Lost In Space: Indoor Localization for Virtual Environment Exploration

Chase Mitchusson
Experimental Music & Digital Media
Louisiana State University
cmitc79@lsu.edu

Dr. Jesse Allison
Experimental Music & Digital Media
Louisiana State University
jtallison@lsu.edu

ABSTRACT

Lost In Space is a project utilizing Bluetooth Low Energy beacons in tandem with mobile devices and 3D panning on the web to overlay virtual sound arrangements onto a physical location in which users can listen through their phones and tablets. The virtual environments are distributed through a website and are populated with virtual sounds and speaker locations. Users activate the Bluetooth on their mobile devices to scan for beacons. User location navigates the corresponding location in virtual environment. Moving around the virtual environment probes the soundscape. The project also touches on issues of unreliability of Bluetooth tracking indoors, the state of BLE-based project development, and the need for Web Bluetooth development to enable these types of projects.

1. INTRODUCTION

Bluetooth Low Energy (BLE) beacons are placed around a room or large space. The user opens an app on their mobile device that scans for these specific beacons, see Figure 1. Based on data such as Received Signal Strength Indication (RSSI), an approximate distance from the mobile device to the beacon is calculated. This data is sent to the *Lost In Space* website's JavaScript. When the website has the proximity to at least three beacons, trilateration is used to find the user's location relative to the beacons. The user's location is then mapped to the virtual sound environment built in Tone.js. The virtual sound environment, see Figure 2, is a series of virtual speaker nodes scattered about a virtual environment analogous to the physical space in which the app is used. When the user gets near virtual speaker nodes, the nodes become audible through the mobile device's speaker, see Figure 3.

The Web Audio API allows for complex processing and routing of audio on the web [1]. Tone.js is an interactive music framework built on the Web Audio API [2]. Artists and developers can take advantage of these interfaces to create applications and pieces for mobile devices [3]. Mobile performance pieces built with the Web Audio API can attempt at localizing each participating mobile device [4]. *Lost In Space* uses indoor localization techniques with Bluetooth to allow users to explore a virtual environment analogous to their physical space.

Bluetooth Low Energy beacons transmit signals that are scanned by mobile devices. With enough beacon data, a user's indoor location is tracked. By tracking physical location, a virtual analogue is positioned on the web. Depending on the user's virtual location, their mobile device outputs audio from nearby Panner3D nodes, which determine the volume and location of a sound [5].

2. RELATED WORK

There are prior location-based projects for GPS-enabled devices by which *Lost In Space* is influenced. Allison's and Dell's *AuRal* is an augmented reality environmental audio system where participants are grouped into ensembles based on their geolocation [6]. Participants share a sonic environment similar to the shared virtual environment of *Lost In Space*. *AuRal* users can affect the sonic environment with musical input, which is unlike *Lost In Space*, where users are encouraged only to explore the soundscape.

The geolocation adaptive music player developed by Alfonso Perez-Carillo et al. varies playback of music tracks based on a users geographic location. The music player is linked to a geographic path users follow which determines song length and transitions between tracks [3]. This approach is close to the Tone.Panner3D parameters set in *Lost In Space*. Users experience the audio roll off based on distance from the Tone.Panner3D which imitates a transition to the next closest Tone.Panner3D.

3. USES

Lost In Space has several artistic applications including soundwalks that track geolocation, distributed mobile performances, and spatialization in challenging environments. The original concept for *Lost In Space* was to emulate SpaceMaps in a browser [7]. SpaceMaps use tri-sets of nodes to pan across speakers mapped to the node tri-set. SpaceMap experimentation revealed that a simpler implementation would be adequate for the project. A single point-source of audio is easier to identify than balancing three audio sources. Using static audio sources scattered in a virtual environment, users control what they hear from their device. The option to explore the space they are in to discover new sounds belongs to them.

As a distributed performance piece, twenty or so users visit the *Lost In Space* website and walk or run around their physical space while listening to the virtual environment.

Web Audio Conference WAC-2018, September 19–21, 2018, Berlin, Germany.

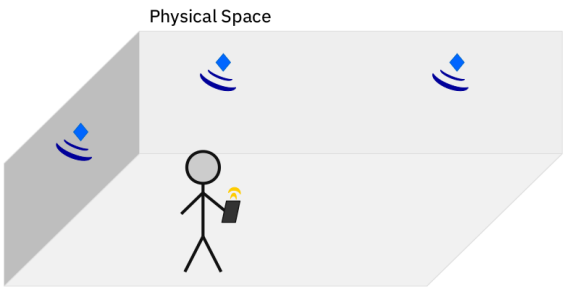

Figure 1: A user in a physical space.

