

Sonic Balls

Fabian S. Klinke

2021-08-31

An interactive playground for children.

Abstract

Sonic Balls is an interactive sound installation conceptualized for children playing with and throwing balls at each other. Its goal is to encourage kids of younger ages to explore the sonic landscape generated through the movement of the balls.

Each movement parameter is linked to a sonic event and each ball to a specific instrument within the sound design. The children can then discover which parameter and ball correlates with which sonic event through playful interaction with the installation and each other.

Concept

Sonic Balls is designed as a playground for children that allows them to primarily play with each other and, through that play, explore the sonic landscape in which they move around. Each of the three children is handed a ball linked to one instrument of the music. The turning and moving of the ball, as well as collisions of the ball with other objects,

are the real-world events that get translated into sonic triggers or changes. Which ball and movement are linked to which sonic outcome will not be disclosed to the children.

To further increase the responsiveness of the installation, the playback speed of the music depends on the average velocity of all the balls combined. A hectic and fast play with all balls moving will lead to fast music and vice versa.

Theremin Playground

The idea came to be as a multi-user three-dimensional adaptation of a theremin, an electronic instrument that can be controlled without touching it. When playing the theremin, the movements of your hands around and above two metal antennas control the instrument's pitch, volume, and sound. This, especially for the inexperienced user, can easily and quickly lead to unexpected sounds. This unexpectedness makes it more interesting to explore the sounds the instrument generates.

Between the player and the instrument emerges a level of interactiveness. It arises through the exploration of unknown sounds and the device providing them with sometimes surprising twists. The machine is not limited by any buttons and keys, consisting only of thin air and movement, so there is an almost limitless possibility within this space, since there are no predefined input values.

We wanted to adapt this concept, expand it and make it accessible to children. Since the movement of the balls controls the sounds of the installation, the users can, like with a theremin, make almost unlimited changes to the instrument they're controlling. The big difference here is that the changes are not locked to the absolute position in space but the relative acceleration and orientation of each ball. This makes it easier to implement technically while also further detaching the visually apparent from the sonic results.

Expected Results

The highly automated approach where the only control we as designers have is what to connect to where and how much the raw data influences the parameter will allow for more surprising results since we can never fully predict the children's actions, especially not multiple. The music hence will almost always be new to everyone involved at all times during the installation. The sonic landscape will seem to some extent almost semi-random but in a strictly deterministic way.

Through this approach, the children will only get some glances at the connection between their actions and the reactions of the installation, creating a certain level of mystery that can intrigue the childlike curiosity to try to explore Sonic Balls.

Types of Interaction

Child <-> Child The most fundamental of all interactions is between the children themselves. As this installation is targeted towards children, especially younger children, it is important to always encourage a certain playfulness within the setup.

The project aims to not only have kids throwing some balls but to motivate them to act voluntarily and on their own in exploring the game with their friends or strangers. To even just for a moment build a human connection with another, maybe even forgetting the existence of the installation in which they act, allowing Sonic Balls to sonify the natural occurring interactions among the children.

Child <-> Installation While most of the more apparent interactions will happen within the interplay of the children with the reactions of the sound design, the individual user can also explore how their movements of a single ball influence the sonic environment.

