Title: The Steam Controller as Portable Sound Synthesis

JONAS SIIM ANDERSEN, Aalborg University, Denmark

Additional Key Words and Phrases: Game Controller, Synthesis, Alternative Mappings, Portability

ACM Reference Format:
Jonas Siim Andersen. 2020. Title: The Steam Controller as Portable Sound Synthesis. 1, 1 (July 2020), 10 pages. https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

Fig. 1. A look at the Steam Controller.

The Steam Controller was a gaming peripheral released back on the 10th of November, 2015. It was Valve's [11] shot at making a controller exclusively for the large array of PC games, that their service Steam hosted. The controller was marketed with features such as having a unique interface with dual track-pads for precise movement that also contained HD haptic feedback. Other than that, it contained familiar controller features such as buttons, bumpers, and triggers [10].

Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). Copyright remains with the author(s).

DOI: https://doi.org/10.1145/1122445.1122456
Music Proceedings of the International Conference on New Interfaces for Musical Expression
NIME'20, July 21-25, 2020, Birmingham, United Kingdom

Through the Steam platform, it allows users to craft and customize their own profiles through a GUI. Buttons, trackpads, etc. could be mapped to specific events. It is even possible to map multiple events to one mapping. Additionally, the haptic feedback response is also customizable. Users can choose to feel a *strong* response or a more *faint* one.

Due to its many different functions, the controller is also able to emulate a large list of other first-party controllers such as Xbox controllers and Playstation controllers. It can even simulate a keyboard and a mouse which makes it possible to navigate your operating system using this controller. The idea of Valve releasing a piece of hardware with the intend of its high customizability begs to ask the question, which other fields outside of gaming can its interface be beneficial?

2 RELATED WORKS

2.1 the Musical Interface Technology Design (MITDS)

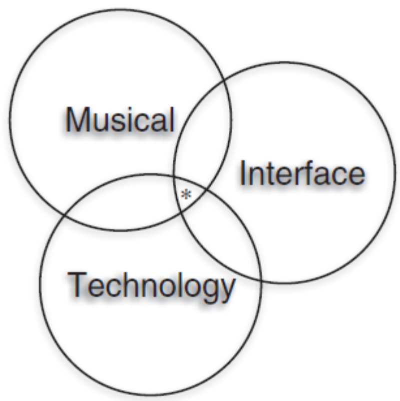

Fig. 2. The MITDS represents the overlapping field of Musical, Interface and Technology.

The MITDS [7] is a framework to dig into the development of gestural interfaces for both music performance and composition that can guide the development of new musical instruments This framework is a core concept that the Steam Controller Synthesis is built from. Although a lot of relevant concepts can be extracted from the MITDS framework, the following section will only jump into theory from the framework which I deem most relevant in correlation to the Steam Controller.

2.1.1 Rich Mapping Methodology: Mappings on an instrument is a way to describe through HCI that we humans have either a very technical or basic control over the parameters of a sound-synthesis algorithm. It's important to stress that the relationship between controls and parameters should not be simply tweaking a mathematical variable, but it should also contain musical attributes. An example could be that the strength of a stroke also has an impact on the attribute of the loudness of the signal played. MITDS considers this form of expression important for rich mapping methodology. It also allows for a higher level of skill to be developed for the instrument. The key take is to map the performer inputs to sonic outputs.

Coordination and use of muscular actions are required to play an instrument. Bowing motions require the arms to convey large movements in order to expressively play string instruments. Fingers cover more tactile responses in the sense that they cover small distances but are especially sensitive to force on the contact surface. The MITDS also acknowledges that several types of gestural inputs can be used. The following is a display of three proposed areas:

- Semiotic gestures
- Ergotic gestures
- Epistemic gestures

Semiotic gestures relate to gestures used to convey communication, ergotic gestures relate to gestures used to employ physical objects, and finally, epistemic gestures are are related to movements that give a haptic and tactile response and information.

