

# VibEd: A Prototyping Tool for Haptic Game Interfaces

Mathias Nordvall<sup>1,2</sup>, Mattias Arvola<sup>1,2</sup>, Emil Boström<sup>2</sup>, Henrik Danielsson<sup>2,3,4</sup>, Tim Overkamp<sup>1,2</sup>

<sup>1</sup>Department of Computer and Information Science, Linköping University, Linköping, Sweden

<sup>2</sup>SICS East Swedish ICT AB, Linköping, Sweden

<sup>3</sup>Linnaeus Centre HEAD, Department of Behavioural Sciences and Learning, Linköping University, Linköping, Sweden

<sup>4</sup>The Swedish Institute for Disability Research, Linköping, Sweden

## Abstract

Haptics in the form of vibrations in game interfaces have the potential to strengthen visual and audio components, and also improve accessibility for certain populations like people with deafblindness. However, building vibrotactile game interfaces is difficult and time consuming. Our research problem was how to make a prototyping tool that facilitated prototyping of vibrotactile game interfaces for phones and gamepads. The results include a description of the prototyping tool we built, which is called VibEd. It allows designers to draw vibrotactile patterns, referred to as *vibes*, that can easily be tested on phones and gamepads, and exported to code that can be used in game development. It is concluded, based on user tests, that a haptic game interface prototyping tool such as VibEd, can facilitate haptic game interface design and development, and by that contribute to game accessibility for persons with deafblindness.

**Keywords:** game accessibility; haptics; vibrotactile feedback; game interfaces; prototyping tools

**doi:** 10.9776/16138

**Copyright:** Copyright is held by the authors.

**Acknowledgements:** This analysis was made possible through a grant from The Swedish Post and Telecom Authority (PTS). We wish to thank Professor Emeritus Sture Hägglund for contributing to the project.

**Contact:** mattias.nordvall@liu.se, mattias.arvola@liu.se, emil.bostrom@gmail.com, henrik.danielsson@liu.se, tim.overkamp@liu.se

## 1 Introduction

Haptic feedback in games allows the player to physically *feel* the game world and the effects of their actions. However, games are today primarily visual and auditory. What happens then if you cannot see or hear; if you have deafblindness? One consequence of having deafblindness is that you are at risk of being excluded from both the experience, and the community of computer games. It would be worthwhile to make games and game culture more accessible for people with deafblindness. In this respect accessibility can be argued to be a multimodal design issue (Obrenovic, Abascal & Starcevic, 2007). Therefore, we set out to see if we could design computer games for people with deafblindness using vibrotactile feedback and simple consumer technology like game console gamepads. The Sightlence game (Figure 1), which is a translation of Pong to vibrotactile signals, demonstrated that it indeed was possible (Nordvall, 2014; Nordvall & Boström, 2013). It takes seeing and hearing players about half an hour to learn how to play Pong in the haptics-only mode. It is played with two Xbox 360 gamepads: one in the hands, and one in the lap. The paddle is controlled with the gamepad's thumb stick. The vibrations in the gamepads represent where the paddle, and the ball, is. The gamepad held in the hands has a steady vibration with low amplitude if the ball is above the paddle, and a vibration with high amplitude if it is below the paddle. If the ball is level with the paddle there is no vibration. The gamepad laying in the lap increases the vibration frequency as the ball moves towards the paddle and slower as it moves away. The gamepads also give pulses as the ball hits paddles and walls.

The problem reported in the translation of Pong into haptics was that it was time-consuming to design and develop the haptic game interface since every vibrotactile signal had to be designed through writing code. This added significant time delays between imagining, designing, testing, adjusting, and balancing the vibrotactile signals. That kind of distance hinders explorative design and iterative prototyping. Iterative design is imperative for gradual improvement of ideas, and explorative design caters for divergence in the design process. A divergent process means that more ideas are uncovered. If the design process uncovers more ideas, there is less risk of iterating on a locally optimal solution (Greenberg, Carpendale, Marquardt & Buxton, 2012). Earlier research in game design has indicated that prototypes can serve many functions. They can be used to set design goals, decompose design goals,

simulate player interaction and experience, monitor one's own design process, create shared representations, and facilitate mutual learning in the design team (Manker & Arvola, 2011). The purpose of this paper is to describe our efforts in building a prototyping tool that facilitate the prototyping of vibrotactile game interfaces, since it is time consuming to create such interfaces directly in code.

