Design and Interaction of 3D Virtual Music Instruments for STEAM Education Using Web Technologies

Kosmas Kritsis[1,2], Aggelos Gkiokas[1], Quentin Lamerand[3], Robert Piéchaud[3], Carlos Acosta[4], Maximos Kaliakatsos-Papakostas[1], Vassilis Katsouros[1]

[1]Institute for Language and Speech Processing, Athena Research and Innovation Center, Greece
{kosmas.kritsis,agkiokas,maximos,vsk}@ilsp.gr

[2]Department of Informatics, University of Piraeus, Greece
{kritsisk}@unipi.gr

[3]S3AM team, IRCAM-CNRS-UPMC, France
{quentin.lamerand,Robert.Piechaud}@ircam.fr

[4]Leopoly, Hungary
{carlos.acosta}@leopoly.com

ABSTRACT

In this paper we present our current work on the development of a web-based system that allows users to design and interact with virtual music instruments in a virtual 3D environment, providing three natural and intuitive means of interaction (physical, gestural and mixed). By employing the Leap Motion sensor, we benefit from its high performance on providing accurate finger tracking data. The proposed system is integrated as a creative tool of a novel STEAM education platform that promotes science learning through musical activities. Our approach models two families of music instruments (stringed and percussion), with realistic sonic feedback by utilizing a physical model-based sound synthesis engine. Consequently, the proposed interface meets the performance requirements of real-time interaction systems and is implemented strictly with web technologies that allow platform independent functionality.

1. INTRODUCTION

Motion and gestural interaction have been studied for decades in areas including cognitive science, communication theory, linguistics and music. However, in recent years, gestural interaction has gained an increasing interest in the Human-Computer Interaction (HCI) research community, due to the reduction of cost and widespread availability of various sensors that allow the acquisition of natural gestures in great detail and in a non-intrusive manner [1]. As a consequence, the trends of 3D User Interfaces (3DUI) and Virtual Reality (VR) re-emerged [2]. Their significance in the context of virtual interaction lies in the addition of an extra layer of realism to the visual feedback [3], which is also known with the general term of *"immersion"* [4].

Even though there are multiple projects that experiment with augmented and virtual music instruments, most of them are custom made and platform dependent, while their setup is usually difficult to reproduce [5]. However, web technology standards, such as HTML5 [1], Web Audio [2] and WebGL [3] specifications, render web browsers flexible and powerful platforms for developing and designing such interactive multimedia applications. Additionally, browsers are able to "hide" the underlying hardware and provide a common framework for developing real-time and platform independent applications.

In this sense, our approach focuses on combining state-of-the-art web technologies in order to display their potentials in developing a cross-platform virtual 3D environment, where the user is capable to design and interact conveniently in real-time with virtual music instruments. The virtual environment, as described in this paper, is part of a novel STEAM (Science, Technology, Engineering, Arts, Mathematics) education platform, that aims to help students learn basic science principles, through creative and interactive music activities. For instance, a use case scenario of our STEAM platform includes the designing of virtual music instruments based on mathematical principles, which can be played in a virtual environment.

Additionally, our proposal approaches the aspect of musical expressiveness by providing realistic aural feedback based on a quite mature physical model-based sound synthesis engine, named Modalys [6]. As a motion capture system we employ the Leap Motion controller, a depth sensor that comes with a JavaScript client library which can be integrated easily in any web application.

The rest of the paper is organized as follows; Section 2 introduces related research work. Next, we describe our methodology and system architecture along with the different tools that we employed in the development of our application. In Section 4, we present the evaluation proce-

[1] https://www.w3.org/TR/2010/WD-html5-20100624/
[2] https://www.w3.org/TR/webaudio/
[3] https://www.khronos.org/registry/webgl/specs/latest/

dure based on usability testing campaigns and analyze the results. In Section 5, we discuss future developments and finally we provide some further acknowledgements.