2.2 Couchsynth

The concept of having the Steam Controller mapped as a synthesizer was already a project started back in 2018 by, the YouTube user, FlipCoder. He instead opted for the Steam Controller to be used as a device to send midi information to his custom software. A short demonstration of this implementation from FlipCoder can be seen below:

- Video: https://www.youtube.com/watch?v=kunyz46TTJc

However, according to a forum post by FlipCoder, the last update was made last year with no repositories published yet to try out the implementation [1]. Meaning that either the implementation is still being worked on or the project has been abandoned. From the posts made by FlipCoder, it seemed that his idea of mapping the controls shares the same idea of having notes be triggered by the dual trackpads, and the bumpers/triggers on the controller serve to activate different arpeggios. It's hard to get a clear idea of the modality chosen for this prototype without a demo, but it seems to function very well with radial note playing using the trackpads.

2.3 Mini Synthesizers

Fig. 3. The Korg Volca Modular Synthesizer.

DMI (Digital Music Instrument Synthesizers are usually quite large in scale, emulating pianos while hosting a body of a large range of parameters to tweak the sound with. Small/mini synthesizers also exist and Korg [6] provides a very small synthesizer with only 16 keys as well as adjustable parameters as seen from figure 3. It provides a good sense of familiarity in terms of mappings in line with other synthesizer consumer products.

[1] Forum Post made by FlipCoder: https://www.reddit.com/r/SteamController/comments/8d7iyi/couchsynth_i_heard_you_guys_like_steam_controller/?sort=new

However, the key question this project tries to look into is, that if just shrinking a larger 88 keys synthesizer down to a 16 key synthesizer is the best in terms of portability? As stated previously, it's familiar in terms of interface. But there are also a lot of buttons, mappings, settings, and parameters loaded onto this small 193 x 115 x 45 mm device.

And most importantly, while being small in size, since the interface is that of a "mini" piano the stage performance is still very static as pianos and synth DMIs have always been. They don't exactly give free embodiment for the performer on stage. Even though he can carry it by hand, the tactile response of both fingers wouldn't be optimal as well as wiring limiting the free movement.

We'll try to see if the Steam Controller's interface could be an answer to these restrictions in the design.

3 DESIGN

Fig. 4. Sketch of a finalized design.

The above image is a suggestion for a finalized prototype of the Steam Controller Synthesizer as an independent instrument that isn't exclusive to MIDI. It's proposed that the controller has two states applying the same modality to all three areas of the controller as seen from image 2. For the first state, it contains its own visual interface to allow users to get information about parameters and settings that can also be adjusted by the user (Technological). The second state of the controller is where the fun begins. This is where the musicality and gestures of the controller are enabled in order for the performer to express themselves (Musical, Interface). The following sections will go in more depth for why this design is proposed:

3.1 The Gestalt Principles

The Gestalt Principles[1] is an important fundamental framework for creating interaction design, which is also relevant to physical interfaces such as instruments. The proposed design of the Steam Controller Synthesizer has drawn strong inspiration from (but not exclusive to) the following principles:

3.1.1 Similarity: "The human eye tends to build a relationship between similar elements within a design. Similarity can be achieved using basic elements such as shapes, colors, and size."[1]

Applying this to the controller means that both the radial trackpads are influencing the frequencies played with the left track being used to play notes and the right acting both as a method to play additional notes/arpeggios/mod wheel. The circular small buttons on the controller are all related to oscillations being picked while the color-coding can aid in the tweaking for interface settings. The bumpers both control octaves as well as the back triggers enabling effect and chord modes on the controller.

3.1.2 Proximity: Simple shapes arranged together can create a more complex image.[1]

For applying this principle, in addition to similarity giving the user *touch* the law of proximity deals with how all the elements on the controller are also situated in the same groupings. All the oscillations and menu toggling is happening within the same grouping. The bumpers dealing with octave shifting and triggers enabling modes are all within the same group in the far back of the controller.

