low floors): Gaze input should be as easy and accessible as possible.

2. Expressive control (high ceilings): Live play should be fun and engaging with room for growth.

3.1 Low Barrier of Entry

To provide a low barrier of entry, the Cyclops is inspired by a synthesizer for kids, utilizes predefined and user-selectable dwell times, and gives ample visual feedback.

Table 1: Dwell Times for Controls on the Cyclops

Control	Total Dwell Time
Play/Pause Button	800 ms
Sequencer Selection	450 ms
Synthesizer Attack (Slow)	450 ms
Synthesizer Attack (Medium)	200 ms
Synthesizer Attack (Fast)	60 ms

3.1.1 *Dato DUO Inpsired*

The chief design inspiration for the Cyclops is the Dato Duo Synth [4]. The Dato Duo is designed for two collaborators, one responsible for an 8-step sequencer and the other for timbral qualities. Unlike the Dato Duo, we divide control into three screens, "Instrument," "Sequence," and "Effects" to support more playable notes and sequencer control. We reduce interaction to a single player model primarily through effects automation. Splitting musical control across screens renders targets extra large compensating for jitter and eye tracker error. It is also intended to reduce cognitive load since only a subset of controls are available at any given time. (While the EyeHarp also supports expressive control, its design of multiple selections within one radial input results in over a hundred selectable elements on screen at once [24].) To compensate for divided control, every screen contains unifying features including a Play/Pause button and a miniature keyboard ensuring there is always a way to produce sound. Furthermore, at the very bottom is a non-selectable representation of the pitch sequence. This region can be safely fixated upon without affecting the synthesizer in any way.

3.1.2 *Dealing with Midas Touch*

"Midas Touch" is a design challenge in creating responsive gaze-only interfaces. When dwell time is long, the user has enough time to identify and comprehend controls before activation, but the resulting interaction can feel sluggish. Reducing dwell time results in "Midas Touch" where the user activates controls accidentally, e.g. when moving from a button on one corner of the screen to another (in the same way that everything Midas touched turned to gold). Thus, dwell time determination presents a trade-off between precision and timing amplified in musical contexts where timing is literally everything. We address dwell times in two primary ways. First, dwell times across the interface are not homogeneous, but vary widely based on musical task (Table 1). Low-level/timing-critical tasks, (e.g. playing a note, toggling the gate sequencer), require a short dwell, while higher-level tasks that have a major effect, (e.g. powering the sequencer, altering global settings), take longer reducing the chances of accidental triggering. Effects automation uniquely requires an activation time, but then becomes dwell-less tracing the user's gaze path. Second, like the dwell time musical keyboard implemented in EyeJam [16], we support user control over note activation dwell times on the "Instrument" screen through a single button that toggles between slow, medium, and fast modes. This feature affords different musical tasks. When trying to learn or recreate a melody, a longer dwell grants more time to identify correct notes and reduces the chance of accidental notes. When soloing/improvising, a shorter dwell time enables speed.

3.1.3 *Visual Feedback*

In gaze-based interfaces, it is important balance feedback with simplicity indicating tracker accuracy without too much distracting or uncomfortable clutter. The Cyclops uses a gaze cursor, a small dot, to show where the player is looking on-screen, but goes much further than prior eye-gaze instruments in providing visual feedback to make play feel responsive and fluid. Screens slide in and out of view, control surfaces expand when looked at, and note-triggering buttons snap outward during synthesizer attack all providing large central targets without much visual clutter. We have found that the sluggishness of dwell can be partly mitigated by earlier-triggered animations when the user enters, fixates on, and exits controls. Actions always produce an immediate visual effect. (Of course, this only affects the perception of responsiveness, but does not accelerate control input or solve musical timing challenges.)

