

THE DECIBEL SCOREPLAYER – LEARNING FROM AND FOR USABILITY

Aaron Wyatt, Cat Hope

Monash University

Melbourne, Australia

{Aaron.Wyatt, Cat.Hope}@monash.edu

ABSTRACT

This paper outlines new developments in the Decibel ScorePlayer application for iPad, the functionality of the associated .dsz score file, and the evolution of the Decibel Score Creator desktop application that have been driven by recent user experience. This includes the introduction of an annotation layer, so that users may ‘write’ directly on to the score via the iPad screen; new methods for transferring scores to the iPad application, including the prototyping of a score server using QR code functionality; the ability to have different audio parts within a single score file and the expansion of the Decibel Score Creator’s capabilities. Each of these developments has been primarily driven by user feedback, and to a lesser extent the evolution of various operating system compatibilities. The changes have enabled the Decibel ScorePlayer to remain relevant, easy to use and a valuable tool for reading animated, graphic notation.

1. INTRODUCTION

The Decibel ScorePlayer is an iPad application designed to coordinate the reading of scores featuring predominantly graphic notation in rehearsal and performance. It features scrolling and ‘tracking’ modes for reading, and can receive drawing commands for real-time score generation [1]. Developed by the six piece Australian Decibel new music ensemble from 2012, the application is distributed on the Apple App store and is used by composers and performers worldwide. The project has recently been open sourced and has been made available on GitHub [2] under the GNU GPLv3. The Decibel Score Creator desktop application creates the unique score file format required for upload to the Decibel ScorePlayer application. These files can include multiple parts, audio, and instructions [1]. It is available for free download from the Decibel new music website [3].

2. USER FEEDBACK

Users play a dominant part in the innovation process of software development [4], and this has held true for software designed for the display of generative and animated notation. Recent developments in MaxScore, for example, have catered to user demand for features like proportional

notation [5] and the ability to view notation for MIDI clips in Max for Live [6], as well as being driven by a desire to increase user-friendliness. In the Bach library, observations of the shortcomings that users faced in Max when trying to implement complex, generative algorithms led to the development of the bell (bach evaluation language for lllls) language, simplifying the process for users [7]. Likewise, the Decibel ScorePlayer and Score Creator have been developed in close collaboration with end users.

Initially, these were the performers and composers in the Decibel new music ensemble, but once the application became available internationally via the Apple App store, a broader demographic of users became engaged. These users sometimes contact Decibel members with feedback and requests. Decibel composers also use the applications in conjunction with other performances, gathering feedback in those experiences, and large scale projects often drive new ideas, as was the case in Hope’s opera *Speechless* (2019) [8]. It has been demonstrated that high levels of innovation result when suggestions come from non technical users, that is, those without the specific programming knowledge used in the applications development [9]. These suggestions, however, need to be tempered against the practical, structural dimensions of the project and the ability to maintain it into the future. Given the large number of extant score files, created as far back as 2012, maintaining backwards compatibility has become an increasingly important task. To that end, when implementing new types of scores based on composer feedback, it remains important to consider how such features might fare into the future. Does the proposed code and the design allow sufficient flexibility given that these choices will likely need to be maintained long term? Sometimes choosing sensible defaults is sufficient to ensure this, while other times a more concerted design process is required to identify potential pitfalls that could hamper future programming directions. What follows is a discussion of a number of features in the Decibel ScorePlayer and its associated software that have been driven by user input.

3. NEW DEVELOPMENTS

3.1 Annotation Layer

The annotation layer was developed in direct response to performers expressing a need to make notes on scores in the Decibel ScorePlayer, as they would do with traditional, paper scores and with other tablet music reading applications, such as forScore. The feature first appeared in version 2.0.0. It enables users to activate an annotation layer by

touching an ‘Annotate’ button in the top right hand corner, that becomes visible while the player is stopped or paused (Figure 1). This button reveals the option to write, or remove, annotations via pencil and eraser icons (Figure 2). The primary default activation setting is the pencil icon. Touching the pencil means annotations can be made on the section of the score displayed, in red, made using a finger (Figure 3) or the purpose built Apple Pencil that can be purchased for use with the iPad. Touching the eraser icon enables the user to erase annotations in a similar fashion (Figure 4). When activated, the ‘Annotate’ button changes to ‘Done’ and when this is touched, editing is disabled and the player returns to the normal score display mode. Changes are automatically saved to the user’s iPad when no touches have been detected for more than three seconds, when the user changes where they are in the score, or when the ‘Done’ button is pressed. The annotations made are specific to the individual iPad and are not propagated across the network. This enables each performer to make their own personal notes on the score (Figure 3). While annotating, a user cannot change the playback state of the score and the ‘Play’ and ‘Reset’ buttons at the bottom left of the screen (Figure 1) are greyed out (Figure 2).