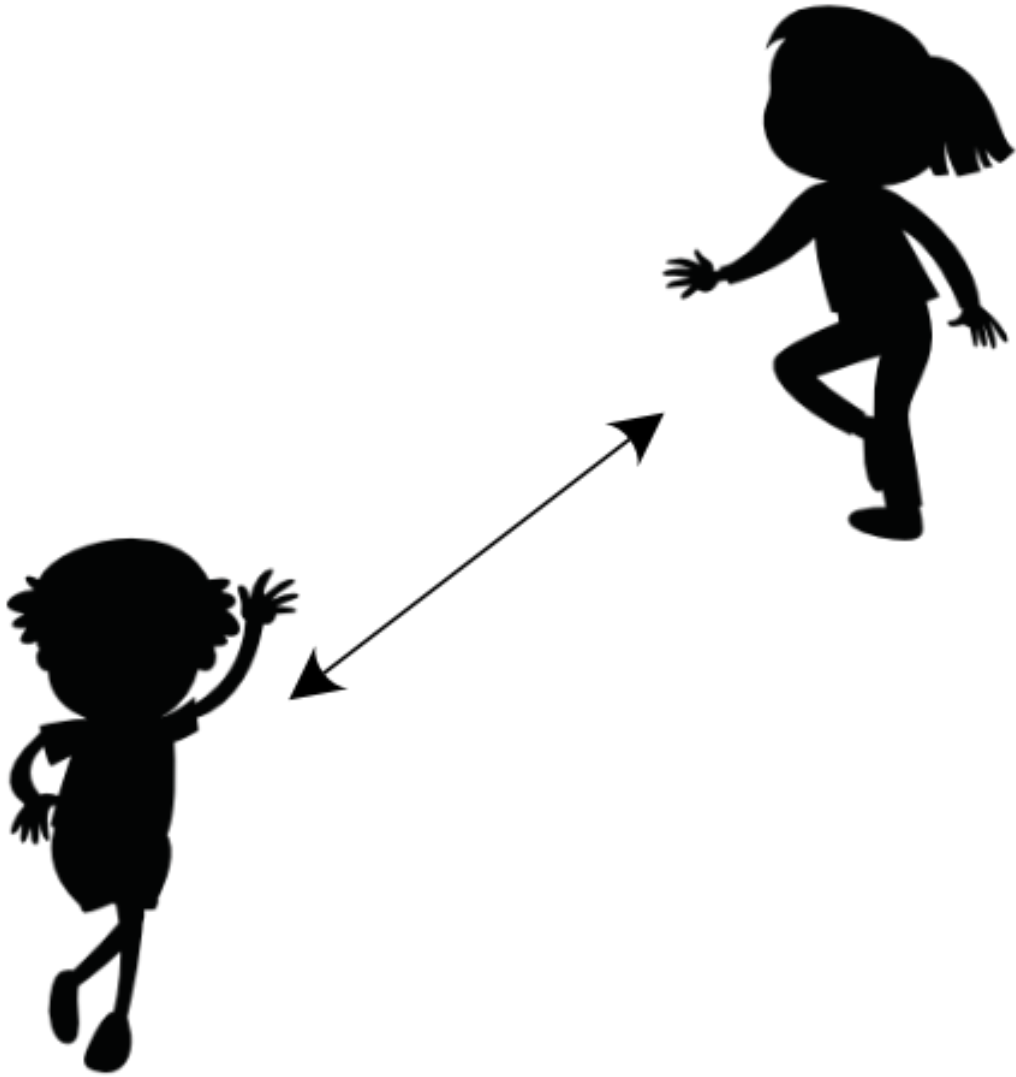


Figure 1: Child $\leftrightarrow$ Child

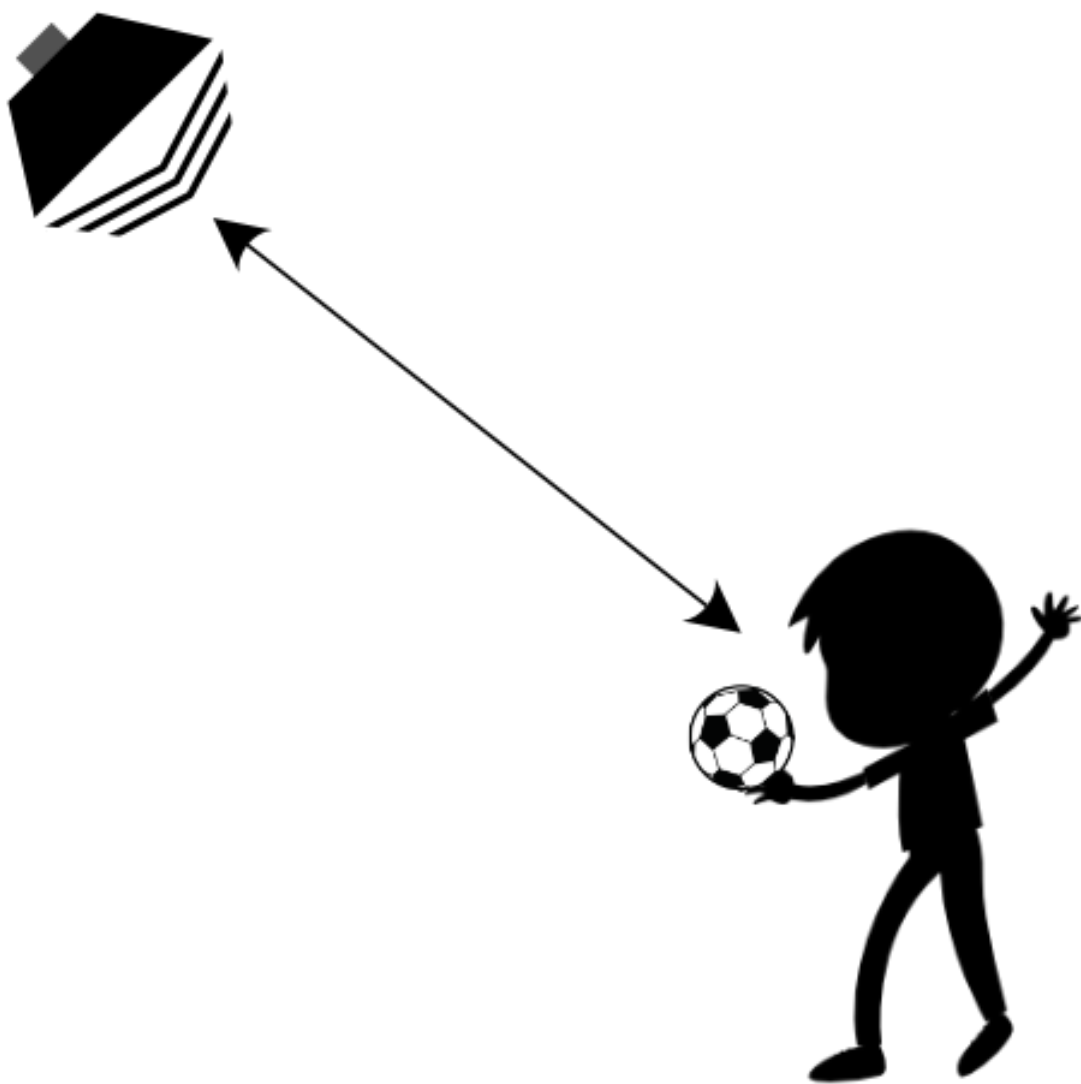


Figure 2: Child $\leftrightarrow$ Installation

Since the parameters of each ball are directly linked with the installation, a child can explore the sonic responses by turning the ball or doing small or erratic movements, observing and maybe even trying to figure out the relationships between their actions and the music. It would be interesting to see how the installation reacts if all participants only interact with themselves and not each other.