2. RELATED WORK

In order to play a musical instrument, the musician is required to interact continuously with the instrument by performing a set of complex subtle control gestures, which are affected by certain proprioceptive sensorial feedback, including the sensorimotor, haptic, visual and auditory cues [7]. This compound aspect of musical gestures is usually required to follow a multidisciplinary approach for adapting an effective mapping strategy between the gestural information and the controlled musical parameters [8]. However, building computational models for the analysis of higher-level gestural features still remains a challenging task. Therefore, natural gestures are usually employed for controlling virtual music instruments in a way that simulates the corresponding sensory conditions, by approximating the actual interaction status of real instrumental performances. In this sense, three main research directions can be specified in the context of gesture-controlled computer music interaction [9]. *Sound synthesis control*, where the user, in real-time, modifies fundamental sound synthesis properties of the virtual instrument, such as note pitch, timber and velocity; *score-level control*, where the user alters semantic features of a predefined musical sequence; and *sound processing control*, by means of post-production events, where the user may manipulate the amount of digital audio effects or control the spatialization of sound during a live performance.

There are various research approaches that experiment with more than one of the aforementioned directions. For instance, "NexusUI" [10] is a JavaScript library that addresses both sound synthesis and processing functionalities, since it offers various touch-compatible interfaces which can be integrated in web audio applications. Another project called "Handwaving" [11] is a participatory musical system that has been developed based on web standards and takes advantage of the built-in accelerometer of the smartphone in order to recognize specific gestures and produce sounds in a given musical context. Other approaches employ electromyography sensors like the Myo Armband [4] for analyzing musical performance gestures [12] or to control sound and light spatialization [13].

Depth camera sensors, such as Leap Motion [5] and Microsoft Kinect [6], in addition to common RGB cameras, are frequently used in gesture-driven musical projects as well. Due to its high precision in hand tracking, Leap Motion has found application in projects regarding expressive real-time sound synthesis [14, 15], as well as modulation of digital effects and spatialization of sound in multi-array speakers installations [16]. Microsoft's Kinect depth camera was designed with the aim to provide skeletal body tracking data. The software called "Disembodied Voices" interprets the gestural data from the Kinect sensor and controls articulated events which are performed by a virtual choir [17]; skeletal tracking information is also employed for evaluating music conduction gestures, using various Machine Learning (ML) techniques, such as Gaussian Mixture Models (GMMs) [18], multi-modal Hidden Markov Models (HMMs) [19] and Dynamic Time Warping (DTW) [20].

Nevertheless, few proposals focus on the aspect of visual and haptic feedback. For instance, Leonard et al. [21] have experimented with custom-built haptic controllers that allowed them to simulate realistic touch feedback of keyboard and bow based instruments. Other proposals like "Revgest" [22], try to bridge the gap between the transparency of virtual instruments and the touch stimuli of physical objects. Their proposal utilizes depth camera sensors as gestural acquisition systems, in addition to projectors for augmenting visually the physical objects by displaying virtual graphics. Examples with immersive VR applications usually employ Head-Mounted Displays (HMD) and high precision marker-based motion capture systems like OptiTrack [7] for controlling virtual avatars in VR environments [23].

Going through the literature, we notice the absence of intuitive installations regarding musical virtual instruments. Most of the studies, either employ expensive motion capture systems [12, 23] and custom-built controllers [21] or platform specific software tools [5]. However, there are some applications that focus on web-based audio interaction [10, 11] but the aspect of realistic visual feedback is still missing.

3. PROPOSED METHOD

Our system comes as part of a STEAM education platform that aims on promoting the learning of sciences though music in secondary education. The novelties of our work are twofold; first, we enhance the virtual music instruments with realistic visual and audio feedback; and second, we try to provide easy access to virtual music interaction activities by strictly utilizing web technologies that allow us to develop and run the application in almost any modern Internet browser. The rest of this section describes in detail the overall architecture and the individual components of the proposed method.