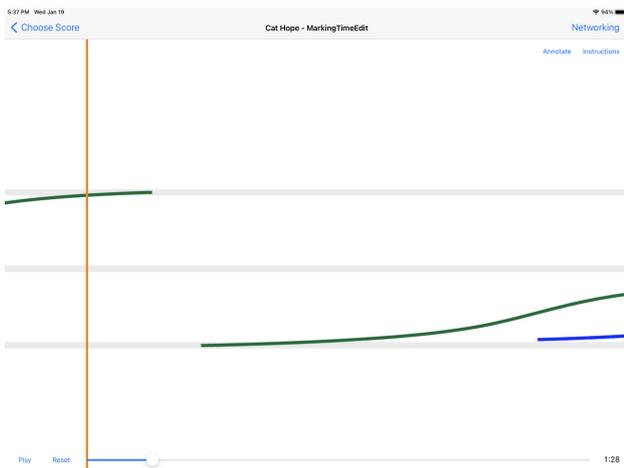


Figure 1. A screenshot showing the annotate button in the top right hand corner of the open score file on the iPad, in this case as it appears in a score by Cat Hope.

While this functionality seems relatively straight forward, implementing it was a complicated process that required a major change to the architecture of the application. In previous versions, there was a tight coupling between the user interface (UI) and the network messaging code that synchronised any connected iPads: a single module of code, `PlayerViewController.m`, was responsible for both. If the position of the score was changed on one iPad using the location scroll bar, the UI on all of the other connected iPads would instantly reflect this as soon as the network messages generated by the event were received and interpreted. While this is effective when simply displaying scores, it sets up a conflict when one user wants to write on a score while others can still control its position and playback. To resolve this, the code was split into two parts: the original module continued to control the UI while a new module,

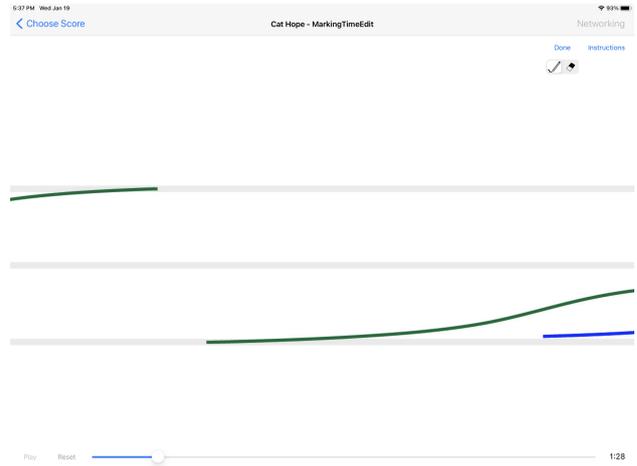


Figure 2. A screenshot showing the screen once the ‘annotate’ button has been touched. Note the greyed-out ‘erase’ icon at the top right, and the grey (instead of blue) ‘Play’ and ‘Reset’ buttons in the bottom left.

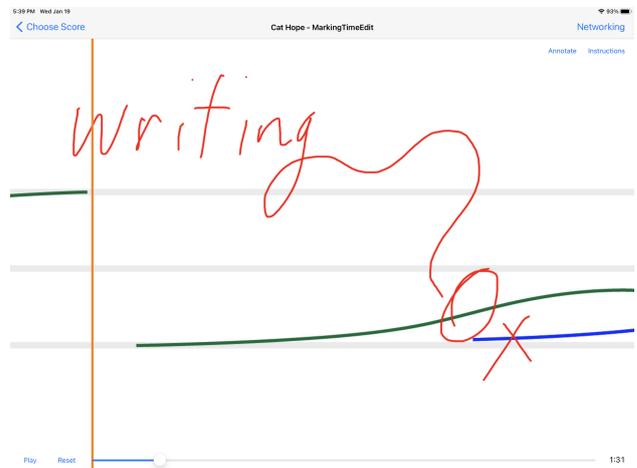


Figure 3. A screen shot showing some annotations made on the score in the ‘annotation’ layer, with the layer closed and in pause mode, ready to play or reset to the start.