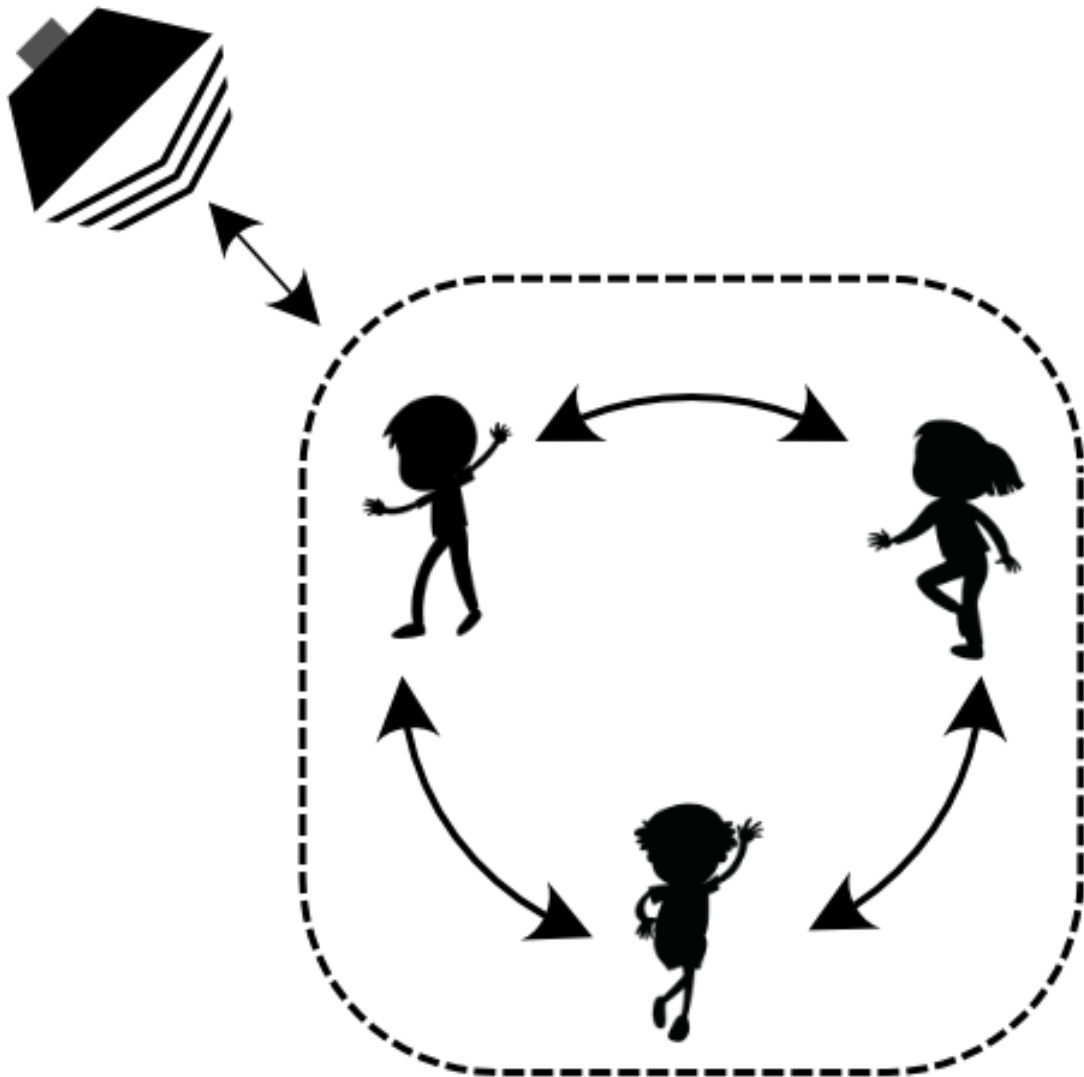


Figure 3: Child <-> Child <-> Installation

Child <-> Child <-> Installation The installation focuses primarily on achieving an interesting but also pleasing musical experience from all actions happening within. So

the interaction between the children's interactions with each other and the installation is always at the centre of all sound events. Some parameters like the overall tempo of the music can only be influenced optimally when all users interact with each other.

Ideally we would see the children act together and deliberately try different approaches to interact with each other to influence the soundscapes and observe how their interactions relate to the overall sonic experience; to have one child for example to suggest to the group to go faster and find out if that does anything, or to all, at the same time, throw their balls at each other and explore how that will be sonified.

Implementation

Structural Implementation

The installation is conceived as a virtual room of sound with a playground for the children in the centre. Other than for the speakers, the installation is not visible to the user's eye; all technical aspects will be hidden. This is to further mystify the perceived connection between action and sonic reaction.

A speaker system will surround the playground where the children will play and explore within four speakers (one in each corner). This creates a virtual space since the participants will be immersed in sound from all sides, further hiding the installation since the localisation will not be possible, while also making the participants feel like being part of it for being in the middle and inside of it.

Sonic Balls can be either installed on an outside or inside location. It will work in both situations. The implementation provided covers the sound system and all other technical aspects. Decorations and a kid-friendly environment will be provided by the location

and event organizer but can additionally be planned by us. It is recommended to put it within a *room* of any kind, so that it is possible to put all the technical devices and aspects around and behind the visible limits of the installation.