The work presented here is about vibrotactile interfaces. There are also other methods for allowing users to experience haptic sensations while interacting with technology, i.e. force feedback systems, distributed tactile displays and surface displays (Hayward & Maclean, 2007). These other technologies are not the focus of this paper.

One kind of tactile feedback is tactile icons, called *tactons*, that carry an abstract representation of a concept. Tactile icons can be constructed using parameters such as frequency, amplitude, waveform, duration, rhythm, body location and spatiotemporal patterns (Brewster & Brown, 2004; Brown, Brewster & Purchase, 2005; Hoggan & Brewster, 2007).

Many mobile phones have vibrotactile engines that can only be turned on and turned off. Because of software limitations this means that they can basically just express things in some variation of Morse code. Gamepads are usually more versatile and have a low frequency engine and a high frequency engine with variable amplitude. This makes it possible to create more advanced and subtle signals on gamepads, compared to mobile phones.

The location of a vibration on the body can also be used to express different meanings. That is, a signal on the left shoulder could represent something else than a signal on the right shoulder (Oskarsson, Lif, Hedström, Andersson, Lindahl & Tullberg, 2013; Prasad, Teale, Goldberg, Hammond, 2014; Prasad, Teale, Olubeko, Hammond, 2014). A tactile vest with several vibrotactile actuators allow for spatiotemporal patterns, such as a vibration that moves down the users back or from left to right.

Haptics in interfaces could improve game accessibility for people with deafblindness. However, the difficulty of making good haptic game interfaces stands in the way of using the haptic modality in regular development of games and applications. The problem approached in this paper is therefore how to make a game interface prototyping tool that facilitates prototyping of vibrotactile game interfaces for phones and gamepads.

## 2 Method

The research problem was approached in an iterative process of design, implementation and empirical validation with users. The process covered five iterations and was documented continuously in a decision log, in release notes, and in stage reports.

### 2.1 Design and Development Process

The design and development followed largely a human-centered process with iterations of user research, requirement definition, design solutions and user evaluations (ISO 9241-210; 2010).

#### 2.1.1 Iteration 1 – Concept Design

Platform and technical framework were chosen in the first iteration, and the concept design work set the overall direction for the design. Initial interviews with three designers and developers in game industry and haptic interfaces were conducted. The material from these initial interviews was used to develop and assess fifteen design concepts.

#### 2.1.2 Iteration 2 – Prototype A

A first prototype was developed in the second iteration focusing on creating an Android application, a server back end and a web application. Four different user interface proposals were made based on the initial concepts.

#### 2.1.3 Iteration 3 – Prototype B

The possibilities of interacting with the web application where haptic signals were developed were increased in the third iteration. Simple libraries for playing haptic signals were created for iPhone and Android. The server side was further developed to allow communication between the web-based editor and mobile phones. Prototypes of players for iPhone and Android were developed. A workshop with four graduate students in interaction design was made to assess the usefulness of the prototype for creating haptic interfaces and get feedback on ideas for future development. The results of this test were used formatively during the design process of subsequent iterations and are not described in any detail in this paper.

### 2.1.4 Iteration 4 – Prototype C

The design of Prototype C focused on connecting multiple signals to facilitate overview, connection of multiple signals to create more complex signals, support for connecting multiple output devices to be able to do spatiotemporal vibration patterns, and a new interface for the players on iPhone and Android.

### 2.1.5 Iteration 5 – Beta

The beta version of the editor was finally designed, developed, and user tested in a design workshop. That test is described in more detail below.

## 2.2 Putting the Editor to Use in a Design Workshop

The purpose of the user test of the beta version was to get formative feedback on the design of VibEd putting it to use in a user interface design workshop.