3.2 Transferring the Epistemic Gestures

As mentioned in section 2.1.1, the Epistemic Gestures explores gestures such as haptic feedback or tactile feedback. If you correlate gestures of using a game controller, this is the type of gestural theory that would be most appropriate for the medium of computer games. Quick finger movements are required for the player to beat any skill-based obstacles, while the haptics in the controller can intensify the action happening on the screen. In music, epistemic gestures can be used for a more expressive play of synthesizers as well with even some products marketing the tactile resistance of keys to make them emulate the feel of pianos. The Steam Controller Synthesizer should still deploy these tactile musical attributes, but in a way that is compatible with its interface. As mentioned in 1, the controller has built-in "HD" haptic feedback on both the grip and the trackpads. The attribute of amplification could be reflected in this feature.

4 IMPLEMENTATION

Max [9] is a high-level visual scripting language, that allows for comprehensible prototyping of audio programming. It covers a lot of features such as being compatible with audio plugins (VSTs) and so on. For its strong set of utilities, the Steam Controller Synthesis was programmed using Max.

4.1 Code Implementation

This following section will provide a few snippets into the overall system of the Steam Controller Synthesis.

4.1.1 Loading the inputs. To load in the inputs from the steam controller, we use the **[hi]** object which stands for Human Interface, and reads the input of the selected registered USB device. In order to read these inputs constantly, the hi object receives a bang every 10 ms through the poll message. This poll message has a **[loadbang]** attached, which makes sure that the program can be launched and the information from the controller starts updating automatically. This information is then sent to the **[Route]** object, which passes on each input accordingly in the patch. For example, buttons are routed to the oscillators, trackpads to the Frequency Table patch, and so on. It should be noted that the Steam Controller API is sadly very closed sourced, so in order for this patch to work it needs to be added as a game through the Steam software. A snippet of this implementation can be seen at appendix 7.

4.1.2 Frequency Table. This function is responsible for retrieving the left and right bumper information. The output of the bumper presses is being normalized to binary data using the scale function. The messages sent to the counter simply tells it to either decrease **[dec]** or increase **[inc]**. Depending on the index, the appropriate note within the certain octave (2 - 8) gets set to the message triggered by the left trackpad of the controller. A snippet of this implementation can be seen at appendix 5.

4.1.3 SelectOsc. The purpose of this function is to select an oscillator based on the input coming from the controller buttons. Based on the input pressed, it triggers an on **[1]** message to the oscillator that should get activated by the toggle object that sets the amplifier of the signal to either 1 (on) or 0 (off). It deactivates the toggle for all other signals present in the selection. The activation for Chord mode is actually a direct inheritance of this patch, but with 2 additional sines, saw, tri oscillators routed to the amplifier. The given frequency is then multiplied up to a third and a perfect fifth. A snippet of this implementation can be seen at appendix 6.

5 EVALUATION

5.1 Methodology

As of writing this paper, no actual Evaluation has been carried out using test participants, however, to thread together the topics discussed from 2, 3 and 4 I find it necessary to propose a methodology that could find a stable framework for how DMIs similar to the Steam Controller could be tested.

Barbosa et al. [5] discussed that many papers evaluating DMIs often don't employ any criteria, methods and in rarer circumstances even a goal. Therefore, I find it relevant to propose the following problem statement,

- *Can the taxonomies of DMI synthesis be translated to a portable interface utilizing a rich mapping approach with epistemic gestures that allow for more embodied freedom?*

The first step would be to employ a non-probability sample [3]. Going even further, I would employ purposive sampling to select a group of synthesis performers/composers to represent the stakeholder for my test. I would further gather background data such as

- Their expertise with synthesis instruments
- Their expertise with Game Controllers (E.g The Steam Controller or similar controller (Xbox, Playstation).

