WELLE - a web-based music environment for the blind

Jens Vetter
Tangible Music Lab
University of Art and Design
Linz, Austria
jens.vetter@ufg.at

ABSTRACT

This paper presents WELLE, a web-based music environment for blind people, and describes its development, design, notation syntax and first experiences. WELLE is intended to serve as a collaborative, performative and educational tool to quickly create and record musical ideas. It is pattern-oriented, based on textual notation and focuses on accessibility, playful interaction and ease of use.

WELLE was developed as part of the research project *Tangible Signals* and will also serve as a platform for the integration of upcoming new interfaces.

Author Keywords

Musical Interfaces, Accessibility, Livecoding, Web-based Interfaces, Collaborative Music-Making

CCS Concepts

•**Applied computing** → **Sound and music computing;** *Performing arts;* •**Human-centered computing** → *Accessibility systems and tools; Collaborative and social computing;* •**Software and its engineering** → Designing software;

1. INTRODUCTION

Using the computer for music production is exciting and opens up possibilities that are only achievable with digital electronic instruments. Not only sound generation, but also the integration of a multitude of interfaces is often a starting point for creative processes. If, however, the computer has to be used with assistive technologies - as is the case for blind people - digital music production quickly becomes a challenge that can only be realized with a lot of preparation and external help. Difficulties do not only arise during the initial installation of new software. Also, the integration of devices such as the Braille keyboard or the Braille display into the computer system is challenging. When it comes to computer music software, a widespread problem is the integration and the accessibility support for screen readers[1], and therefore the support of Braille devices.

To improve the process of music production on the computer for the blind, several attempts have been made in the past focusing on assistive external hardware, such as the Haptic Wave [14], the HaptEQ [6] or the Moose [5]. All these external interfaces, however, are based on the integration into a music production environment. But only very few audio software platforms support screen readers like JAWS. Amongst them is the cross-platform audio software Audacity. Yet in terms of music production, Audacity is only suitable for editing single audio files, but not for more complex editing of multiple audio tracks or even for composing computer music. DAW's (Digital Audio Workstations) like Sonar or Reaper allow more complex audio processing and also offer screen reader integration, but using them with the screen reader requires extensive training.

2. COMPUTERMUSIC AND CODE

While the interaction with the GUI and thus the use of a screen reader can be demanding, some blind people are much faster and more intuitive when writing and reading program code. Highly trained blind software developers and programmers like Florian Beijers, Michael Forzano, Herwin Haliman and many others[2] work alongside sighted colleagues. The only difference to the way their sighted colleagues work is their use of screen readers and their ability to memorize once written and read code better than their colleagues.

2.1 Coding for Kids

Coding is also already used to inspire and teach children and teenagers. Educational projects that deal with the playful learning of programming skills through physical objects such as Torino and its successor Code Jumper [11], or with regard to sound generation, the Little Bits project [12], have gained huge attention. On the software side, tools like the visual programming language Scratch, or Sam Aaron's and Alan Blackwell's groundbreaking work on the Sonic Pi [2][3], which even runs on the single board computer Raspberry Pi, illustrate the success this approach can have and have been used in numerous classroom workshops and live performances. Unfortunately, Scratch is not useful for blind children [4].

2.2 Audio Programming Languages

Thus, programming could offer an obvious approach towards the computer as music machine for blind people. There are already numerous music-oriented programming languages, such as Csound or SuperCollider [16], that use

[1]list of screen readers `https://www.afb.org/blindness-and-low-vision/using-technology/assistive-technology-products/screen-readers`

[2]examples of blind programmers `https://economictimes.indiatimes.com/tech/software/meet-the-ones-who-cant-see-but-they-can-code/a-software-that-changed-aloks-world/slideshow/69292299.cms`

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

text editors as their main interfaces. Some of them even were primarily developed for the practice of livecoding, e.g. TidalCycles [10] or Overtone [1]. Particularly interesting for blind beginners are systems like the aforementioned Sonic Pi or ixi lang [7], that work with reduced complexity and highly abstracted programming language.

However, regarding blind users, problems can also arise with text-based music environments: often the installation of software itself poses a challenge, or the user interfaces are not optimized for screen readers and Braille devices, e.g. the SuperCollider IDE or Sonic Pi IDE.

2.3 Web-based audio programming

Web apps avoid having to install software and run on common browsers, and screen readers are able to navigate most websites.