3.1 Architecture Overview

An overview of the proposed architecture is illustrated in Figure 1. It consists of three main components, that will be described individually in the following subsections. The aim of the first component is to provide the environment in which the student would be able to design a 3D virtual music instrument by altering certain parameters of the instrument that affect directly its sound. The second component is the physical model-based sound synthesis engine of the virtual instrument. The physical parameters of the virtual instrument are closely simulated with those found

[4] https://www.myo.com/
[5] https://www.leapmotion.com/
[6] https://developer.microsoft.com/en-us/windows/kinect
[7] http://optitrack.com/

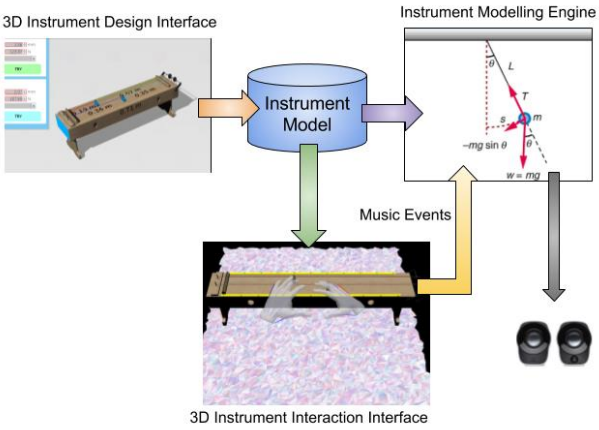


Figure 1: General overview of the system architecture.

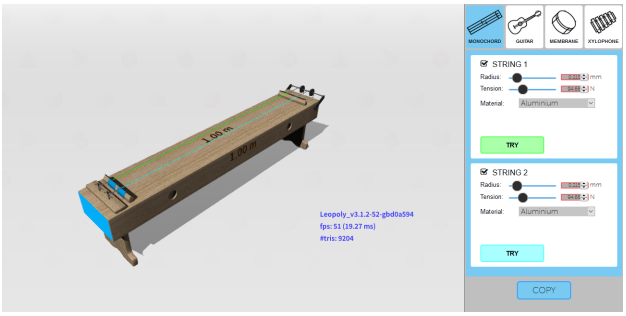
in physical, "real life", analog instruments. The third component is the 3D environment in which the user performs the virtual instrument by using the Leap Motion sensor. Furthermore, it takes into account the geometrical parameters of the designed instrument, in order to compute the corresponding interaction feedback between the captured hand motions and the virtual instrument. Gesture data are processed specifically and sent in real-time to the physical model-based sound synthesis engine, bringing "life" to the musical instrument. In the current state of the proposed system, we consider two families of instruments, namely string and percussion instruments. More specifically, as a string instrument we consider a two stringed monochord instrument with each string being divided by a bridge, resulting in a total of four string components. As percussion instruments we consider simple drum membranes with circular or square shape.

3.2 3D Instrument Design

The 3D Instrument Design environment enables designing of 3D graphical models, of predefined virtual instruments, without requiring any advanced skills in 3D graphics design. Even though the modeling software has several tools for editing 3D objects, it was essential to limit the number of available functionalities in order to prevent the users from designing deformed 3D models, while keeping only the essentials. In addition to the 3D editing tools, the user is encouraged to modify several physically-based modeling parameters of the instrument, such as string material or membrane tension. The 3D modeling environment consists of a 3D editing software with features like painting, sculpting and parametric design. The modular core engine is written in C++ and it utilizes OpenGL [8], thus enabling a cross platform architecture. The modeling engine is already ported to several platforms, including desktop, mobile and web-based versions.

In the proposed system we employed the web-port of the 3D modeling engine, that was produced with the Emscripten [9] transpiler to JavaScript, thus resulting in an

[8] https://www.opengl.org/
[9] http://emscripten.org



(a) Monochord

(b) Percussion with circle membrane

(c) Percussion with square membrane

Figure 2: Illustration of the considered 3D instrument models in the design environment. The blue elements on the 3D models indicate the area where the user can change the shape of the instrument. On the right side lies the options menu for choosing materials and modifying tension among others.

HTML5-compatible web library. As it is illustrated in Figure 2, the 3D Instrument Design environment currently provides two models of instruments: a monochord, where the user is free to enable any of the two strings, modify the length of the instrument, move any of the two bridges that divide each string into two parts and choose different string materials as well as specify its radius and tension; and a percussion which can be either circular (Figure 2b) or square shaped (Figure 2c), where the user can modify the size of the instrument and choose the material of the membrane along with its tension. In addition to the 3D editing tools, the modeling engine is also enabled to work with 3D printers and modern VR HMDs as an embedded solution.