`PlayerCore.m`, dealt with networking and the state of the `ScorePlayer`.

When a user enters the annotation mode these two modules are temporarily ‘detached’ from one another. The player core continues to receive network messages, storing the external state of the score’s location and playback, while the user can manipulate their local copy independently. When the annotation mode is exited using the ‘Done’ button the player synchronises with the other iPads, effectively catching up to what the rest of the ensemble has been doing. Another member of the ensemble, for instance, can start a score playing and the annotating user remains undisturbed from completing their markings until they have finished. The only case in which an interruption occurs is if another user triggers a change of score. In this case, any existing annotations are automatically saved immediately, the user is placed back in the standard viewing mode and the new score is loaded. While this isn’t an ideal situation, it is likely to be an edge case that doesn’t warrant the increased

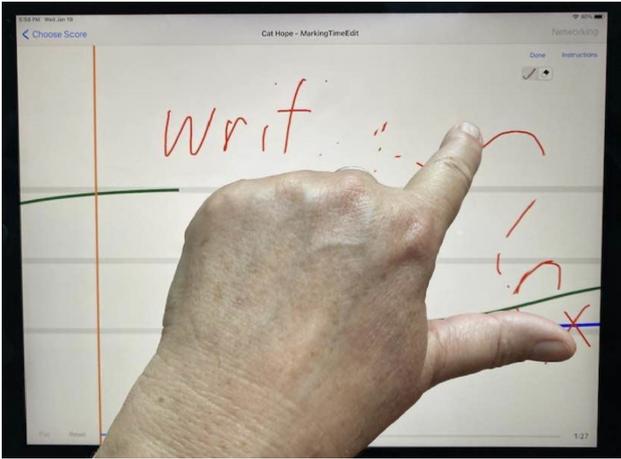


Figure 4. A photograph of a user in ‘erase’ mode using their finger on the iPad screen. Note the pencil icon is now greyed.

code complexity that would arise from delaying the load of the new score.

Another issue faced when implementing the annotation layer arises from the modular nature of the Decibel ScorePlayer. Different rendering modules control the drawing and animation of different types of scores, allowing wildly different score paradigms to share common UI and networking code [10]. Code that would work to handle the annotations on a scrolling score, the default format for the application, wouldn’t work for a score that consists of a series of timed flash cards, for instance.¹ And for some highly generative or non-linear scores, there is no real necessity for an annotation layer. To deal with this, the responsibility of managing the layout of annotations is given to the individual rendering modules. If a rendering module lacks the required methods to handle this then annotation is disabled and the ‘Annotate’ button is hidden from the user. A new module, `PlayerCanvas.m`, deals with the common task of accepting drawing input from the user. This module receives a snapshot of the currently visible section of annotations from the rendering module which can be used as a working copy. Changes are made to this image based on user input, and when a save is triggered the altered image is sent back to the rendering module to be incorporated into the saved annotations at the appropriate location.

3.2 Transferring Scores

When first launched on the App Store, the Decibel ScorePlayer shipped with four scores from composers within the ensemble. With the addition of Cat Hope’s *Her Pockets Full of Inertia* (2019) in version 2.0.2 it now has five [12]. This number provides a good balance between showcasing some of the features that the ScorePlayer offers to composers and performers, and keeping the download size of the app small enough that users aren’t dissuaded from installing it on their device. This work is the first score containing

¹ Flash cards is a Decibel ScorePlayer mode where different images appear on the screen in random or pre-determined order or timings. See works such as Cat Hope’s *Sub Arial* (2015) for an example of this format. [11]

inbuilt audio to be shipped with the application. The real power of the Decibel ScorePlayer however comes from the ability to upload additional scores to the application, and this function is commonly used. In the past, this was done by connecting the iPad to a computer with a cable, and using the file sharing capability of iTunes software on either a Mac or Windows machine to transfer files². This was clunky and not particularly intuitive, to the point that an instructional video was made by Decibel to deal with the high number of queries from users unfamiliar with the file sharing capacity of iTunes. Facility of upload has therefore been an important aspect sought by users and developer alike.