3.2 Expressive Control

Through interacting with the DATO Duo, our team was surprised to discover that we did not all unanimously prefer one side and each enjoyed it differently. Similarly, the Cyclops uses separate screens that can be played independently or together through automation. A new player might stick entirely with the "Instrument" screen learning various tunes. Because the sequencer is always recording, they may input a melody on their own tempo and then toggle the sequencer looping it in time. The Cyclops provides room to grow. More experienced players may add dynamics and warping timbral changes on the "Effects" screen or create complex rhythms adjusting the relative sizes of gate and pitch sequences on the "Sequence" screen. Due to high-quality audio and tonal pitch sets, virtually everything performed sounds "good," from minimalist grooves to ambient clouds.

3.2.1 *Instrument: Key Layout*

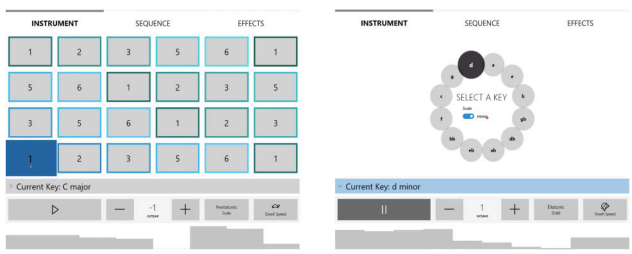

Figure 1: "Instrument" screen. The number of notes available in the Key Grid can be toggled to a pentatonic mode (left), while the available pitches are locked to a user-selected key (right).

On the "Instrument" screen, we elected for a two-dimensional grid design, inspired by a string instrument fretboard, for its simplicity and understandability (Figure 1). (This is most similar to the Netytar [5].) The available pitches in the grid are locked to a user-selected root, octave, key (major or natural minor), and mode (diatonic or pentatonic). The bottom row begins with the root, while above rows are transposed such that the top row is an octave higher than the bottom. Duplicate note targets enable a user to repeat notes in a single gaze motion rather than look away from and back onto a note (a problem also mitigated with switch control). Horizontal movement along the rows produces the steps in a scale. Vertical movement produces arpeggios. Gutters, gaps in between buttons, help reduce "Midas Touch."

3.2.2 *Sequencing/Recording*

Given that the Cyclops is intended to be an instrument above composition tool, our design enforces the rule that

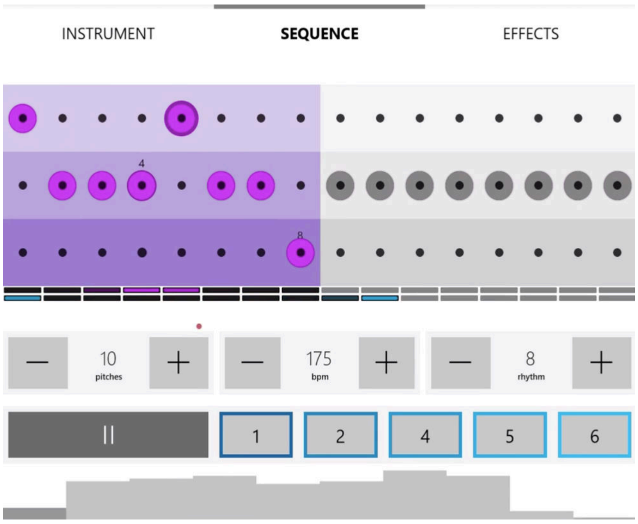

Figure 2: "Sequence" screen. A sequence with 10 pitches is controlled by a gate sequence of length 8.

users must "play" each and every note. Rather than implementing a standard step sequencer grid, we again emulate the design of the Dato Duo Synth in which every note played is automatically recorded to the sequence. This feature promotes simplicity since there is no extra "record" or "loop" button, as well as reduce fatigue enabling a player to input a phrase only once at a tempo comfortable to them and then loop it. When the sequencer is toggled on, notes inputted by the player overwrite the current step in the sequence locking their inputs in time. The "Sequence" screen provides additional expressive control allowing notes to silenced or accented through an independent gate sequencer (Figure 2). Choosing distinct prime numbers for the pitch and gate sequence lengths can produce long, complex patterns with little effort.

3.2.3 Effects

Figure 3: "Effects" screen: The player controls crush, mapped to Y, and chorus, mapped to X.

