Ha Dou Ken Music: Different mappings to play music with joysticks

Gabriel Lopes Rocha
Federal University of São João del-Rei
São João del-Rei, MG, Brazil
gbr.cdc@gmail.com

João Teixeira Araújo
Federal University of São João del-Rei
São João del-Rei, MG, Brazil
teixeira.araujo@gmail.com

Flávio Luiz Schiavoni
Federal University of São João del-Rei
São João del-Rei, MG, Brazil
fls@ufsj.edu.br

ABSTRACT

Due to video game controls great presence in popular culture and its ease of access, even people who are not in the habit of playing electronic games possibly interacted with this kind of interface once in a lifetime. Thus, gestures like pressing a sequence of buttons, pressing them simultaneously or sliding your fingers through the control can be mapped for musical creation. This work aims the elaboration of a strategy in which several gestures performed in a joystick control can influence one or several parameters of the sound synthesis, making a mapping denominated many to many. Buttons combinations used to perform game actions that are common in fighting games, like Street Fighter, were mapped to the synthesizer to create a music. Experiments show that this mapping is capable of influencing the musical expression of a DMI making it closer to an acoustic instrument.

Author Keywords

DMI, video game, dual-analog, mapping strategies

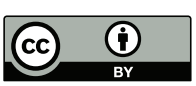
CCS Concepts

•**Applied computing** → **Performing arts;** *Sound and music computing; Computer games;*

1. INTRODUCTION

The way an artist expresses him/herself when interacting with a musical instrument and with the audience can be as important as the sonic result of a musical performance. Several researches have been done in Computer Music field with the goal of developing expressive interfaces that provides an exciting experience to the audience and stimulate the creativity of the musician in the presentation.

For a digital musical instrument (DMI), in which sound synthesis is done via software, the interaction interface serves only as a means of capturing the performer's gestures through sensors. This middle layer that links the gestures to the sound result is called mapping [3, 1, 2]. As possible strategies, we can classify the mapping as "1-to-1", "many-to-1", "1-to-many" and "many-to-many"[1]. Our project is focused on exploiting a creative mapping strategy to obtain the desired musical outcome when using joysticks as a DMI.



NIME'19, June 3-6, 2019, Federal University of Rio Grande do Sul, Porto Alegre, Brazil.

2. THE JOYSTICK AS A DMI

Joysticks are the kind of control that can be found easily at low prices, either from the internet, in gaming shops or even in popular markets. In this work, we chose to use the Double Shock B-Max Controller, which includes a Universal Serial Bus (USB) connection to be connected to a Personal Computer (PC). Currently, there are APIs in several operating system that allow interaction with events triggered by video game controls. There is even a Webaudio library that runs this function, the gamepad API. In this work, specifically regarding the implementation, the Linux library "joystick.h" was used capture the hardware events.

It was decided also to use existing synthesizers and to use the MIDI protocol, allowing the use of different synthesizers from the same device or type of mapping. Apart from a large number of interesting synths, we decided to use the LMMS (Linux Multimedia Studio) as our synthesis platform since it has a default MIDI interface to several different synths with different default inputs like the pitch, intensity and duration of the note and also LFO and ADSR envelope, which are common in several synthesizers in this DAW. Also, it is possible to variate pitch control over time in a way that make possible the execution of glissandos. With this, we maintain the advantage of being able to switch easily from synthesizer in order to explore new results and keep our focus on the mapping strategies.

One-to-one

The first formulated implementation used a simpler 1-to-1 mapping to test the control possibilities. The 1-1 mapping served as a starting point for our research. In this mapping we used the 8 main buttons, four front and four upper, to trigger MIDI notes, organized in an octave, as musical notes. The D-pad served to determine the velocity parameter, where the value 80 corresponds to the neutral position and keeping it pressed under any of the directions can change it. The resulting values increased from the bottom to the top, from left to right. The right lever was used to control the main gain. Rotating it clockwise increased it and counterclockwise will decreased it. The buttons pressed when pressing the joysticks served to raise, with the left, and descend, with the right, an octave.

The start button served as the Panic Button, sending a NOTE_OFF for all possible notes to be used in case an error occurs and some note is lost. This implementation is recurrent in all mappings. Among those who tested it, the simple fact of turning a video game controller into a musical interface served as a great draw.

One-to-many

In relation to 1-to-many mapping, it granted a macro-level control of the sound event, though it failed to give more detailed control over the parameters composing it [3].

In order to test such characteristic, we implemented the idea of using the D-pad or the left analogue not only to control the dynamics, but to change several characteristics of the resulting sound by pressing the buttons. This time, the use of diagonals was included to expand the possibilities. We started using the Envelope and LFO features present in the synthesizers. In the neutral position the sounds had short duration. As the left analog was moved clockwise, the sounds began to acquire longer duration, lift and strength. The last three positions added the LFO oscillation. Finally, the rest of the control continued with the same functions as the previous mapping.