3.3 Physical Model-Based Sound Synthesis

The core of the physical model-based synthesis utilizes a web port of Modalys [6], which unlike sound sampling of

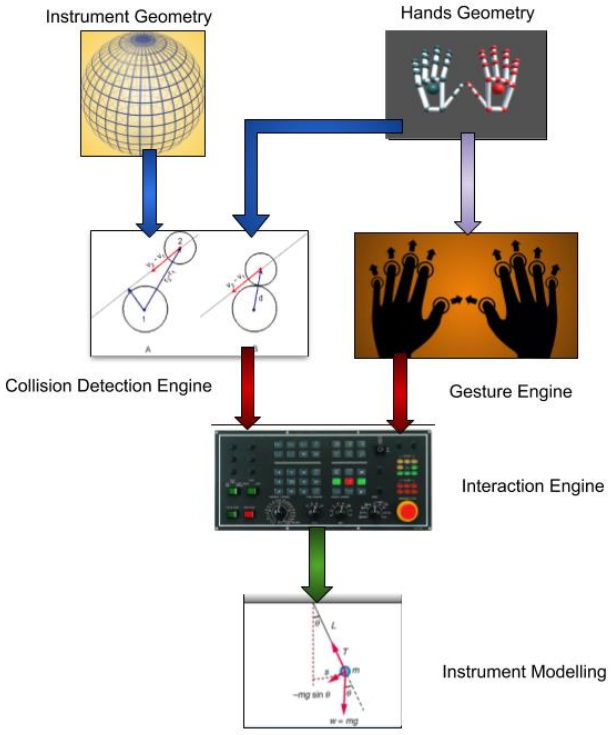


Figure 3: Overview of the 3D Interaction Engine

additive synthesis, physical model-based synthesis tries to mimic mother Nature as closely as possible. Under this paradigm, sounding objects (strings, plates, bars, membranes or tubes), which are described in physical terms of geometry, material and other properties, can be connected with one another through specific interactions, including striking, bowing, and blowing. The evolution of the physical model system is processed in real-time according to the complex physics equations that rule the corresponding phenomena of both objects and interactions, resulting to a very subtle and lively sound.

Although the model that is at the heart of our sound engine is a modal one, describing object resonances as linear combinations of modes, processing the sound synthesis of a virtual musical instrument in real-time still implies heavy CPU computation. The code base of the engine being originally in C++, it was ported to JavaScript using Emscripten transpiler in order to be able to run it in HTML5 environments. We then used various optimization strategies until we were eventually able to perform a musical instrument in a satisfactory way, without perceptible latency and audio artifacts, while rendering the instrument in the 3D environment, receiving data from the gesture module, and also using sound visualization tools at the same time.

3.4 Interacting with the Physical Instrument

The interaction with the virtual instrument contains three modes of interaction, namely *Physical Based Interaction*, *Gesture Based Interaction* and *Mixed Based Interaction*, all three being encoded in the architecture that is presented in Figure 3. The input to the proposed interaction method are the geometrical parameters of the instrument and the output data from the depth sensor (the hands' skeleton coordinates). This information is fed to the two core components of the interaction environment, the Collision Detection Engine (CDE) and the Gesture Engine (GE). The CDE detects in real time, whether the hands collide with the instrument, in addition to information regarding the collision point, speed, direction etc. On the other hand the GE takes as input only the hands data and detects if certain gestures are performed by the user. The Interaction Engine (IE) component uses as input the results of the Collision and the Gesture Engines and triggers the corresponding messages to the physical model-based sound synthesis engine according to the type of the selected interaction. Moreover, the interaction environment is equipped with a hand recording feature where the user is able to record his hand movements, and then reproduce his recorded performance. During the playback, the recorded hands appear in the virtual world according to the recorded sensor data, and triggers the CDE and the GE respectively. Furthermore, the 3D environment that is illustrated in Figure 4 has been developed using the Three.js [10] 3D library, which is a mature and easy to use, lightweight JavaScript library that offers WebGL rendering. A video demonstrator of the virtual instrument performance environment can be found in this URL [11] . Next we describe the implementation of the three modes of interaction.