Bringing support for Apple’s AirDrop protocol to the ScorePlayer in version 1.10.3 addressed these difficulties, at least on MacOS.³ This is a feature of Apple’s operating systems that is well known to users of iOS, being used for sharing in other music applications such as the aforementioned forScore and piaScore. The ScorePlayer is able to take advantage of this feature because of the separation that exists between the distribution and playback of scores, a convenience that is not afforded to platforms that rely on web based playback mechanisms, like MaxScore’s websocket mode [5] or SmartVox [13]. (Although this approach has its own, distinct advantages.) And because this is largely handled by the operating system, no additional code needed to be written to support it. Instead, the application simply needed to let the operating system know that it was capable of handling files with the extension `.dsz` by declaring this in its metadata [14]. Specifically, in the application’s property list file. Users can then choose the score in the Finder on their computer, click the ‘share’ icon, select AirDrop from the drop-down menu that appears, and choose the iPad with the Decibel ScorePlayer installed that they want to send it to. To date, this has proven to be the easiest and most reliable method of uploading the scores to the application.

A more platform neutral way of loading scores from a network was added in version 1.9.6 in the form of a downloads window in the Score Management screen. Users can enter a URL to a score file hosted on a web server, and this would then be downloaded and would appear immediately in the application’s score selection screen. For added convenience, and to save the user from having to type in a long web address, a QR code hosted on the same web server or generated locally with freely available tools can be used to enter this. Apple’s AVFoundation framework is used to capture the camera feed from the iPad after the user grants the application permission to access it, in line with Apple’s privacy requirements. The framework includes the `AVCaptureMetadataOutput` object which can be attached to the camera’s capture session and will alert the application when a QR code is detected. The existence of these system libraries that can do the heavy lifting makes adding these sorts of conveniences to the App a very straightforward procedure which can be accomplished in relatively few lines of code.

² Use iTunes to share files between your computer and your iOS or iPadOS device: <https://support.apple.com/en-au/HT201301>

³ Windows users are still required to use iTunes to deliver scores to the application.

3.3 Score Server

Using the download manager, with its QR code reader, as a platform agnostic means of distributing scores is convenient, but requires some initial IT setup. At the bare minimum it requires a functioning web server. Beyond that basic setup, there's a lot of scope to develop additional features on the server to manage the available collection. The Decibel Score Server is a prototype of such an environment (Figure 5). It uses a number of PHP scripts to allow users to upload and then access scores. The initial upload script modifies the uploaded score so that it contains a user supplied version number and a reference to the site in its metadata. That way, if changes are required to the score, the composer can upload a new revision with a higher version number and such updates can then be discovered from within the Decibel ScorePlayer's downloads window without the need to manually revisit the website [8]. Once the upload is completed, a QR code is also generated which can be used by the Decibel ScorePlayer to directly download the initial version of the file from the server. An additional script can be used to generate a table showing QR codes and direct download links for all of the scores that have already been uploaded.

While these scripts are still a work in progress, lacking any sort of security or authentication mechanisms, there are plans to eventually release them under the GNU Lesser General Public License [15]. Porting the code to a platform like node.js in the future would also allow for a more self-contained server solution [16] that wouldn't be reliant on a more traditional web server like the Apache server that the current prototype runs on [17]. This is similar in some regards to the approach taken by the SmartVox project [18], but is much more limited in scope: the node.js server would only be responsible for the distribution of the score and would not be involved in playback or synchronisation. And like other score distribution systems, if a web presence isn't required, the server could easily be packaged onto an embedded device like a Raspberry Pi to aid with deployment [13].

Decibel Score Server

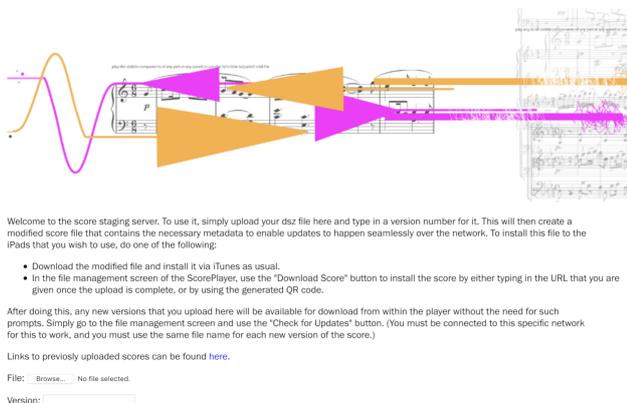


Figure 5. A screenshot of the Decibel Score Server prototype. At the bottom of the image you can see the html form elements that allow the user to select a score file and enter a version number for it.