Many-to-one

Advancing a bit more in mapping styles, the idea was to use some "many-to-1" mappings to extend control possibilities. For this purpose, instead of just using the 8 main buttons to trigger MIDI notes, it was also implemented the possibility of making combinations between them.

We created 16 buttons combination with the top four buttons, 4 with one button pressed, 6 with two buttons pressed, 4 with three buttons pressed, 1 with four buttons pressed and 1 with no button pressed. Thus, for each of the four buttons on the right side of the control, we had 16 note possibilities totaling 64 achievable notes. From the greater number of degrees of freedom achieved now, we had the possibility to control ADSR and LFO.

The right lever was used now to set up the value of parameters, selected according to the combination of the buttons pressed. When the buttons are released, the vertical position of the lever determines the value of the parameter. Thus, our instrument ended up having different forms of configuration, although the control of individual parameters could be a complex task.

The left lever, which previously had the same effect as the directional ones, started to control parameters such as the cutoff frequency and the resonance of a low pass filter, under its vertical and horizontal axes, respectively. The D-pad kept determining the velocity of the notes, however, to facilitate control, the simple press of the directional keys changed the effect, in an on/off behavior.

Many-to-Many

In our final mapping, we have sought to take inspiration on commands used in fighting games such as Street Fighter. We thus sought to exploit greater expressive capabilities of video game control with the capture of complex movements.

Here the controls became somewhat more simplified than in the previous mapping because we abandoned the idea of using the upper buttons to select notes and parameters, transferring this function to the directional buttons. When making combos by sliding fingers through the D-pad, in the classic style of Arcade, a number of parameters could be modified in the sound, be it in pitch, intensity, duration, ADSR, LFO and filters.

Aiming for greater similarity to the controls used in the game, the eight main buttons were used like in the first mapping. The "hold forward" action in games means moving toward your opponent (assuming you're on the left side of the screen, which is common for player 1 at the beginning of the fight). This resulted in stronger sounds with short envelopes and without the LFO to symbolize simple blows.

Holding the back button would mean defending or retreating your character. We chose to vary the message channel in order to activate another synthesizer with percussive sounds for that case. If any of these buttons was pressed twice in a row, it activated the LFO in a way that the sounds would be executed in a quick succession. In the game those commands would make the characters to dash.

The button Up, which causes the character to jump, meant to go up an octave while down, which would be to crouch, meant to go down an octave. These commands could be used in the same way described above with the same effect. The diagonals could be use to achieve the results of pressing the combination of directions, like up and forward at the same time. For the more complex commands an analysis was performed under the list of possible combos of the game Street Fighter IV. In this way, the most common commands used in the game were grouped and every group was responsible for generating a specific sound effect.

Also, it was possible to do glissando. Holding the joystick back and then moving it forward would result in a crescent glissando. This movement can be performed in succession to increase duration. Such movements performed in the opposite direction would result in decreasing glissando.

The quarter-circle resulted in the execution of chords that varied according to the button pressed at the end and whether it was executed clockwise or counterclockwise. If executed twice, it increased the attack and sustain of the envelope.

A half-circle would perform these chords in increasing glissando. Just as before, running them in sequence would increase their duration. To perform a decreasing glissando, simply finish the combo on the D-pad with the direction opposite to the one half circle was executed.

All commands ended by pressing one of the main buttons. Which of these had been pressed would determine the pitch of the resulting sound. To vary the octave in this way simply press D-pad or move the joystick down or up before the combos. An alternative was to use the buttons on the left and right joysticks to vary the octave in a fixed way.

3. FINAL CONSIDERATIONS

The task of developing mappings, taken as the focus of our research, brought up an unlimited range of possibilities. Finding mappings with strong semantic meanings is not a trivial task, especially in the case where there is no acoustic instrument to be taken as a reference [2]. So for our more complex mapping we made the choice to base ourselves on the universe of electronic games.

The video game control we used also has some drawbacks. Analogs are too sensitive to perform a precise movement. For the choice of mappings some physical impediments must be taken into account: it is not possible to manipulate the D-pad at the same time as the left analogue; it is not possible to reach the 4 front buttons at the same time when the right analogue is used and the simultaneous use of many buttons at the same time requires training.

Authors would like to thanks to all ALICE (Arts Lab in Interfaces, Computers, and Else) members, the support of PIDAC-Artes-Af/UFSJ and the *strictu sensu* graduate incentive program (PIPG), for the promotion of the scholarship in the UFSJ.

4. REFERENCES

[1] A. Hunt, M. M. Wanderley, and R. Kirk. Towards a model for instrumental mapping in expert musical interaction. In *ICMC*. Citeseer, 2000.

[2] A. Hunt, M. M. Wanderley, and M. Paradis. The importance of parameter mapping in electronic instrument design. *Journal of New Music Research*, 32(4):429–440, 2003.

[3] J. B. Rovan, M. M. Wanderley, S. Dubnov, and P. Depalle. Instrumental gestural mapping strategies as expressivity determinants in computer music performance. In *Proceedings of the AIMI International Workshop*, pages 68–73, 1997.