### 2.2.1 Participants

Four graduate students in human-centered systems were recruited to participate in the test of the beta version. Two were male (28 and 36 years old) and two were female (32 and 39 years old). Two were experienced programmers, while two had only some programming experience. All had training in human-computer interaction. They were selected to be able to give critique on a range of issues on user interface design and development. None had however any experience in game design.

### 2.2.2 Materials and Setting

The following materials and equipment were available to the participants during the workshop: Pen and paper, two Apple Macintosh laptop computers that ran VibEd in a web browser, one Microsoft Xbox gamepad and one Google Nexus 5 to run haptic signals on. A video camera was pointed at the table to record the workshop. Silverback 2 usability testing software was used capture the screen and record participants' facial expressions during use of VibEd. Notes were taken on paper. The workshop was held in a meeting room at the department where the participants worked.

### 2.2.3 Procedure

The four participants were asked to use the editor to create haptic signals. They worked in pairs on a design brief during the workshop, which consisted of four parts.

The first part was a ten-minute introduction to the workshop itself and to the design brief. We chose a design brief that was not game-related, since the participants did not have experience with game design. The brief was to design haptic signals for different situations in everyday life where elderly people could have use of haptic feedback. The participants had experience with interaction design and one of them also had experience with elderly users. Four situations to design for were then introduced: (1) to remember groceries to buy in the store; (2) to remember all things to bring when leaving home; (3) to get support for technical processes in public situations; and (4) to remember the keys and not lock yourself out. The participants could choose which situations to work with, and Pair A worked with situation 2, and Pair B chose to design for situation 4.

The second part of the workshop was to, in 40 minutes, develop at least four different scenarios within the chosen situation where haptic feedback could be useful. The signals were in this iteration primarily designed using pen and paper.

The third part was to, during 40 minutes, use VibEd to translate the designed signals into digital graphical signals and play them on a phone or Xbox gamepad to assess and, if needed, re-design them.

The fourth and last part of the workshop was a half-hour semi-structured discussion on how the use of VibEd worked as rapid prototyping tool for designing and assessing haptic feedback. Topics covered experiences, how they understood the editor, how it was to test signals on the phone and gamepad, and if there were features that they felt lacked.

### 2.2.4 Data Collection

During the workshop different kinds of qualitative data was collected. When the participants were designing on paper, we noted articulated thoughts on why haptic signals could work better than visual or auditive feedback, and thoughts on how to express interface semantics in haptics. When they were prototyping in the VibEd we noted how they assessed the difference between intended signals made in the editor, and how they came out on the phone or gamepad. We also noted comments on the usability of the editor. During the discussion afterwards, we noted comments on the usefulness of the editor for rapid prototyping.