Reason why I find it relevant to ask about experience with Game Controllers, is to find any potential patterns between the motor skills of gaming and this interface, that could be interesting to investigate.

Sile [8] proposes a framework for evaluating Digital Music Instruments. From this framework, it's suggested that when the stakeholder is a performer/composer, the development of the user's skill are observed to gather any insights on their enjoyment of the interface. For playability, it's suggested that quantitative methods are used to gain insight on the user interfaces, mappings, etc.

To satisfy this framework, mixed methods would be used for the evaluation. Qualitative methods would be employed to both observe, and interview the users to gather data on the enjoyment of the interface. For the playability of the interface, quantitative methods could be used measuring System Usability Scores (SUS) [4] [2]. In order to test, *"... epistemic gestures that allow for more embodied freedom?"*, it would also be relevant to start introducing the audience as a stakeholder to see how dynamic freedom of the performer influences the experience.

At this stage, I'd deem it too early to test robustness of the interface, besides bugs that could pop up in the implemented synthesis. The criteria of internal robustness should already have been handled by Valve. However, if the interface would branch off to become an independent design and consumer product, then it's of course also worth to employ methods for that as well.

6 CONCLUSION

The Steam Controller Synthesizer shows that there is potential in making a game controller into a full instrument. It provides rich mappings and a lot of features for an interface all held with two grips. This could perhaps provide great support for stakeholders that are interested in more free expression on stage. It also provides a fun alternative way to approach tactile note playing.

The design presented covers most mappings that the stakeholder would want from traditional synthesis performance as well as design principles to hopefully make everything accessible with the user's thumbs. Most of the core features from the design were implemented, but due to the closeness of the Steam API, it was not possible to make the HD haptics more dynamic and instead opted for a more static method through the API.

The evaluation proposed a methodology to further expand and iterate the controller by utilizing user tests, since the interface holds a lot of potential for future works.

- Link to the Steam Controller Synthesizer in action: https://youtu.be/l6bJUsCiB_s

REFERENCES

[1] [n.d.]. What are Gestalt Principles? https://www.interaction-design.org/literature/topics/gestalt-principles

[2] Philip Kortum Aaron Bangor and James Miller. 2009. Determining What Individual SUS Scores Mean: Adding an Adjective Rating Scale. *Journal of Usability Studies* 4 (may 2009), 114 – 123.

[3] Thomas Bjørner. 2016. *Qualitative Methods for Consumer Research: The Value of the Qualitative Approach in Theory and Practice.* Gyldendal Akademisk. https://www.amazon.com/Qualitative-Methods-Consumer-Research-Approach/dp/8741258533?SubscriptionId=AKIAIOBINVZYXZQZ2U3A&tag=chimbori05-20&linkCode=xm2&camp=2025&creative=165953&creativeASIN=8741258533

[4] John Brooke. [n.d.]. SUS - A quick and dirty usability scale.

[5] Marcelo M. Wanderley Jeronimo Barbosa, Joseph Malloch and Stéphane Huot. 2015. What does "Evaluation" mean for the NIME community?. In *Proceedings of the International Conference on New Interfaces for Musical Expression*. NIME.

[6] Korg Corporation. [n.d.]. *Korg Volca Modular.* https://www.korg.com/us/products/dj/volca_modular/index.php

[7] Dan Overholt. 2009. The Musical Interface Technology Design Space. *Organised Sound* 14, 2 (2009), 217–226. https://doi.org/10.1017/s1355771809000326

[8] Sile O'Modhrain. 2011. A Framework for the Evaluation of Digital Musical Instruments. *Computer Music Journal* (2011), 28 – 42.