Knobs and sliders do not transition easily to eye gaze control. We drew inspiration from the Korg Kaoss Pad, a touch-surface in which two audio parameters are altered based on the x/y position of the cursor [12]. Like the note input screen, the "Effects" control automatically records user behavior allowing for real time eye-gaze gesture recording and effects automation (Figure 3). For instance, the player can record a five second volume swell on the "Effects" screen, then move to the "Instrument" screen and improvise with a looping, dynamic volume. Or, the player can trigger the sequencer and move to the "Effects" screen and alter the loop's timbre, or even record both and take a break. In effect, they can control many musical parameters at once, despite only two dimensions of input. While this feature does not replace an acoustic musician's simultaneous control over pitch, timbre, and amplitude, it does add a layer of expressive freedom not evident in previous eye-gaze instruments.

4. FUTURE WORK

Numerous updates and additions would make the Cyclops an even more adaptive and fully-featured music technology. On the "Instrument" screen, we could support customization over the number of columns and transposition of rows the same way string instruments afford alternate tunings. On the "Effects" screen, we could allow automations to be synced and de-synced to the sequencer. We would like to explore error-aware features such as automatically resizing/repositioning targets based on the deviation of gaze points captured during fixations [25]. Such features may support adapting the interface to various screen sizes. While the Cyclops began as a gaze-only design, adding switch support to enhance timing control for users with some body movement is a low hanging fruit, and the ability to split screens into multiple windows would enable cooperative play. Finally, given that gaze-only interfaces map well to single-touch inputs and the Cyclops is already almost fully touch compatible, it might support individuals with other disabilities and less severe motor impairments.

4.1 Formative Design and Testing

At this stage, the Cyclops has been primarily played by the first author in solo and group improvisation settings. We received feedback from developers of Windows Eye Control and the Gaze Interaction library who helped us understand the affordances and constraints of eye tracking affecting most aspects of design from target layout to dwell times. They emphasized that accessibility barriers beyond the software itself can block use, e.g. an inaccessible popup, leading us to choose UWP as our development platform. Furthermore, without telling us how it should sound, they felt strongly that it should be capable of producing recognizable tunes. In designing our "Instrument" screen, we made sure we could play "Yankee Doodle" and "Happy Birthday" along the way. We also demonstrated the Cyclops to two individuals with Cerebral Palsy and their caretakers but we were not able to position them in tracker range for testing. We are actively working to schedule a session with a musician with ALS who has given feedback to our team about interface designs in the past. This individual uses a very old version of Windows that we cannot install the Cyclops on and we have been unsuccessful to date at scheduling a test session.

5. CONCLUSION

The Cyclops is an expressive eye-controlled synthesizer intended to enable live improvisation and play for musicians with severe motor impairments. Work on the Cyclops is ongoing and has yet to be evaluated. Next, we intend to conduct usability tests, iterate, and ultimately release the Cyclops for much wider use.

6. ACKNOWLEDGMENTS

We would like to thank the MSR Enable and Ability Teams, especially Dwayne Lamb and Christopher O'Dowd.

7. REFERENCES

[1] I. C. Almeida, G. Cabral, and P. G. B. Almeida. AMIGO: An Assistive Musical Instrument to Engage, Create and Learn Music. *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 168–169, 2019.

[2] R. Baath, T. Strandberg, and C. Balkenius. Eye Tapping : How to Beat Out an Accurate Rhythm using Eye Movements. *Proceedings of the International Conference on New Interfaces for Musical Expression*, (June):441–444, 2011.

[3] R. B. Conservatoire, J. Wright, and J. Dooley. On the Inclusivity of Constraint : Creative Appropriation in Instruments for Neurodiverse Children and Young People. pages 162–167, 2019.

[4] Dato Musical Instruments. Dato DUO: The synth-for-two. Retrieved January 20, 2020 from `https://dato.mu`.

[5] N. Davanzo, P. Dondi, M. Mosconi, and M. Porta. Playing music with the eyes through an isomorphic interface. *Proceedings - COGAIN 2018: Communication by Gaze Interaction*, pages 1–5, 2018.