3.4 Multiple Audio Uploads In a Single Score File

The .dsz score file for the Decibel ScorePlayer stores visual, motion, audio and other organisational information. The capacity for audio to be included in the score file is a long standing feature, having been introduced back in 2015 [19], and more recently featured in a score shipped with the application as noted above. However, some composers wanted the ability to ascribe unique audio to different parts within the score. This way, each iPad in an ensemble environment can generate it's own audio, linked to the part of that particular performer. While this has become available in the Decibel ScorePlayer with the latest release (version 2.0.6), workarounds have been used for a while. Using the Score Creator to create multiple different score files that contain exactly the same data but different audio files allowed this to be realised before it was an official feature, and the first score to use this workaround was Lindsay Vickery's work *Sanctuary* (2018) for percussion quartet [20]. The individual audio files were used to deliver unique click tracks to each of the performers, enabling them to deliver radically complex poly-temporal rhythms. As demand for this functionality has increased, work was undertaken to bring it officially into the Decibel ScorePlayer, culminating in the most recent release. The next step will be adding this to the Score Creator, so that the feature will be accessible to all composers.

As this feature was an evolution of existing features such as audio playback and multiple parts, rather than something entirely new, implementation was relatively straight forward and simply involved decoupling the code that loaded audio files from the code that initialised the audio playback object. That way, different audio files could be loaded as required, independent of the playback object's creation, which only needs to happen once. Notifying the audio player of the part change was a little more convoluted. Swiping up and down on the score has become the de facto standard when it comes to changing parts in the Decibel ScorePlayer, but in spite of this, it's still technically a mechanism that is dependent on the type of score. It was implemented in this way to allow different score types flexibility in the way that they reacted to user gestures. As a result, the UI code that passes the gestures to the rendering code has no knowledge of the connection between the swipe and a part change. To keep everything neatly compartmentalised, this means that the UI sends notification of the gesture to the current score renderer, and if that swipe corresponds to a part change the renderer notifies the UI in turn of this so that the UI can notify the audio player of the part change.

3.5 Decibel Score Creator

The Decibel Score Creator is the tool that converts images into the score files readable by the Decibel Score Player. For scrolling scores, the most common score type and the first to be implemented in the Score Creator, the image supplied by the user passes a playhead, which appears in its default form as a vertical, orange line. With the addition of an increasing number of score format options to the Score Creator, it is now possible for an image to be supplied as a 'playhead,' replacing the default vertical line.

Two works that provide different ways of using this function are Cat Hope’s *The Rupture Exists* (2020) [21] and Bergrún Snæbjörnsdóttir’s ‘SOAP’ (2020) [22]. Hope uses an image of coloured dots as points of performance in her work (Figure 6) and Snæbjörnsdóttir creates the illusion of a ring which score images pass through in her work (Figure 7). This feature enables experimentation with the concept of ‘point of performance.’

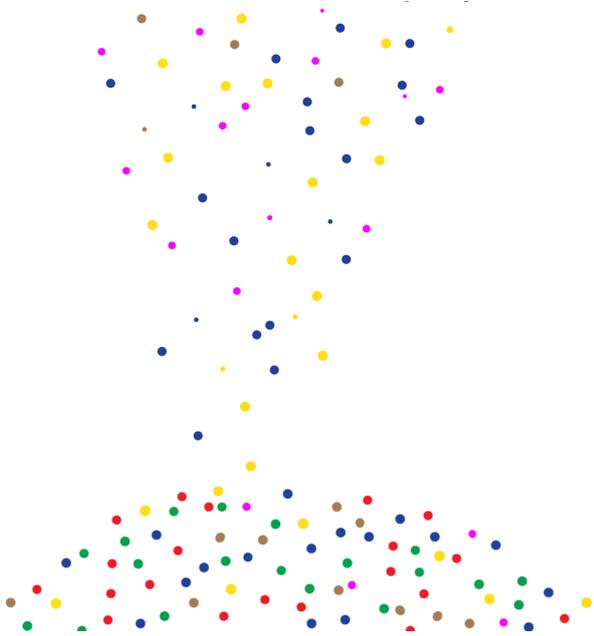


Figure 6. A screen shot from Cat Hope’s *The Rupture Exists* score, showing a cloud of dots (at the bottom of the image) acting as a ‘playhead’ for the image descending from the top of the page.