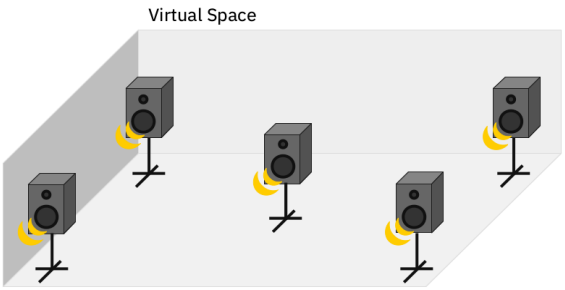

Figure 2: Panner3D nodes in a virtual space.

This creates an instance of a musical piece localized in the physical space users occupy. Every performance will be different. Performer number, device types, and physical space can all change for each performance.

The version of *Lost In Space* that is a soundwalk is a more solitary one. A user in a physical environment will walk about with headphones and hear field recordings from the same location to create historical perspective. This application is great for museums and historical districts of cities. Since it is proximity based, users can walk at their leisure to enjoy what they hear.

Lost In Space is an attempt to make massive multi-channel concert halls portable. A multi-channel piece is programmed across a large number of virtual speakers and played back in challenging locations, like forests or lakes. The audience coordinates to arrange their devices into a speaker configuration appropriate for the piece, and listen to it spatialized in an extreme location. This configuration still lacks the clarity of great concert hall speakers, but it is a great place to start experimenting.

4. METHODS

BLE Beacons/Bluetooth 4.0 are used for indoor localization. *Lost In Space*'s website is built with HTML, CSS, and JavaScript to comply with web standards. The JavaScript portion relies on the Web Audio API and Tone.js to create virtual speaker nodes. Android Studio is used to build a mobile application to access mobile devices' Bluetooth in combination with a WebView to share Bluetooth data to the JavaScript in the website [8, 9].

4.1 Bluetooth

Bluetooth beacons have the ability to track proximity indoors. They are portable and will soon work on the web in a browser [10]. At least three Bluetooth Low Energy (BLE) beacons are distributed among a space. With three beacons, trilateration equations are used to determine a mobile device's position relative to the beacons [11]. Location data for these beacons and the mobile device are fed to JavaScript objects for positioning in the virtual environment. Position data is stored using right-hand Cartesian coordinates to match the orientation of Tone.js Panner3D objects.

4.2 Web

Tone.js has a simple and effective interface for handling web audio. Specifically, the Tone.Panner3D object is ideal for *Lost In Space*'s need to audibly signal location in a virtual environment. Tone.Player objects are created and given audio files to playback. The Players are connected to Tone.Panner3D objects which are positioned in a virtual environment with right-hand Cartesian coordinates. The Tone.Panner3D nodes are connected to the master output, a user's mobile device's speaker(s). There are five Tone.Players with five different audio files. The Tone.Players are sent to Tone.Panner3D nodes, which are in a row of two speakers, a row of one speaker, then a row of two speakers. The layout consists of two sets of stereo pairs, separated by twenty units each in the depth axis, and a center channel ten units away from the other speakers.

4.3 Android Application

Due to lack of web Bluetooth support, an Android application is bridging the mobile Bluetooth scanning to the browser. Android applications have access to mobile device's Bluetooth hardware, and several libraries exist for handling the Bluetooth data. The AltBeacon Android Bluetooth library offers the option to scan constantly while relaying the mobile devices' approximate distance from BLE beacons [12]. The Android application is simply a WebView with AltBeacon running in the background. The WebView visits the *Lost In Space* website and sends beacon distance data to the JavaScript in the *Lost In Space* website. In this fashion, as the Web Bluetooth API is implemented, the Android application wrapper can be jettisoned for a pure web browser experience.

5. DISCUSSION

Fine tuning the Web Audio portion of the project is necessary to adapt to each physical space in which the project is deployed. Mono output is used because tracking orientation of users for output based on head-related transfer functions is too complicated where devices may only have one speaker. Tone.Panner3D objects must be calibrated to get drop off levels correct when users leave a panner's area. Since Tone.js units are not measured to a specific unit type, e.g. meters, feet, inches, calibration is required when determining how the size of a physical space will map to the virtual environment. Correcting these measurements is simple in Tone.js, and the orientation of the virtual environment can be updated into any configuration desired.