Web-based music platforms, that use a simplified programming language are already available, such as Livecodelab 2.5, SLANG, Punctual and more. Their highly abstracted language quickly explains itself to beginners and, after a short exploration, encourages to make music. To what extent these specific web apps are accessible for screen reader and Braille devices, will be discussed below.

However, despite a wide variety of existing music platforms and approaches, it is still difficult to find accessible tools for sound creation, that can easily be used by the blind.

3. WELLE - A SIMPLIFIED MUSIC ENVIRONMENT

To offer a simple, collaborative and playful music environment based on just-in-time programming to blind people, and in order to overcome barriers like software installation or incomplete screen reader integration, the WELLE platform was developed. A live version is accessible under `https://tangible.uber.space/`

The development of this web app is part of the research project *Tangible Signals* [15], which focuses on the physical representation of sound and haptic control feedback. For this project a collaboration with blind and visually impaired pupils was initiated that led to the creation of WELLE. The app will also serve as a platform for integrating interfaces, that will be built in the course of the research.

3.1 Concept and development

The concept for WELLE was developed during several workshops at the Institute for the Blind in Vienna while working with a group of five pupils aged 12-14 years and their teacher. The teacher and four of the pupils were blind and one pupil was partially sighted.

In search for a suitable and accessible music platform for this workshop, several sound platforms were presented to the group and were explored by the participants during the workshop. The selection focused on web-based livecoding platforms in order to use text as musical interface and to circumvent any software installation, so that the participants could also rehearse also on their computers at home for example. The platforms Gibber and Lich [13][9] were excluded from the exploration because their interface was not accessible at the time to the screen reader and their programming language was rather complex for beginners. For the same reasons Audiomasher, Herokuapp, Wavepot, 8bitworkshop, SOUL, FAUST, Klangmeister, Estuary and also the online DAW Earsketch were excluded.

The preselected platforms were the Livecodelab 2.5[3], SLANG[4] and Punctual[5] were chosen because of their simplified programming language and because a screen reader could at least navigate the website.

However during the exploration in the workshop none of the chosen platforms were functional and accessible when used with screen readers and Braille displays. As a result, without installing software on the computers provided by the school, the participants of this workshop could not receive an effective introduction to sound generation.

Since other web-based platforms than those already mentioned are not known to the author, the conclusion of this first exploration, is that none of the existing web-based livecoding platforms were either accessible or suitable for the blind and visually impaired participants of the workshop.

As a consequence, and in order to provide a musical tool based on a textual interface to the pupils for the next workshop, a first web-based prototype was created (Figure 1), featuring very basic text commands to trigger sounds. But more importantly, an accessible web design and audio feedback were integrated. The resulting web app consisted of several subpages, which enabled users to play, stop or change prepared sample loops. Available subpages were *drums*, *bass*, *melody*, *pads*, *hihats* and *vocals*. The pupils could choose one instrument, open the corresponding subpage on their computer and start experimenting. Initially all pupils in the room were using their headphones, since they could not only hear the audio samples, but also the screen reader information. Later the headphones were unplugged in order to hear the group's overall sound.

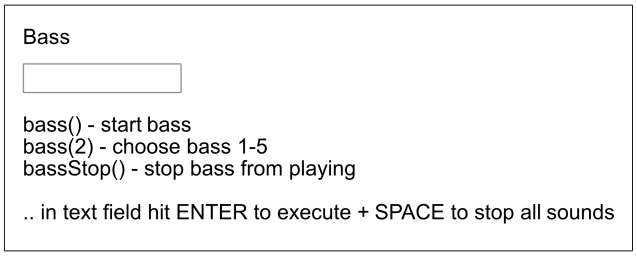

Figure 1: screenshot, web prototype

In the discussion with the participants about another iteration of this website, various aspects were addressed:

- **simple language** the text commands should allow broader interaction with the sound, but should stay simple and easy to remember
- **assistive design** apart from being optimized for the use with screen reader, the design of the website should be assistive, e.g. the cursor should jump directly to the input field after loading the page, also audio feedback should be integrated, e.g. indicating input errors
- **audio recording** it should be possible to record and download the created music
- **collaboration** a collaboration across multiple computers should be possible

[3] `https://livecodelab.net/play/`

[4] `http://slang.kylestetz.com/`

[5] `https://dktr0.github.io/Punctual/`

3.2 Software design

Based on the results of the first workshops described above, WELLE was developed as a simplified music environment, accessible to blind people, that runs in a browser and is based on a textual interface. The core component is a text-based notation system, that according to Magnusson's book *Sonic Writing* could be described as a hybrid between live-coding and "action notation" [8]. The notation is heavily inspired by Livecodelab 2.5, ixi lang and Steno (SuperCollider) and allows intuitive access to pattern-based sound generation through simplification and abstraction.