3.4.1 Physical Based Interaction

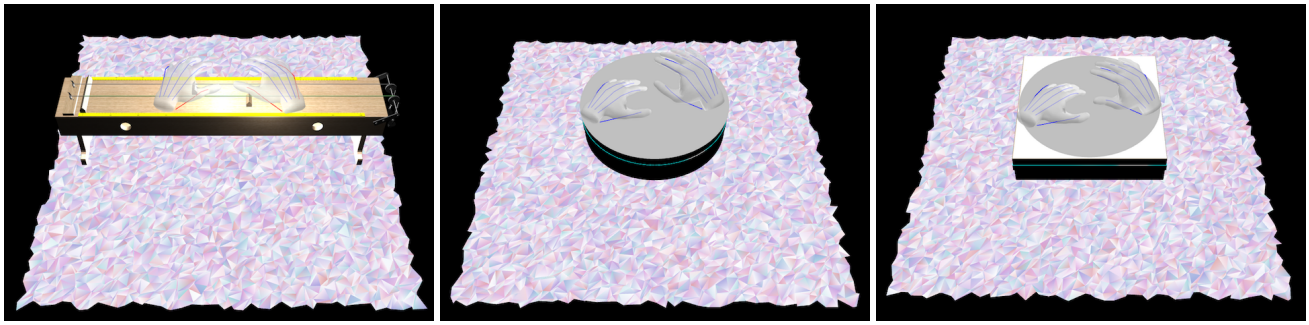
By choosing the Physical Based Interaction the user interacts with the music instruments as if they were in the physical world. This type of interaction is applied to both instrument families, string and percussion, and thus the IE takes into account only the output of the CE. Moreover, the CE receives the hands' data as a set of line segments, where each segment corresponds to a different finger. With respect to the anatomy of the human hand and the specifications of Leap Motion, each finger is represented by four continuous line segments, except thumbs, which are formed by three segments.

Music instruments on the other hand, are represented only by a set of geometrical shapes that correspond to the *intractable* components of the instrument, and not the whole 3D mesh of the instrument model. Hence, in the case of the stringed instrument, each string is represented as a line segment, while the drum membrane is simulated by a disc or a square surface according to its shape. The CDE contains internal libraries for detecting the collision events between various geometrical shapes. For instance, in the scenario of a stringed instrument with four strings, the CDE detects if any of the line segments that correspond to the fingers collide with the four line segments of the instrument. According to the collision results, the IE sends the corresponding messages to Modalys.

3.4.2 Gesture Based Interaction

In contrast to the physical interaction mode, the Gesture Based Interaction ignores the CE and the virtual instru-

[10] https://threejs.org/
[11] https://zenodo.org/record/1213560



(a) Monochord (b) Percussion with circle membrane (c) Percussion with square membrane

Figure 4: Examples of the considered musical instruments in the 3D interaction environment.

ment is performed solely with finger gestures. Moreover, the position of the fingers does not play any active role in the context of interacting with a virtual 3D avatar, rather than being visible to the camera's Field-Of-View (FOV) in order to recognize if the user performs specific gestures. Currently, we consider a certain type of gesture, the finger tapping, which is used to control the four strings of a virtual instrument. Each finger but the thumb of the right hand, corresponds to one of the four strings of the instrument, while on the left hand, the tapping gestures mimic the control of guitar's fretboard. Moreover, if all fingers of the left hand are "open", then the produced notes by the right hand correspond to the fundamental pitch of the strings. Consequently, if the user taps the index finger it raises the pitch of each string by one semitone, the middle to one tone, the ring finger by three semitones and the pinky by four semitones. The finger tap gesture is recognized by using a simple heuristic approach, that considers the angle between the metacarpal bone of each finger along with its direction. If the angle exceeds a specified threshold, the finger movement is recognized as the tapping gesture. To adapt the recognition to different users, there is a calibration procedure which allows manual adjustment of the threshold for each finger individually.