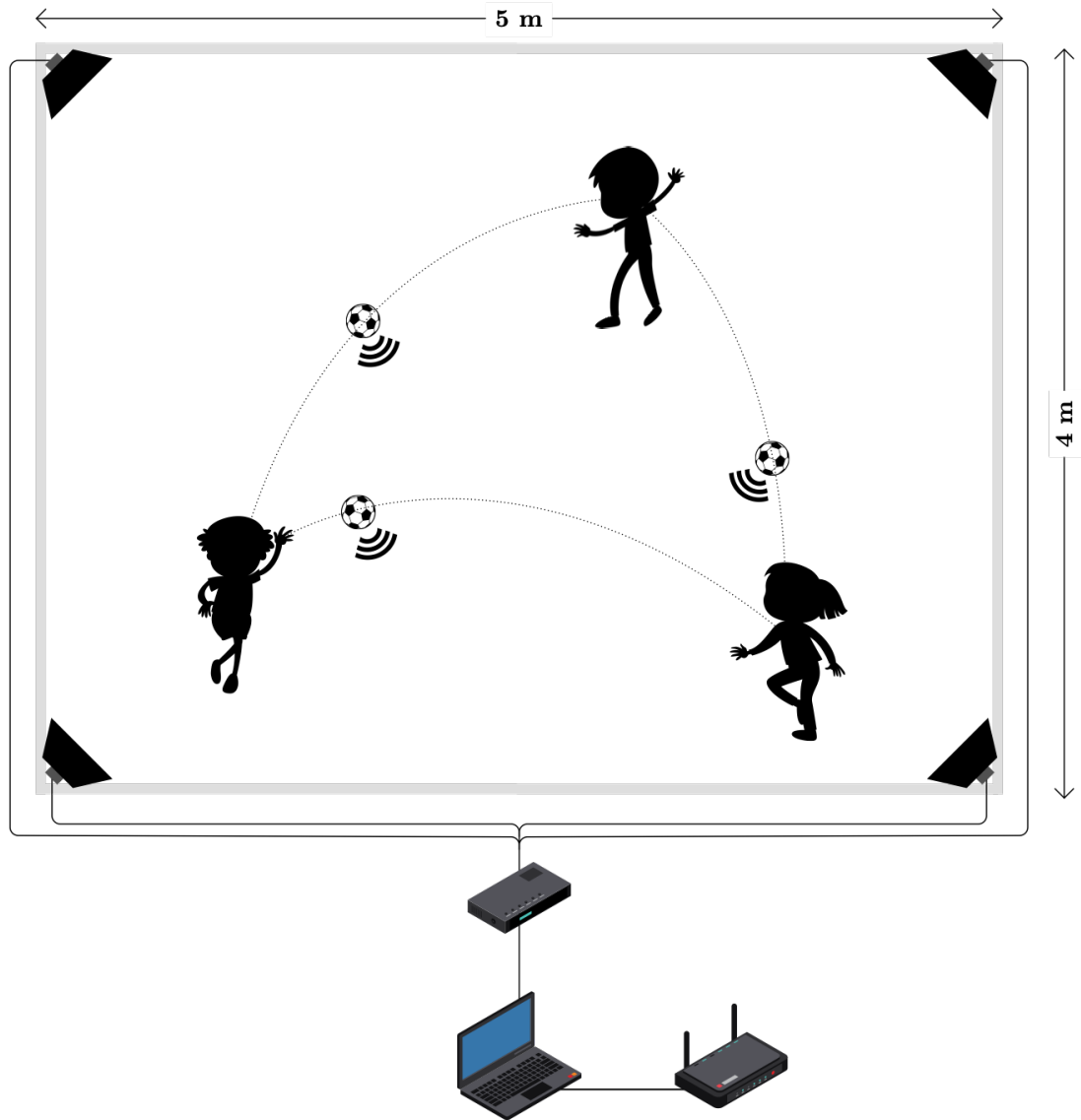


Figure 4: Overview of the layout and interaction within the installation

Technical Implementation (Overview)

From a technical perspective, the installation comprises of three steps that the information passes through before it will be audible to the users.

The source data is generated by the three balls handed out to the children. Each ball is filled with a custom Arduino-based device (<https://www.arduino.cc/>) with a high-quality accelerometer that tracks the orientation and acceleration of the ball. Furthermore the Arduino processes some of the raw data on device. The balls then send the prepared data over OSC (<https://opensoundcontrol.stanford.edu/>) and a local WiFi network to the main computer in the back. (Arduino, 2021; Control, 2002)

The central computer then receives all the data and processes them through Max for Live (<https://www.ableton.com/en/live/max-for-live/>), as well as calculates some parameters that only can be observed across all balls at the same time (e.g. average velocity). These data points then are directly mapped to parameters within the Ableton Live set. (Ableton, 2021)

Balls Each ball is equipped with an Espressif ESP32 chip connected to a Bosch BNO055 9-axis IMU sensor. The BNO055 was chosen since it combines multiple sensors into one while also providing more accurate data than other chips by constantly calibrating itself internally. This also means that when installing Sonic Balls, there is no need for on-site sensor calibrations. The balls are, when charged, always ready to go.

The chip runs code that constantly gathers data from its sensors and prepares them to be sent via WiFi and OSC. Besides raw data, the Ableton Live set needs processed data points, including velocity (approximated from acceleration), collision detection (based on sudden changes in acceleration; see Appendix), and a formant scaler (derived from orientation data).