For the creation of the syntax the parser generator Ohmlang was used[6]. The website is programmed in Javascript and HTML[7] and utilizes the Tone.js library for sound generation. The web app runs on standard browsers, except the IE Internet Explorer, which was commonly used in the school environment where the workshops took place, alongside Mozilla's Firefox.

3.3 Interface design

The interface of the web app (Figure 2) is text-based and centred around an input field, which serves as main interaction element. All elements are programmed with regard to accessibility e.g. as text, tables or check boxes. The main elements from top to bottom are:

- **Control section** various control options, e.g. mute switches for the entire sound or for alerts, an overview about the status of recordings, media player and groups

-**Command history** an overview of all commands entered. The arrow keys can be used to recall commands. In group sessions the history is shared between all participants

- **Input field** commands can be entered and will be executed by hitting the return key

- **Instrument overview** a list of all active instruments and parts

- **Help section** an overview of all commands and interactive examples

3.4 Notation syntax

WELLE's underlying principle is based on sequences of repeated patterns. By entering patterns for a specific instrument, a sequence is created that loops after it has reached its last step. If patterns of different lengths are entered, polyrhythmic loops are created. Shorter patterns will loop earlier. The instruments run in sync with each other. To start an instrument, its name followed by a pattern has to be entered in the input field and has to be sent to evaluation by pressing return. The pattern can consist of a hash '#' for a hit, and a minus '-' for a pause. The pitch of a note or respectively the sample rate of a sample can be altered by adding a number after the hash. Examples of expressions:

```
bass # - # # - - #
drums #2 # - # #3 #2 -
drums > bass
save mybeat
. bass
>
```

Typing an instrument name followed by a pattern starts the instrument. Typing a dot '.' followed by the instrument name stops the instrument. Using '.' or '>' without additional specification name will stop and start the whole sequence.

Patterns can be copied to other instruments, using the '>' character. Patterns can also be combined and abstracted by using brackets. Every instrument offers some additional parameters, e.g. instrument volume and randomization of the pattern. When randomizing an instrument, e.g. 'rand 2 drums', the note and rest events of the underlying pattern will be randomized after the pattern finished two cycles.

The tempo of the whole sequence can also be changed, e.g. using 'BPM 132'. Moreover, slowly increasing or decreasing tempo changes are also possible by adding a number, which determines the seconds of the transition. Most of those functions can also be applied to multiple instruments at the same time.

Regarding compositions, it is possible to save individual parts while playing. They are snapshots of the current active collection of instruments, patterns and other parameters such as volume and randomizing. These parts are then displayed below the input field and can be reactivated at any time. They can be used to e.g. structure a performance. Other features are network collaboration, preset storage, interaction alerts, recordings of the session. The help section at the bottom of the website contains interactive examples and more detailed explanations of implemented features.

Figure 2: screenshot of the live version of WELLE at https://tangible.uber.space/

[6] `https://ohmlang.github.io/`, the use of Ohmlang was inspired by SLANG

[7] other used frameworks are Node.js, Websockets and Socket.io

4. WORKSHOP

The WELLE web app was first used at the *OCC - Austrian Computer Camp* during a 3-hour workshop with a group of six pupils aged 10-14 years, of whom five were blind and one was visually impaired. Also, two blind teachers participated in the workshop.

The pupils were using their private computers. None of the pupils had prior experience with computer music or programming. Within the first hour the pupils got used to WELLE's notation syntax, commands and pattern generation, and were able to create musical results, which at the end of the workshop were presented to an audience of 20 participants and supervisors of the computer camp.

The experience gained at the workshop shows, that the web app is accessible to the blind. All elements of the app were accessible to the screen reader. Furthermore, the teachers that participated in the workshop, confirmed the ease of use and the musical capabilities of the website. None of the teachers and supervisors of the computer camp were aware of a platform that enables blind children to create sounds so easily and playfully.

5. CONCLUSIONS

Despite a wide variety of music production software, music programming environments and web-based audio platforms, blind people are still excluded from using the computer as a musical tool, mainly because of the lack of screen reader integration, or software, that conceptually focuses on graphical elements. The difficulties of installing software on the computer add up to this. Code-based music production offers a more accessible approach, which, combined with the zero-installation benefit of web apps, could be a real solution. But even though there are many web-based music platforms, at the moment none are optimized for the use with screen reader and Braille display. The development of WELLE was a consequence of this situation and enabled a group of blind and visually impaired pupils to use their computers for the creation of music for the very first time.