3.4.3 Mixed Interaction

In the Mixed Interaction the IE considers both the outputs of the CE and the GE. It can be considered similar to the physical interaction, but with the restriction that the finger tapping gesture is performed to hit the string. If a finger gesture is detected, then the CE is activated solely for that finger, and only if it collides with a string, then the sound synthesis engine is triggered.

4. EVALUATION AND USABILITY TESTING

The evaluation of the 3D environment was mainly focused on the overall user experience of the interface, according to the feedback received from students between the ages of 13 and 16 during usability testings in the context of the *iMuSciCA* project [12] . These tests have been carried out at schools in three countries, namely Belgium (11 students), France (10 students) and Greece (29 students), where participants have been provided with a use-case scenario and its corresponding questionnaire, while the researchers that were conducting the tests were engaged in informal conversations with the students and keeping notes about remarks that were not accounted for in the questionnaires.

The usability scenario that was handed to the participants incorporated simple instructions for performing tasks and playing music by navigating in the User Interface (UI) of the STEAM platform and using the provided tools; the intention was to allow participants to explore the gestural environment and reveal potential weaknesses of the UI or misinterpretations related to the functionality of the tools. The questionnaires incorporated questions regarding whether the user had an enjoyable experience, if the UI elements of the environment were easily accessible and intuitive, as well as whether the setup was audio-visually pleasing. The evaluation presented in this paper concerns only the feedback received for the instrument interaction environment, while readers interested on the overall scope of the STEAM project and its technical as well as educational aspects are referred to the Work Packages documents hosted at the *iMuSciCA* home URL which can be found as a footnote below.

In brief, the usability scenario guided the user through successive steps to select instrument and different modes of interaction at different stages, while at the same time exploring other accompanying tools in the environment. The main goal was to evaluate the free and gesture-based interaction modes, while the additional tasks included zooming in and out the instrument view, in addition to recording and playing back their performance (actual movement of hands). A translated print of the scenarios was given to each participating student in their language, while the coordinating researchers were assisting and giving instructions correspondingly to each step. After carrying out all the tasks included in the usability scenario, the students had to respond to the questions of an on-line translated questionnaire; the included questions regarding the instrument interaction of the environment are shown in the following list:

Q1: How familiar are you with playing a musical instrument? (*Likert scale*: **0:** I don't play any instrument, **1:** I consider myself a musician)

Q2: How would you rate the overall usability of the interface? (*Likert scale*: **0:** Not usable at all, **1:** Very

[12] http://www.imuscica.eu/

Table 1: Compressed representation of answers in the questionnaire by all participating students (total of 50) in a normalized scale from 0 to 1.

Question	Mean	Std	Skewness	Kurtosis
Q1	0.42	0.32	0.52	-0.75
Q2	0.73	0.22	-0.42	-0.58
Q3	0.68	0.24	-0.35	-0.64
Q4	0.71	0.24	-0.98	1.19
Q5	0.55	0.28	-0.38	-0.58
Q6	0.62	0.29	-0.21	-1.03
Q7	0.73	0.36	-0.94	-0.35

easy to use)

Q3: How would you rate the free interaction mode with the instrument? (*Likert scale*: **0:** Very difficult, **1:** Very easy)

Q4: How would you rate the sound quality of the instrument? (*Likert scale*: **0:** Very poor, **1:** Excellent)

Q5: How would you rate the responsiveness of the instrument? (*Likert scale*: **0:** Very poor, **1:** Excellent)

Q6: How would you rate the gesture based interaction mode with the instrument? (*Likert scale*: **0:** Not good at all, **1:** Very good)

Q7: Was it easy to record the gestures of the performance? (*Multiple choice*: **0:** No, **0.5:** Not sure, **1:** Yes)

Question 7 included multiple choice answers, while all other questions were answered in a Likert scale from 1 to 5. The descriptions in the list shown above correspond to the numeric values presented in the "compressed" answer representation of Table 1; therein, the answers are encoded in normalized values from 0 to 1. Regarding multiple choice questions, numeric values have been attributed to possible answers: negative and positive responses are assigned the values 0 and 1 respectively. The Likert scale responses have been normalized by linearly transforming the answers of the range 1 to 5, to the range 0 to 1.