[6] A. M. Feit, S. Williams, A. Toledo, A. Paradiso, H. Kulkarni, S. Kane, and M. R. Morris. Toward Everyday Gaze Input. pages 1118–1130, 2017.

[7] S. Greenhill and C. Travers. Focal : An Eye-Tracking Musical Expression Controller. *NIME 2016 Proceedings of the International Conference on New Interfaces for Musical Expression*, 16:230–235, 2016.

[8] A. Hornof. The Prospects For Eye-Controlled Musical Performance. *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 461–466, 2014.

[9] A. J. Hornof and L. Sato. EyeMusic: Making Music with the Eyes. *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 185–188, 2004.

[10] A. Hunt, M. M. Wanderley, and M. Paradis. The importance of parameter mapping in electronic instrument design. *International Journal of Phytoremediation*, 21(1):429–440, 2003.

[11] S. K. Kane, M. R. Morris, A. Paradiso, and J. Campbell. "at times avuncular and cantankerous, with the reflexes ol a mongoose": Understanding self-expression through augmentative and alternative communication devices. *Proceedings of the ACM Conference on Computer Supported Cooperative Work, CSCW*, pages 1166–1179, 2017.

[12] KORG Inc. Kaoss Pad KP3+ Dynamic Effect/Sampler, 2020. Retrieved January 20, 2020 from `https://www.korg.com/us/products/dj/kaoss_pad_kp3_plus/`.

[13] A. M. Lucas, M. Ortiz, and D. F. Schroeder. Bespoke Design for Inclusive Music: The Challenges of Evaluation. *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 105–109, 2019.

[14] Y. Mann. Interactive Music with Tone.js. In *Proceedings of the 1st annual Web Audio Conference*, 2015.

[15] C. H. Morimoto and A. Amir. Context switching for fast key selection in text entry applications. *Eye Tracking Research and Applications Symposium (ETRA)*, pages 271–274, 2010.

[16] C. H. Morimoto, A. Diaz-Tula, J. A. T. Leyva, and C. E. L. Elmadjian. Eyejam: a gaze-controlled musical interface. In *Proceedings of the 14th Brazilian Symposium on Human Factors in Computing Systems*, pages 1–9, 2015.

[17] S. T. Parke-Wolfe, H. Scurto, and R. Fiebrink. Sound Control: Supporting Custom Musical Interface Design for Children with Disabilities. *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 192–197, 2019.

[18] A. Polli. Active Vision: Controlling Sound with Eye Movements. *Leonardo*, 32(5):405–411, 1999.

[19] M. Resnick and B. Silverman. Some reflections on designing construction kits for kids. *Proceedings of: Interaction Design and Children 2005, IDC 2005*, pages 117–122, 2005.

[20] SMARTLab Digital Media Institute. World premiere: DUET for EYES: Eyejamming & eyebodyweaving., 2008. Retrieved January 20, 2020 from `http://smartlab-ie.com/flyer/dublin_mytobii09.pdf`.

[21] Tobii Dynavox. PCEye Mini, 2020.

[22] Tobii Gaming. Eye Tracker 4c, 2020. Retrieved January 20, 2020 from `https://gaming.tobii.com/tobii-eye-tracker-4c/`.

[23] S. Valencia, D. Lamb, S. Williams, H. S. Kulkarni, A. Paradiso, and M. Ringel Morris. Dueto: Accessible, Gaze-Operated Musical Expression. In *The 21st International ACM SIGACCESS Conference on Computers and Accessibility - ASSETS '19*, pages 513–515, Pittsburgh, PA, USA, 2019. ACM Press.

[24] Z. Vamvakousis and R. Ramirez. Temporal control in the EyeHarp gaze-controlled musical interface. *Proceedings of the 12th International Conference on NIME*, pages 11–16, 2012.

[25] J. O. Wobbrock, S. K. Kane, K. Z. Gajos, S. Harada, and J. Froehlich. Ability-based design: Concept, principles and examples. *ACM Transactions on Accessible Computing*, 3(3):1–27, 2011.