### 3 Results

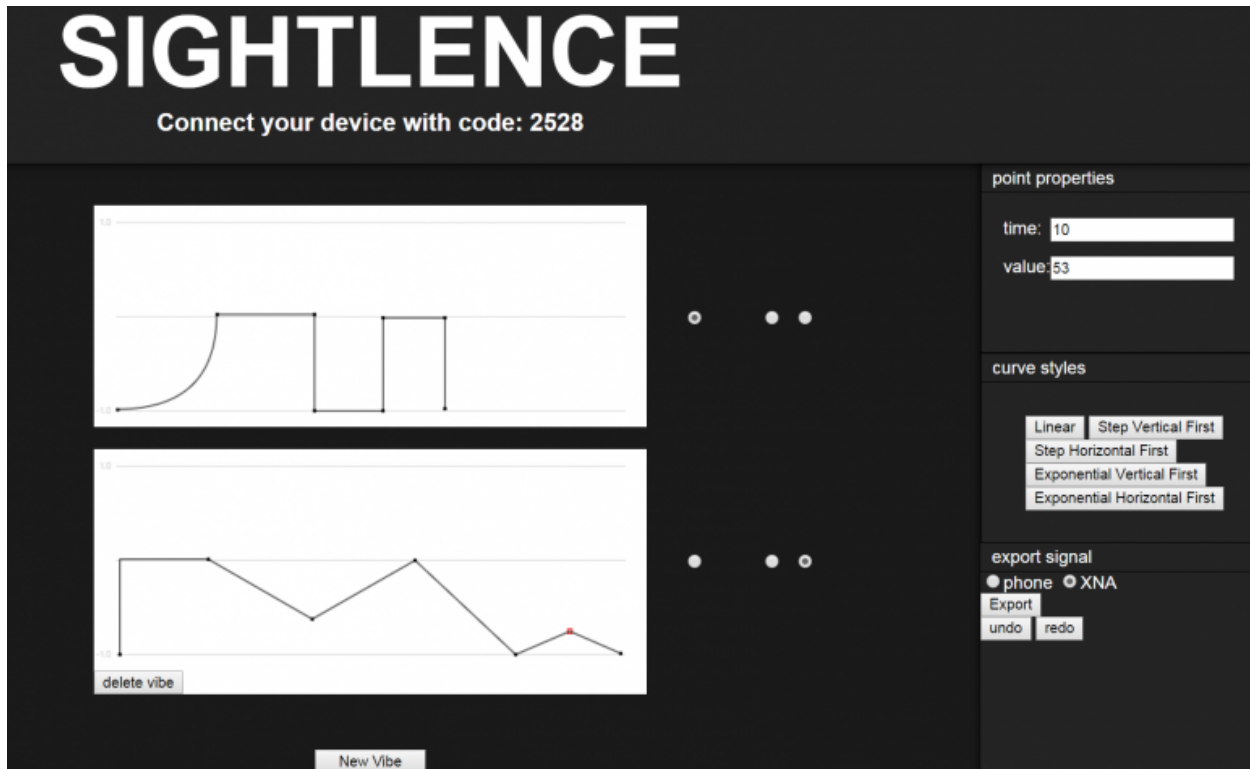
The design and development efforts resulted in the beta version of the prototyping tool called VibEd, and the design, implementation and user test of the editor will be described in this section.

#### 3.1 Results of the Design and Development Process

VibEd is a web-based editor made for easy creation of vibrotactile signals, which we call *vibes*, to be used in applications and games for any device (Figure 2).

The editor is set up much like a digital audio workstation, but for haptic vibrotactile signals instead of audio signals. There are boxes for single signals and samples that the designer or developer creates or imports, mix and match. The designer or developer then chooses player platform for the signal. Supported platforms are currently Xbox gamepad, iOS device or Android device, but custom plugins for other platforms can be set up. The signal can then be tested immediately using a player on the device.

The parameters for a vibrotactile signal are frequency, waveform, amplitude, duration and rhythm. The frequency is decided by the weight in the vibrotactile engine built into the device, which means that it cannot be controlled dynamically in code. The device also decides the waveform. The amplitude is controlled using the Y-axis in the editor. However, the amplitude cannot be controlled on iPhones or Android devices, since they only allow for starting and stopping the engine. The amplitude can therefore only be set for signals made for the Xbox gamepad. The duration of vibration signals is set in the editor