Table 1 presents the mean values (0 and 1 stand for the negative and positive ends), the standard deviation (which indicates the agreement among participants), the skewness (positive values show a skew towards the negative end and vice versa) and the kurtosis (greater values indicate sharper edges towards the mean). The students gave positive answers for the general usability and appearance of the environment (Q2). Regarding free interaction with the virtual instrument, students had moderate to positive opinion (Q3), while for gesture based there were difficulties in understanding how they should perform; however their answers were comparable for both interaction modes (Q6), since, besides the difficulties, they found gesture interaction more interesting, based on informal feedback given to the researchers who conducted the tests. Most students had a positive opinion about the sound quality (Q4) but answers were controversial about the "responsiveness" of the instrument (Q5). This misalignment might be due to different hardware setups used in each validation setup (CPU, memory, video card), which can greatly affect the performance of the environment. Students found it relatively easy to record their gestures and the audio of a session (Q7).

5. CONCLUSION AND FUTURE WORK

In this paper we presented a web-based interface that utilizes the Leap Motion sensor for real-time interaction with virtual music instruments within a 3D environment, in the context of STEAM education. The architecture of our proposal consists of three core modules: an instrument design environment where the user is able to alter physical features of two musical instrument models; a music interaction environment where the instruments can be performed according to three different interaction modes; and a physical model-based sound synthesis engine that produces realistic sound. Preliminary results involving students from three different EU countries are promising, since their answers regarding the overall usability and design of the environment where positive.

However, our system is still under development and future work includes multiple tasks. First we plan to add two more virtual instrument models including a xylophone and a bell. Next we are going to optimize the overall performance of the 3D environment by improving the GE with more convenient interaction mappings, update the 3D modeling engine and reduce the latency of the sound feedback by portings to WebAssembly [13] as this standard allows better performance in browser environments. Furthermore, a VR version of the environment is currently under development.

Acknowledgments

This work has been fulfilled in the context of the iMuSciCA project, which received funding from the European Union's Horizon 2020 research and innovation program under the grant agreement No 731861. We would also like to thank the students that took part in the usability testing campaigns.

6. REFERENCES

[1] S. Trail, M. Dean, G. Odowichuk, T. F. Tavares, P. F. Driessen, W. A. Schloss, and G. Tzanetakis, "Non-invasive sensing and gesture control for pitched percussion hyper-instruments using the kinect," in *Proceedings of the International Conference on New Interfaces for Musical Expression, NIME 2012*, 2012.

[2] M. Hachet, "3d user interfaces, from mobile devices to immersive virtual environments," Ph.D. dissertation, Université Sciences et Technologies-Bordeaux I, 2010.

[3] S. Serafin, C. Erkut, J. Kojs, N. C. Nilsson, and R. Nordahl, "Virtual reality musical instruments: State of the art, design principles, and future directions," *Computer Music Journal*, vol. 40, no. 3, pp. 22–40, 2016.

[13] `http://webassembly.org/`

[4] M.-L. Ryan, *Narrative As Virtual Reality: Immersion and Interactivity in Literature and Electronic Media*. Johns Hopkins University Press, 2001.

[5] R. Medeiros, F. Calegario, G. Cabral, and G. Ramalho, "Challenges in designing new interfaces for musical expression," in *Design, User Experience, and Usability. Theories, Methods, and Tools for Designing the User Experience - Third International Conference, DUXU 2014, Held as Part of HCI International 2014, Proceedings, Part I*, 2014, pp. 643–652.

[6] R. E. Causse, J. Bensoam, and N. Ellis, "Modalys, a physical modeling synthesizer: More than twenty years of researches, developments, and musical uses," *The Journal of the Acoustical Society of America*, vol. 130, no. 4, pp. 2365–2365, 2011.