[9] Mike Sheffield. [n.d.]. *Max/MSP for average music junkies.* http://www.hopesandfears.com/hopes/culture/music/168579-max-msp-primer

[10] Valve. 2015. *Steam Controller.* https://store.steampowered.com/app/353370/Steam_Controller/

[11] Valve Corperation. 1996 - 2020. *Valve Software.* https://www.valvesoftware.com

7 APPENDIX

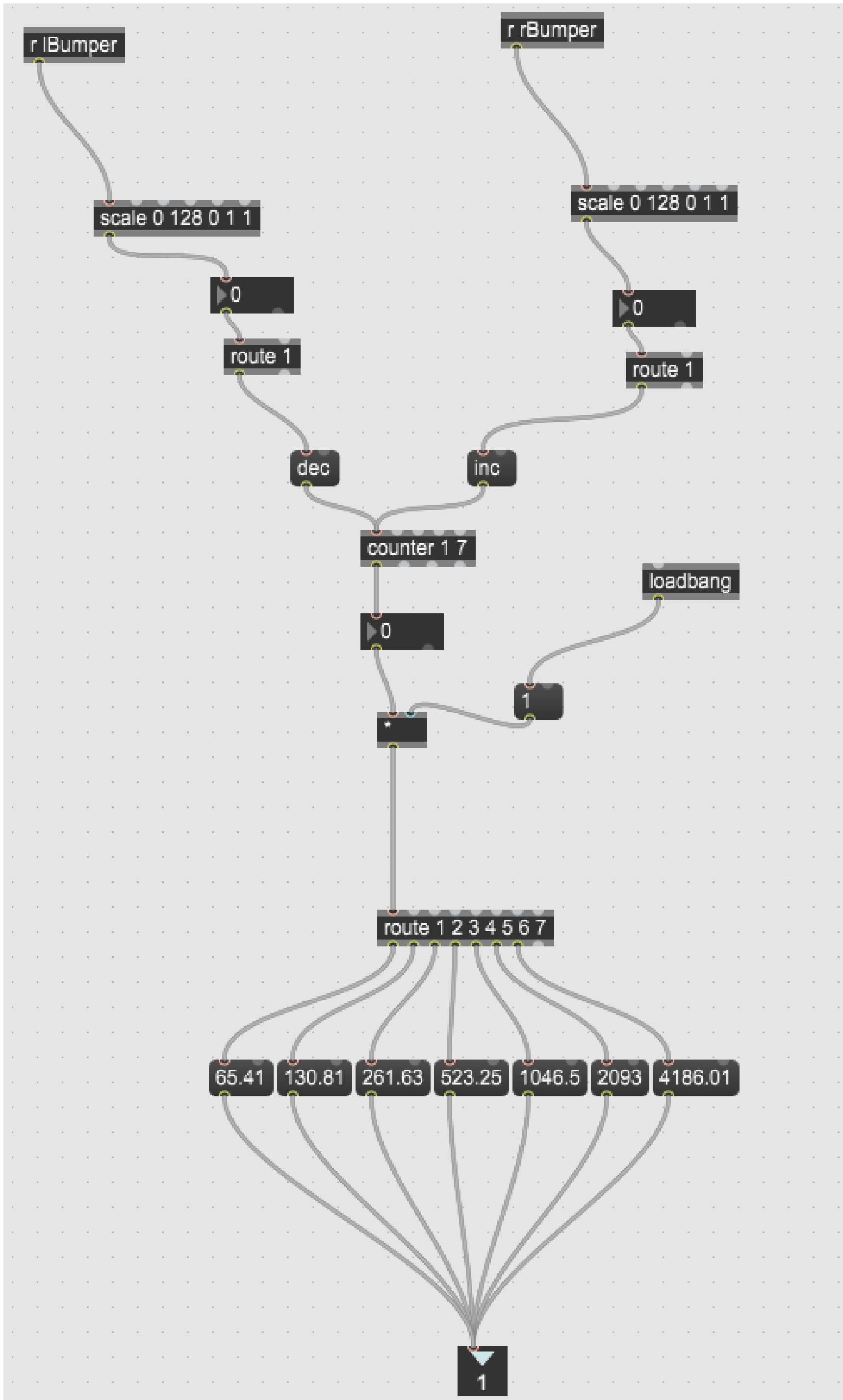

Fig. 5. The Frequency Table function.