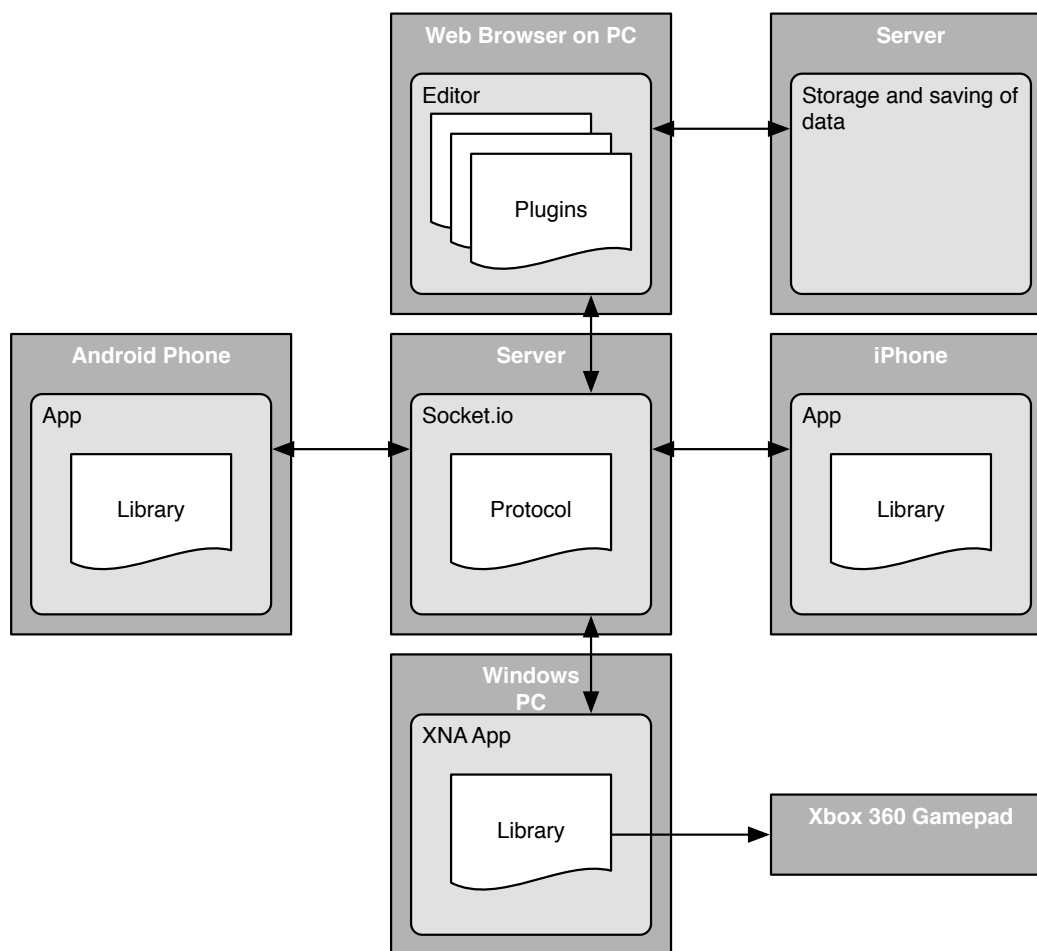
using the X-axis. The rhythm is decided by the combination of amplitude and duration.

Figure 2. Screenshot from VibEd.

VibEd is designed to allow designers to draw vibrotactile amplitude and duration patterns, which form the rhythmic patterns that we call *vibes*. These can then easily be tested on the hardware, and exported to code that can be used in development.

We have also created libraries for application development that take the vibes as input. The vibes that a user creates with the editor can therefore be downloaded and used in applications. VibEd is extendible so developers can make their own libraries for unsupported hardware devices. We have also developed playback applications for Android, iPhone and Xbox 360, that can play the vibes developed with the editor. This means that a designer or developer can create vibes in the editor and immediately try them out on their device as long as it has the playback application installed. The diagram in Figure 3 shows all components and their relation to each other. Socket.io is used for communication between applications in the system in order to support older web browsers.

The libraries and plugins make the editor extendable and allow control of different types of hardware. New libraries, and plugins can be developed to support hardware with other limitations or functionalities. Plugins are written in JavaScript and describes how a signal for a certain type of hardware is made. A plugin can also modify the original vibe's data. An example would be if a vibe made in the editor exceeds the limits of what a certain vibrotactile actuator on a hardware device can express. A library is the code that is runs on the device that plays a vibe, and it reads the data produced by the plugin for that device. This means that a plugin acts as a translator between data in the editor and data



run on the target device.

Figure 3. The architecture of VibEd.

### 3.2 Results of Putting the Editor to Use in a Design Workshop

In the final iteration of the project, four graduate students tested VibEd in a workshop. The results of that empirical validation are described in this section.

The design workshop focused on replacing or complementing visual and auditory feedback with haptic feedback. One pair of participants chose to work with the scenario of supporting elderly in not forgetting the keys and getting locked out. The other pair worked on the scenario of not forgetting things in general as you leave home. The participants could easily relate to these scenarios and give their own examples of problems to address within the scenarios.

The designers used haptics as a modality to *complement* rather than replace audio-visual information. Their discussion indicated that haptic can be used to provide added value in different situations, even though people with impaired vision or hearing most likely have most use for it. In both pairs, haptic signals were used as a way to support the memory. They created combinations of haptic and visual information, as for example a wristband that could play haptic signals while a mobile phone

provided visual information. In that situation the haptic signal was used to get the user's attention while the phone provided more detailed information.