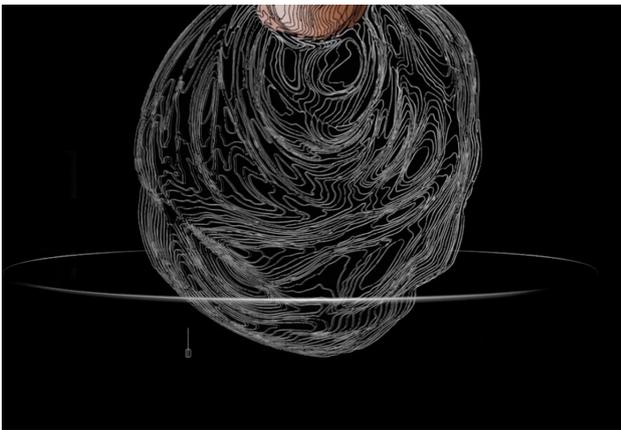


Figure 7. A screen shot from Bergrún Snæbjörnsdóttir’s ‘SOAP’ score. The silver circle is actually the playhead, which the image appears to ‘pass through’ as if three dimensional.

The score creator also now enables users to create their own versions of the ‘Talking Board’, a score format developed for a work of the same name by Lindsay Vickery and Cat Hope [23]. This is a radically different approach to the horizontal or vertical reading of the score format. It is a

roaming form where each player plays the contents of a circle that traverses the image [24]. Now, users can insert their own image to be read this way, and can define the number of circles that appear by default, currently set as up to 6, with a range of colours selectable. There are additional score types available within the Decibel ScorePlayer that are yet to be added to the Score Creator, offering potential to provide composers with an expanded array of possibilities. Before this work is undertaken though, there is a current push to port the existing features from Objective-C using Apple specific frameworks (Cocoa)⁴ to C++ using Qt⁵. This will allow the Score Creator to run on Linux and Windows as well as on MacOS with one common codebase supporting all three platforms. This broader accessibility is something that has often been raised as a concern by composers without access to Apple hardware, and while the Decibel ScorePlayer remains tied to iOS, this is an important first step towards more neutral platform support.

4. CONCLUSIONS

As this paper demonstrates, user feedback from both within the Decibel ensemble and from composers and performers outside the group has continued to drive improvements in the Decibel ScorePlayer and Decibel Score Creator applications. Some of these, like the addition of different audio files for individual parts, have been small incremental changes that have built on established features, while others, such as the annotation layer, have been major additions. Further, user feedback has seen two improved methods for score file upload developed, and new possibilities for playhead design. In each case, the aim of such developments has been to respond to users feedback, and provide more choice in how the software can be used, providing greater access to a wider variety of creative options. These options have been alternately devised by composers and performers in the ensemble, users who download the application for their own use, and by the programmer of the application. With the transition of an increasing number of the project’s components to an open source development model, it is hoped that this increased accessibility will result in more avenues for feedback between user and developer, allowing the application to continue to develop and improve.

Acknowledgments

All members of the Decibel new music ensemble have contributed to the ideas behind the developments outlined in this paper. Special thanks to the other members Lindsay Vickery, Louise Devenish, Stuart James and Tristen Parr. Also, to all Decibel ScorePlayer and Decibel Score Creator users around the world, for your feedback and ideas.

5. REFERENCES

[1] Decibel New Music Ensemble, “Decibel Score-

⁴What Is Cocoa?: <https://developer.apple.com/library/archive/documentation/Cocoa/Conceptual/CocoaFundamentals/WhatIsCocoa/WhatIsCocoa.html>

⁵Qt Framework - One framework to rule all!: <https://www.qt.io/product/framework>