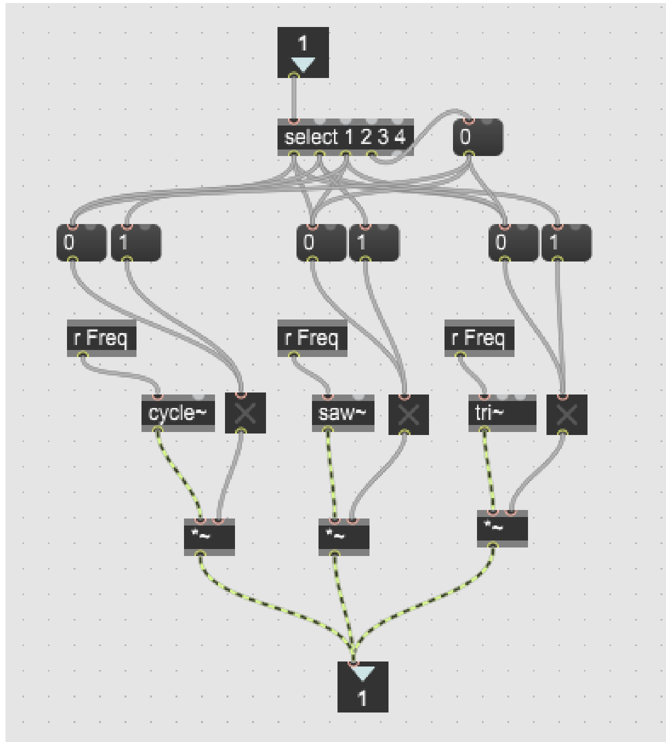

Fig. 6. The SelectOsc function.

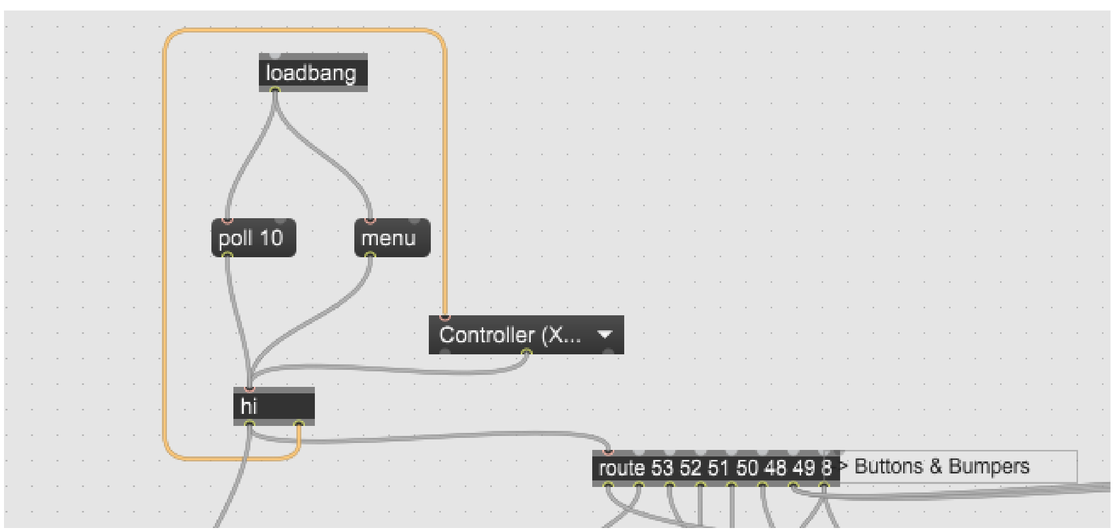

Fig. 7. Loading in the data with the hi object.