The designers first *sketched* haptic signals by *linguistically describing* them in terms of *frequency* and *amplitude*. Enactments, such as tapping the table or vocally simulating the sound that vibrations would make, complemented the description of the signal. One of the pairs also referred to *exemplars* of other systems, like Geiger counters, sonar and parking sensors for cars. There was also a participant that used the possibility on an Apple iPhone with iOS7 to create personalized vibration patterns as well as a synthesizer application to explore and express what a particular signal would feel like. A shortcoming of these tools was that they were largely limited to frequency. The designers also explored amplitude as a variable in the haptic signals, as well as the difference between *static signals* that did not change, and *dynamic signals* that for example changed as the user moved closer to a particular object.

The editor was then used as a user interface prototyping tool to test ideas for haptic signals. The editor allowed the designers to create and play signals to get a feeling for if they would work in their interface. For example, whether or not a signal would be experienced as having a positive valence (e.g. "OK!") or a negative valence (e.g. "Not OK!"), and whether a signal would be distinctive enough or if it could be confused with an already existing signal (e.g. "New text message!"). The insights that playing a signal gave made it possible to change and optimize it. The changes made were mostly at a detailed level.

The designers did not think that VibEd was particularly appropriate for explorative design, since it took them too much time to set up a basic signal that could then be changed at a more detailed level. They also found it difficult to use the editor to demonstrate to a co-designer an idea that you got from a previous exemplar (e.g. the Geiger counter). Furthermore, there were issues with the timeline, which made it difficult to know how long the vibration engine was on or off, and how to make a signal with a repeated pattern. Finally, it was unclear how to change the curve style of a signal (linear, step, or exponential). A more general comment from the designers was that the approach used in the editor might be more familiar if you have experience from music software.

The participating designers gave a number of detailed suggestions for improvements of the editor: (a) Recording of signals or standard patterns to start from, would make it easier to create new vibration signals; (b) a grid or time intervals would make it easier to work on the details of the signals; (c) copy, cut, and paste of signals, and parts of signals, are functions that should be included; (d) a player head that moves along the timeline while the vibration signal is played would make it easier to identify where it need to be changed; and (e) working with more than two signals at the time is difficult because it requires scrolling.

## 4 Discussion

The problem approached in this paper is how to make a game interface prototyping tool that facilitates prototyping of haptic vibrotactile game interfaces for phones and gamepads. The editor we designed and developed visualizes the amplitude and durations of signals as a waveform. Created signals can be played on three platforms: iPhones, Android phones and Xbox gamepads. VibEd is an extendable web application, with libraries for vibes (i.e. vibration patterns) on the three platforms. It also includes playback applications, libraries, and plugins for the same platforms. The results of the user tests indicate that VibEd facilitates testing of vibrotactile signals. However, there are still possibilities to improve the tool. The editor does not support explorative sketching particularly well.

It is interesting to note that the designers that tested VibEd used haptics as a complement to visual and auditory signals. The earlier developed Sightlence game primarily used the haptic modality to completely replace other modalities (Nordvall, 2014; Nordvall & Boström, 2013). Treating haptics as a complement to other modalities is more common. Deciding if the haptic is a complement modality or a primary modality is a fundamental design space dimension of haptic interfaces. We also observed that the designers in the workshop used haptics to obtain the user's attention, and then provide more detailed information with a visual modality. Obtaining attention is a function of a haptic signal that can be contrasted to how the haptic in the Sightlence game is carrying more complex meaning.

Another design space dimension that was explored in the design workshop was that of static versus dynamic signals. A static signal is the same every time, while a dynamic signal is connected to a variable. VibEd does not, in its current beta version, support variables, but a future version could easily connect variables to either frequency or amplitude.

The sketching techniques used by the designers provide clues to two kinds of support that could be given in future releases. The first is to allow designers to pick different kinds of readymade signals to start from, and the second is to let them design haptic signals through audio as well.