[7] A. Bouënard, M. M. Wanderley, and S. Gibet, "Gesture control of sound synthesis: Analysis and classification of percussion gestures," *Acta Acustica united with Acustica*, vol. 96, no. 4, pp. 668–677, 2010.

[8] F. Visi, R. Schramm, and E. Miranda, "Gesture in performance with traditional musical instruments and electronics: Use of embodied music cognition and multimodal motion capture to design gestural mapping strategies," in *Proceedings of the 2014 International Workshop on Movement and Computing*, ser. MOCO '14. ACM, 2014.

[9] U. Rosselet and A. Renaud, "Jam on: a new interface for web-based collective music performance," in *Proceedings of 13th International Conference on New Interfaces for Musical Expression, NIME 2013*, 2013, pp. 394–399.

[10] B. Taylor and J. T. Allison, "Gesture capture, processing, and asynchronous playback within web audio instruments," in *Looking Back, Looking Forward: Proceedings of the 41st International Computer Music Conference, ICMC 2015*, 2015.

[11] G. Roma, A. Xambó, and J. Freeman, "Handwaving: Gesture recognition for participatory mobile music," in *Proceedings of the 12th International Audio Mostly Conference on Augmented and Participatory Sound and Music Experiences*. ACM, 2017.

[12] Á. Sarasúa, B. Caramiaux, A. Tanaka, and M. Ortiz, "Datasets for the analysis of expressive musical gestures," in *Proceedings of the 4th International Conference on Movement Computing*, 2017.

[13] B. D. Donato, J. Dooley, J. Hockman, S. Hall, and J. Bullock, "Myospat: A hand-gesture controlled system for sound and light projections manipulation." Shanghai Conservatory of Music, 2017, pp. 335–340.

[14] J. Françoise, O. Chapuis, S. Hanneton, and F. Bevilacqua, "Soundguides: Adapting continuous auditory feedback to users," in *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems*, ser. CHI EA '16. ACM, 2016, pp. 2829–2836.

[15] E. Togootogtokh, T. K. Shih, W. G. C. W. Kumara, S.-J. Wu, S.-W. Sun, and H.-H. Chang, "3d finger tracking and recognition image processing for real-time music playing with depth sensors," *Multimedia Tools and Applications*, 2017.

[16] L. Hantrakul and K. Kaczmarek, "Implementations of the leap motion in sound synthesis, effects modulation and assistive performance tools," in *Music Technology meets Philosophy - From Digital Echos to Virtual, Ethos: Joint Proceedings of the 40th International Computer Music Conference, ICMC 2014, and the 11th Sound and Music Computing Conference, SMC 2014*, 2014.

[17] M. Mandanici, A. Rodà, and S. Canazza, "A conceptual framework for motion based music applications," in *Proceedings of 2nd IEEE VR Workshop on Sonic Interactions for Virtual Environments*, 2015, pp. 9–13.

[18] Á. Sarasúa, B. Caramiaux, and A. Tanaka, "Machine learning of personal gesture variation in music conducting," in *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, 2016, pp. 3428–3432.

[19] J. Françoise, N. Schnell, and F. Bevilacqua, "A multimodal probabilistic model for gesture-based control of sound synthesis," in *Proceedings of ACM Multimedia Conference, MM 2013*, 2013, pp. 705–708.

[20] R. Schramm, C. R. Jung, and E. R. Miranda, "Dynamic time warping for music conducting gestures evaluation," *IEEE Transactions on Multimedia*, vol. 17, no. 2, pp. 243–255, 2015.

[21] J. Leonard, C. Cadoz, N. Castagné, and A. Luciani, "A virtual reality platform for musical creation," in *10th International Symposium on Computer Music Multidisciplinary Research*, 2013.

[22] F. Berthaut, C. Arslan, and L. Grisoni, "Revgest: Augmenting gestural musical instruments with revealed virtual objects," in *Proceedings of International Conference on New Interfaces for Musical Expression, NIME 2017*, 2017.

[23] K. Kilteni, I. Bergstrom, and M. Slater, "Drumming in immersive virtual reality: The body shapes the way we play," *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, no. 4, pp. 597–605, 2013.