These data points are for efficiency reasons already calculated on the Arduino instead of in the central Max/MSP session. Embedded C++ code performs better in real-time

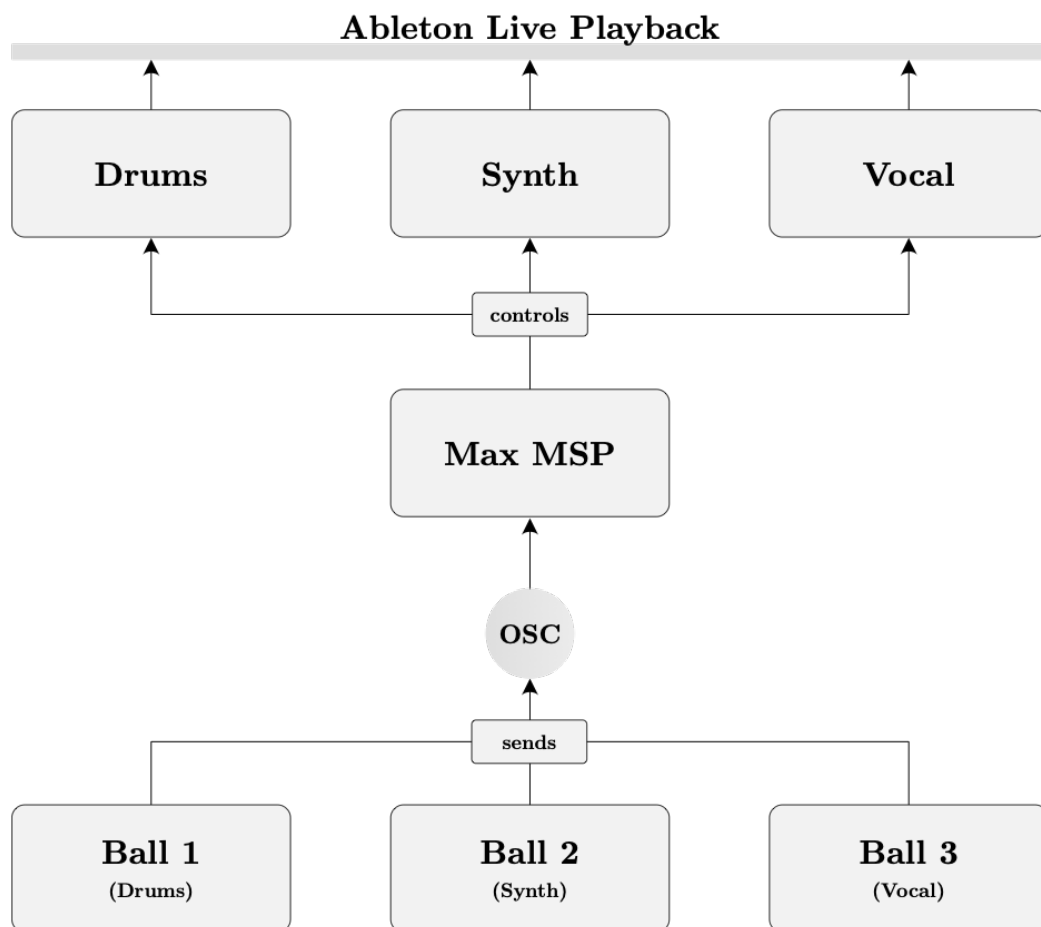


Figure 5: Diagram that shows the flow of data from ball to Ableton Live

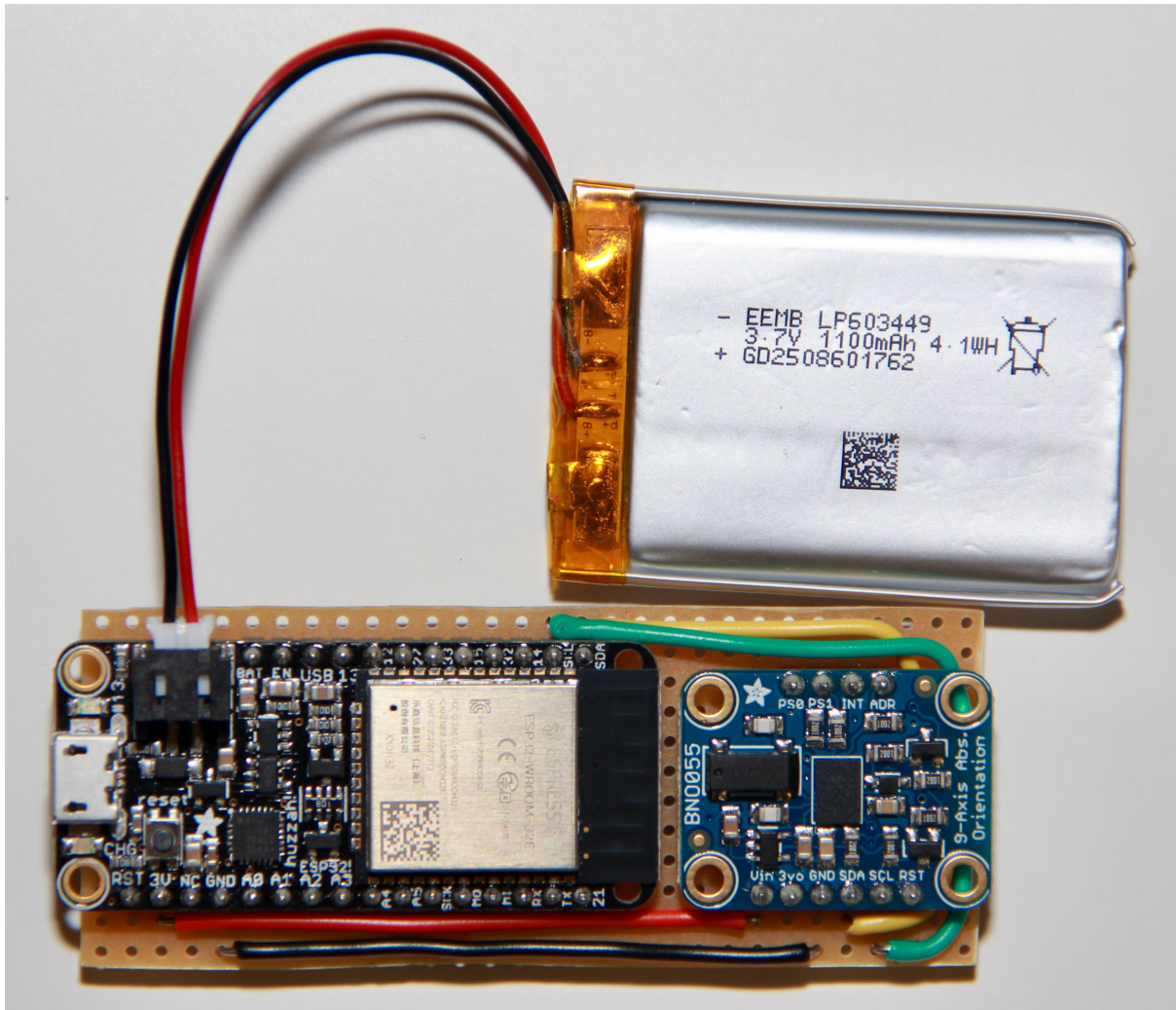


Figure 6: Arduino-based sensor unit

applications and allows for more customization. The balls also send warnings about their battery state and whether they are running too hot.

The WiFi connection data needs to be embedded in source code since there is no practical way of handing them over during runtime execution. This, in turn, means that the boards need to be recompiled when the WiFi settings change. For this reason, the installation includes a pre-configured WiFi router to which every participating device will connect. Through this, we can ensure that the login and connection data remains consistent.

Computer The computer is the central brain of the installation. It hosts the local WiFi network, receives data from the balls, processes the data, controls the sound design, and plays it back over the speaker system. It calculates the average velocity of all balls to interconnect their movement into the playback speed of the music. All the parameters are received through OSC within Max for Live and then passed on to individual controls on the sounds and effects themselves.