- Player - Decibel.” [Online]. Available: <https://decibelnewmusic.com/decibel-scoreplayer/>
- [2] A. Wyatt, “Decibel ScorePlayer,” Jan. 2022, original-date: 2022-01-11T00:19:31Z. [Online]. Available: <https://github.com/psiborg112/ScorePlayer>
- [3] Decibel New Music Ensemble, “Decibel ScoreCreator - Decibel.” [Online]. Available: <https://decibelnewmusic.com/decibel-scorecreator/>
- [4] C. A. Voss, “The role of users in the development of applications software,” *Journal of Product Innovation Management*, vol. 2, no. 2, p. 113, 1985, publisher: Elsevier.
- [5] G. Hajdu and N. Didkovsky, “MaxScore: recent developments,” in *Proceedings of the International Conference on Technologies for Music Notation and Representation—TENOR*, vol. 18, Montréal, 2018, pp. 138–146.
- [6] —, “Maxscore: Current State of the Art.” in *ICMC*, 2012, pp. 9–15.
- [7] J. Bell, “Improvements in bach 0.8.1, a User’s Perspective,” in *SMC-Sound and Music Computing*, Toronto, 2020, pp. 18–24.
- [8] C. Hope, A. Wyatt, and D. Thorpe, “Scoring an animated notation opera—the decibel score player and the role of the digital copyist in ‘speechless’,” in *Proceedings of the International Conference on Technologies for Music Notation and Representation—TENOR*, vol. 18, Montréal, 2018, pp. 193–200.
- [9] C. A. Voss, “The role of users in the development of applications software,” *Journal of Product Innovation Management*, vol. 2, no. 2, pp. 116–117, 1985, publisher: Elsevier.
- [10] A. Wyatt, C. Hope, L. Vickery, and S. James, “Animated Music Notation on the iPad,” in *ICMC*, Perth, 2013, pp. 201–207.
- [11] C. Hope, “Sub Aerial,” Aug. 2019. [Online]. Available: <https://doi.org/10.26180/5cda650b7044b>
- [12] —, “Her Pockets Full of Inertia,” Sep. 2019, publisher: Monash University Type: dataset. [Online]. Available: <https://doi.org/10.26180/5cedc3203741b>
- [13] J. Bell, D. Fober, D. Fígols-Cuevas, and P. Garcia-Velasquez, “The BabelBox: an Embedded system for Score Distribution on Raspberry Pi with INScore, SmartVox and BabelScores,” in *International Symposium on Computer Music Multidisciplinary Research*. Marseille: Springer, 2019, pp. 345–359.
- [14] Apple, “Declaring New Uniform Type Identifiers.” [Online]. Available: https://developer.apple.com/library/archive/documentation/FileManager/Conceptual/understanding_utis/understand_utis_declare/understand_utis_declare.html
- [15] “LGPL - MDN Web Docs Glossary: Definitions of Web-related terms | MDN.” [Online]. Available: <https://developer.mozilla.org/en-US/docs/Glossary/LGPL>
- [16] Lena, “Deploy your Node.js application as a single executable,” Jul. 2020. [Online]. Available: https://medium.com/@tech_girl/deploy-your-node-js-application-as-a-single-executable-4103a2508dd7
- [17] Apache Software Foundation, “About the Apache HTTP Server Project - The Apache HTTP Server Project.” [Online]. Available: https://httpd.apache.org/ABOUT_APACHE.html
- [18] J. Bell and B. Matuszewski, “SMARTVOX-A Web-Based Distributed Media Player as Notation Tool For Choral Practices,” in *TENOR 2017*, A Coruña, 2017, pp. 99–104.
- [19] C. Hope, A. Wyatt, and L. Vickery, “The Decibel Score-Player: New developments and improved functionality,” in *ICMC*, 2015.
- [20] L. Vickery, “Sanctuary [2018] for percussion quartet, by Lindsay Vickery,” 2018. [Online]. Available: <https://lindsayvickery.bandcamp.com/track/sanctuary-2018-for-percussion-quartet>
- [21] C. Hope, “The Rupture Exists,” Sep. 2020. [Online]. Available: <https://doi.org/10.26180/5f741b3e4c4c6>
- [22] B. Snæbjörnsdóttir, “SOAP for flute, clarinet, viola, cello, prepared piano and percussion.” 2020. [Online]. Available: <https://vimeo.com/466824665>
- [23] L. Vickery and C. Hope, “The Talking Board,” Aug. 2019. [Online]. Available: <https://doi.org/10.26180/5ceb41e023365>
- [24] C. Hope and L. R. Vickery Dr, “The decibel scoreplayer—a digital tool for reading graphic notation,” in *Proceedings of the International Conference on Technologies for Music Notation and Representation (TENOR)*, Paris, 2015, pp. 59–70.