The design workshop where VibEd was put to use show that using the editor to test an idea for a vibration signal is more efficient than writing it in code. The participants managed to create signals at the same level of complexity that was used in the Sightlence game (Nordvall, 2014; Nordvall & Boström, 2013) in only a couple of hours.

Tactile icons, tactons, can be constructed using the parameters frequency, amplitude, waveform, duration, rhythm, body location and spatiotemporal patterns (Brewster & Brown, 2004; Brown, Brewster & Purchase, 2005; Hoggan & Brewster, 2007). In VibEd, we have only made use of amplitude, waveform, and frequency, but we observed that the designers in our evaluation would have liked to have support to work with rhythm. Body location and spatiotemporal patterns are parameters that should also be considered for future design work, and releases.

The results also gave an indication of how the designers were sketching haptics. They relied to a large degree on linguistic description, but combined it with enactment of vibrations using both the voice and tapping on the table. Schön (1987) has earlier described how architects employ a spatial-action language where they draw and talk in parallel, and Arvola and Artman (2007) have shown how interaction designers make use linguistic descriptions in combination with both visual sketches, and enactments in interaction walkthroughs and improvised role-play. Exemplars such, as a Geiger counter, were also used to describe what the haptic signal could feel like. Blomkvist and Holmlid (2009) have made observations of a similar kind of use of exemplars in service design.

#### 4.1 Limitations

There are noteworthy limitations in the software development kits for the iPhone and Android platforms on how the amplitude and waveform can be controlled by an application. These limitations put serious restrictions on the expressiveness of haptic signals. They make it impossible to for example create the kind of continuous signals with variable frequency used in the Sightlence game to express relative position between the player's paddle, and the ball.

There are of course many other ways in which a prototyping tool for haptic interfaces can be designed. We have also limited our work to vibrotactile interfaces only. There are other kinds of haptic devices that are also difficult to prototype. Examples include force feedback systems, distributed tactile displays and surface displays (Hayward & Maclean, 2007). Furthermore, we have so far only looked at how to prototype signals for a single output device, the Xbox 360 gamepad device has two vibrotactile engines though. The results of this case study should therefore be transferred to other cases with some care; there is still much work that remains.

#### 4.2 Future Research

The current editor can be expanded by adding plugins for other platforms and other vibrotactile engines. Our evaluation also pointed towards many possible improvements. An interesting venue for development is how to prototype patterns of vibrations that move across multiple engines, as for example in a haptic vest (Oskarsson, Lif, Hedström, Andersson, Lindahl & Tullberg, 2013; Prasad, Teale, Goldberg, Hammond, 2014; Prasad, Teale, Olubeko, Hammond, 2014). How to prototype other forms of haptic feedback (e.g. kinaesthetic/proprioception, skin indention, temperature changes) is also an interesting venue for future research. This includes prototyping feedback on other forms of haptic devices such as force feedback systems, distributed tactile displays and surface displays.

A worthwhile and important strand of future research concerns the creation of new haptic games, and play experiences for people with and without deafblindness.

#### 4.3 Implications

The purpose of this paper is to describe our efforts in building a prototyping tool that facilitate the prototyping of haptic vibrotactile game interfaces, since it is difficult and time consuming to create such interfaces directly in code. More efficient prototyping increases the possibilities of testing different design alternatives, which is a pre-requisite for divergent design as well as iterative design.

A key to the success of haptic technologies in creating new experiences for people with deafblindness, as well as for people with full sight and hearing, is to use haptic feedback to augment the experience and make it more satisfying. That may include both haptics as a primary, and as a complementary modality in keyboards, gamepads, smartphones or other output technologies. The primary version can be found in our translation of Pong into a haptic modality, where we replaced the entire visual and sound based interface by a haptic interface (Nordvall, 2014; Nordvall & Boström, 2013). A more complementary and subtle use of haptic in computer games could for example be to have vibes that help with navigation in the game, or that indicate enemy locations.

We conclude that a haptic game interface prototyping tool, such as VibEd, can improve and simplify interface design and development. Utilizing the haptic modality better in interfaces could improve the user experience for everyone, and the accessibility for persons with deafblindness. For this to happen we need to raise awareness among practicing designers and developers on how to make use of the haptic modality, and offer them tools that allow them to do so effectively, and efficiently.