The computer is connected to an audio interface that then directly is connected to the four active speakers surrounding the installation. This eliminates the need for a more complex playback system setup with separate mixer and output stage.

Parameter Map (Arduino -> Ableton Live)

Parameter	Drums	Synths	Vocal
Collision Detection	Pattern Change	Pattern Change	Pattern Change
Formant Scaler	//	//	Formant
Orientation Vector	Reverb	Filter	//
Acceleration	Distortion	//	Pitch

Parameter	Drums	Synths	Vocal
Acceleration Vector	LFO Rates	LFO Rates	//
Average Velocity	Tempo	Tempo	Tempo

Musical Implementation

The music is heavily inspired by the production style of modern post-club music, drawing influences both from UK bass music and D’n’B, neo-trance and modern dembow. It is written in a way that the rhythmical content scales from lower to higher tempos, since the velocity changes of the game influence the tempo of the music. This leads to the music sounding like a chill dancehall jam, while also increasing its intensity up to a frenetic D’n’B track.

It consists of three major elements. The **drums** are responsible for the rhythmic feel and change described above, controlling the general sense of speed of the installation, motivating the children to go faster and maybe even become motivated to dance with the balls and each other. The music will only allow for dancing when the participants are dancing, reverting the role of the initiator, which usually is the music, to the listener. So the music will play to the dancing, as opposed to the listeners dancing to the music.