Bluetooth is not incredibly reliable when judging distances by signal strength, as interference from other devices, people, or walls can cause fluctuations. New options to perform indoor localization are being worked on that rely on fingerprinting and polynomial regression models instead of signal strength that could improve accuracy of tracking users [4].

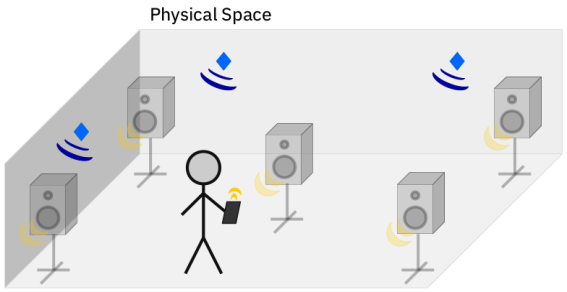

Figure 3: Combined virtual and physical spaces.

Bluetooth is, however, more convenient than indoor tracking using Wi-Fi. Wi-Fi tracking requires several routers nearby and works well in larger spaces, but is not as portable as BLE beacons [13]. Beacons travel better than Wi-Fi routers because they are small and battery operated. They therefore work in more challenging locations where Wi-Fi may not be present, like rural areas [11].

The Android application wrapper works well enough as a proof of concept, but will be removed as soon as Web Bluetooth is reliable. By expanding outside of the browser into application development, *Lost In Space* becomes an exclusive product for Android users. A separate development system is required to include iOS users, and even still requires users to download an ancillary application. Exclusivity is not ideal for exposing users to new forms of digital art.

6. CONCLUSIONS

Bluetooth based indoor localization methods are adequate for tracking users and are becoming sophisticated. BLE beacons are great for traveling, room-scale tracking projects. Mobile device speakers are adequate for indoor distributed performances, but lacking in large-scale outdoor performances. Android applications are versatile, but there is a need for web Bluetooth.

Ideally, this project will work entirely in a browser with no need for an Android application to access mobile devices' Bluetooth. Unfortunately, Web Bluetooth API is highly experimental and unstable currently. It may be in its infancy now, but Web Bluetooth will replace the Android application architecture once the Web Bluetooth API becomes stable and more features are incorporated. Placing the project entirely in the browser eliminates the need for users to download external applications, instead allowing them to use the browser with which they are already familiar. Without mobile apps, development is simpler and not based on Android or iOS frameworks. This approach attempts to make *Lost In Space* an inclusive experience, which is important for publicizing Web Audio development.

7. REFERENCES

[1] Web audio spatialization basics - web apis | mdn. https://developer.mozilla.org/en-US/docs/Web/API/Web_Audio_API/Web_audio_spatialization_basics.

[2] Tone.js. https://tonejs.github.io/.

[3] Alfonso Perez-Carrillo, Florian Thalmann, György Fazekas, and Mark Sandler. Geolocation adaptive music player. 2016.

[4] Yuan Zhuang, Jun Yang, You Li, Longning Qi, and Naser El-Sheimy. Smartphone-based indoor localization with bluetooth low energy beacons. *Sensors*, 16(5):596, 2016.

[5] Tone.panner3d. https://tonejs.github.io/docs/r12/Panner3D.

[6] Jesse Allison and Christian Dell. AuRal: A mobile interactive system for geo-locative audio synthesis. page 3.

[7] Steve Ellison. SpaceMap: 20 years of audio origami. page 8.

[8] Meet android studio. https://developer.android.com/studio/intro/.

[9] Webview. https://developer.android.com/reference/android/webkit/WebView.

[10] Web Bluetooth Community Group, Mar 2018.

[11] Faheem Zafari, Athanasios Gkelias, and Kin Leung. A survey of indoor localization systems and technologies. *arXiv:1709.01015 [cs]*, Sep 2017. arXiv: 1709.01015.

[12] AltBeacon - the open proximity beacon.

[13] Wi-fi vs bluetooth low energy (ibeacon) technology | beaconstac. https://blog.beaconstac.com/2014/11/wi-fi-vs-bluetooth-low-energy-ibeacon-technology/. (Accessed on 03/22/2018).