WELLE is a work in progress and not yet a stable music environment for professional use, but offers blind people quick and uncomplicated access to musical drafts and playful engagements with sounds through a textual interface. The interface of WELLE will be further improved in consultation with Martin Mayrhofer, a skilled and blind computer technician, who also participated in the WELLE workshop at the Austrian Computer Camp.

6. ACKNOWLEDGEMENTS

This research is funded by the ÖAW - Austrian Academy of Sciences. Many thanks to my PhD supervisor Martin Kaltenbrunner for enabling and supporting this project. Thanks to Hansi Raber, Johannes Unger and Enrique Tomaś for inspiration and advice. And many thanks to Erich Schmid of the Institute for the Blind Vienna and to all participants of the workshops for their trust and cooperation.

7. REFERENCES

[1] S. Aaron and A. F. Blackwell. From Sonic Pi to overtone: Creative musical experiences with domain-specific and functional languages. In *Proceedings of the ACM SIGPLAN International Conference on Functional Programming, ICFP*, pages 35–46, 2013.

[2] S. Aaron, A. F. Blackwell, and P. Burnard. The development of Sonic Pi and its use in educational partnerships: Co-creating pedagogies for learning computer programming. *Journal of Music, Technology and Education*, 9(1):75–94, may 2016.

[3] A. F. Blackwell and S. Aaron. Craft Practices of Live Coding Language Design. In *First International Conference on Live Coding (ICLC2015)*, 2015.

[4] C. Correa De Oliveira. *Designing educational programming tools for the blind: mitigating the inequality of coding in schools*. PhD thesis, 2017.

[5] R. B. Gillespie and S. O'Modhrain. The Moose: A Haptic User Interface for Blind Persons with Application to the Digital Sound Studio. In *Third WWW6 Conference*, pages STANM–95, Stanford Calif., 1995. CCRMA Dept. of Music Stanford University.

[6] A. Karp and B. Pardo. HaptEQ: A collaborative tool for visually impaired audio producers. In *ACM International Conference Proceeding Series*, volume Part F1319, 2017.

[7] T. Magnusson. THE IXI LANG : A SUPERCOLLIDER PARASITE FOR LIVE CODING University of Brighton. In *Proceedings of the International Computer Music Conference*, pages 5–8, 2011.

[8] T. Magnusson. *Sonic writing : technologies of material, symbolic and signal inscriptions*. 2019.

[9] C. McKinney. Quick Live Coding Collaboration In The Web Browser. In *Nime - Proceedings of the International Conference on New Interfaces for Musical Expression Technologies*, pages 379–382, 2014.

[10] A. McLean. Making programming languages to dance to: Live coding with tidal. In *FARM 2014 - Proceedings of the 2014 ACM SIGPLAN International Workshop on Functional Art, Music, Modelling and Design*, pages 63–70, 2014.

[11] C. Morrison, N. Villar, A. Thieme, Z. Ashktorab, E. Taysom, O. Salandin, D. Cletheroe, G. Saul, A. F. Blackwell, D. Edge, M. Grayson, and H. Zhang. Torino: A Tangible Programming Language Inclusive of Children with Visual Disabilities. *Human-Computer Interaction*, 2018.

[12] J. Noble. Livecoding the SynthKit: Little bits as an embodied programming language. In *Proceedings - 2nd IEEE Working Conference on Software Visualization, VISSOFT 2014*, pages 40–44, 2014.

[13] C. Roberts, M. Wright, J. Kuchera-Morin, and T. Höllerer. Gibber: Abstractions for creative multimedia programming. In *MM 2014 - Proceedings of the 2014 ACM Conference on Multimedia*, pages 67–76, 2014.

[14] A. Tanaka and A. Parkinson. Haptic wave: A cross-modal interface for visually impaired audio producers. In *Conference on Human Factors in Computing Systems - Proceedings*, pages 2150–2161. Association for Computing Machinery, may 2016.

[15] J. Vetter. Tangible signals - Physical representation of sound and haptic control feedback. In *TEI 2019 - Proceedings of the 13th International Conference on Tangible, Embedded, and Embodied Interaction*, pages 741–744, 2019.

[16] S. Wilson. *The SuperCollider Book*. MIT Press, 2011.