The **synthesizers** play pads and simple melodic patterns that either embrace the chill vibe of slow rhythms but also empower and fit well to the upbeat dancing moments of the sonic experience.

The **vocal** adds yet another level of audible dimension to the landscape. It primarily reacts to speed and orientation of the balls. Since humans are keen to notice changes

to the voice, the synthetic changes to the vocal based on motion data are especially interesting and provoke exploration.

Current State & Next Steps

We already developed a working prototype of one ball with its devices and the Ableton Live set with the sound design during the last semester.

Louis Koehler is responsible for hardware development due to his extensive experience in Arduino-based development and embedded system engineering. He built the internal devices for all the balls and will be responsible for hiding them within the balls.

I, Fabian S. Klinke, am the software engineer of the team and, as such responsible for the development of all the algorithms needed which process values and read the raw data from the sensors on the Arduino, as well as the implementation of the connection between the balls and the sound design. Most of these are completed as a prototype as of now.

Michele Sinatti is the sound designer of the team and through that also responsible for the artistic vision behind the entire installation. With his background as an electronic avant-garde composer, he is able to construct a sonic landscape that will be interesting but also pleasant to listen to.

Currently missing is the full installation with all balls and playback speaker system since the goal was to develop a working prototype and solve the technical difficulties of such an installation. This would be the next step to realize.

Appendix

Figures



Figure 7: Image of a Theremin (source: <https://www.tonecontrol.eu/moog-etherwave-theremin-standard-ash>) (Tonecontrol, 2021)

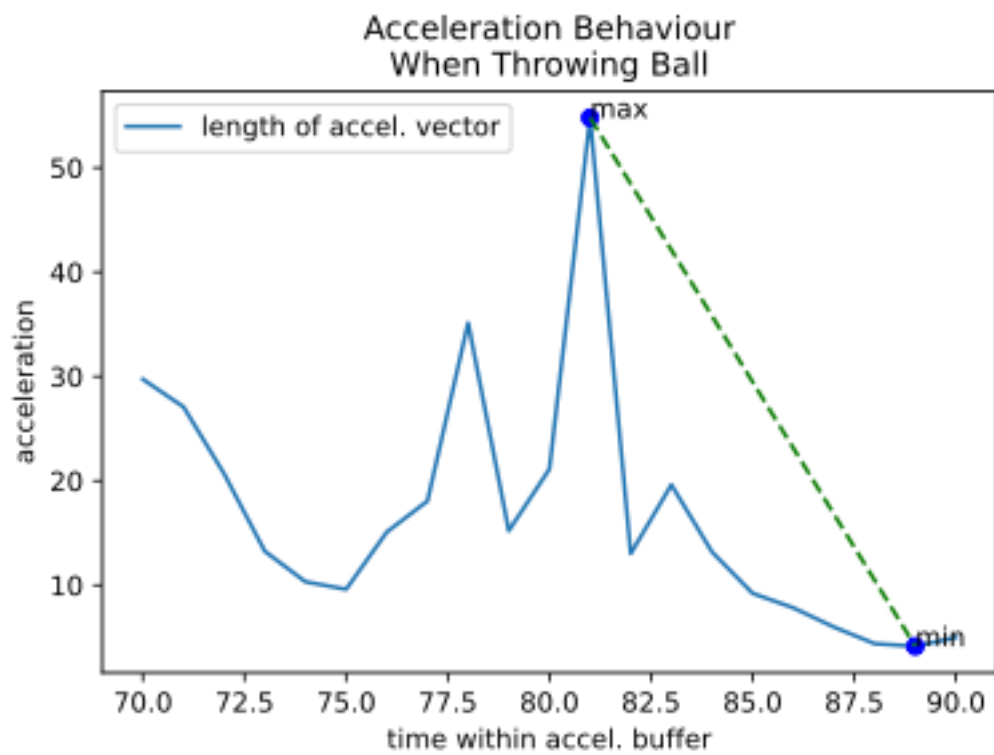


Figure 8: Graph showing an example of the acceleration buffer with the green line marking the detected sudden change in acceleration and hence a collision.

References

Ableton. (2021). *Max for live*. <https://www.ableton.com/en/live/max-for-live/>

Arduino. (2021). *Arduino*. <https://www.arduino.cc/>

Control, O. S. (2002). *OSC: Open sound control*. <https://opensoundcontrol.stanford.edu/>

Tonecontrol. (2021). *Moog etherwave theremin standard ash*. <https://www.tonecontrol.eu/moog-etherwave-theremin-standard-ash>