## 5 Acknowledgements

This analysis was made possible through a grant from The Swedish Post and Telecom Authority (PTS). We wish to thank Sture Hågglund for contributing to the project.

## 6 References

- Arvola, M., & Artman, H. (2007). Enactments in interaction design: How designers make sketches behave. *Artifact*, 1 (2), 106–119. doi:10.1080/17493460601117272
- Blomkvist, J., & Holmlid, S. (2009). Exemplars in service design. In *Proceedings of ServDes '09*, 19–30. Retrieved from [http://www.ep.liu.se/ecp\\_article/index.en.aspx?issue=059;article=003](http://www.ep.liu.se/ecp_article/index.en.aspx?issue=059;article=003)
- Brewster, S., & Brown, L. M. (2004). Tactons: structured tactile messages for non-visual information display. In *Proceedings of AUIC '04*, 15-23. Retrieved from <http://crpit.com/confpapers/CRPITV28Brewster.pdf>
- Brown, L. M., Brewster, S. A., & Purchase, H. C. (2005). A first investigation into the effectiveness of tactons. In *Proceedings Eurohaptics/WHC '05*, 167–176. doi:10.1109/WHC.2005.6
- Greenberg, S., Carpendale, S., Marquardt, N., & Buxton, B. (2012). *Sketching User Experiences: The Workbook*. Waltham: Morgan Kaufmann.
- Hayward, V., & Maclean, K. E. (2007). Do it yourself haptics: Part 1. *Robotics & Automation Magazine*, 14 (4), 88–104. doi:10.1109/M-RA.2007.907921
- Hoggan, E., & Brewster, S. (2007). New parameters for tacton design. In *CHI EA '07*, 2417–2422. doi:10.1145/1240866.1241017
- ISO 9241-210. (2010). *Ergonomics of human-system interaction -- Part 210: Human-centred design for interactive systems*. Geneva: ISO.
- Nordvall, M. (2014). The Sightlence game: Designing a haptic computer game interface. In *DiGRA '13*. Retrieved from [http://www.digra.org/wp-content/uploads/digital-library/paper\\_473.pdf](http://www.digra.org/wp-content/uploads/digital-library/paper_473.pdf)
- Nordvall, M., & Boström, E. (2013). Sightlence: Haptics for games and accessibility. In *FDG '13*, 406–409. Retrieved from <http://www.fdg2013.org/program/festival/sightlence.pdf>
- Manker, J., & Arvola, M. (2011). Prototyping in game design: Externalization and internalization of game ideas. In *BCS-HCI '11*, 279–288. Retrieved from [http://ewic.bcs.org/upload/pdf/ewic\\_hci11\\_s4cpaper3.pdf](http://ewic.bcs.org/upload/pdf/ewic_hci11_s4cpaper3.pdf)
- Obrenovic, Z., Abascal, J., & Starcevic, D. (2007). Universal accessibility as a multimodal design issue. *Communications of the ACM*, 50 (5), 83–88. doi:10.1145/1230819.1241668
- Oskarsson, P.-A., Lif, P., Hedström, J., Andersson, P., Lindahl, B., & Tullberg, A. (2013). Visual, tactile, and bimodal presentation of lateral drift in simulated helicopter. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting September 2013*, 57(1), 1254–1258. doi: 10.1177/1541931213571278
- Schön, D. (1987). *Educating the Reflective Practitioner*. San Francisco, CA: Jossey-Bass.
- Prasad, M., Taele, P., Goldberg, D., & Hammond, T.A. (2014). HaptiMoto: turn-by-turn haptic route guidance interface for motorcyclists. In *Proceedings of CHI '14*, 3597–3606. doi:10.1145/2556288.2557404
- Prasad, M., Taele, P., Olubeko, A., Hammond, T.A. (2014). HaptiGo: A navigational 'tap on the shoulder'. In *Proceedings of HAPTICS '14*, 1–1. doi:10.1109/HAPTICS.2014.6775543