Final Project Report

Ahhyun Moon
am12180
am12180@nyu.edu

Abstract

Augmented reality blends the real world and digitally immersive world in one place, providing new pathways for user interaction and experience. In this project, we experiment an augmented reality music interface, which is an extension of analog LP record playing experience to augmented reality space.

1. Introduction

Rapidly evolving technology has vastly changed our way of consuming music. However, despite the convenience of one-tap streaming of favorite music from widely used services such as Apple Music and Spotify, there remains the demand for an analog way of playing music with LP records as it provides consumers with a tangible experience along with a feel of ownership of records. Being an old technology dating back to 1948, LP records and players have limited interactivity and functionality. Therefore, the objective of the project is to enhance and extend the analog music experience with the help of augmented reality technology.

2. Results and Demonstration

Augmented Reality Record Player is built as a mobile application available on iOS. There are two perspectives to reach the project objective: 1. Making improvements to aid the limitations in analog experience to match the current standard of music player and 2. Further extending the analog experience with AR to provide a dynamic music experience.

2.1. Track Information Display

First is improving the analog experience by showing currently playing track information and album art above the LP record. When application detects the record player in the scene, 2D box will be first instantiated above the record player (Figure 1-1). User can tap on the box to search currently playing song. It will listen to the song from microphone input, and will show the search result. If the song search is successful, it will show track information including album image, song title, and artist name (Figure 1-2).

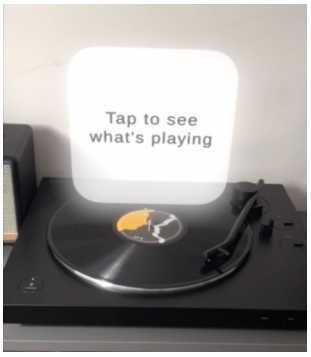

Fig. 1-1 Search UI

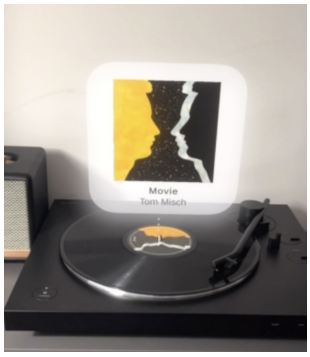

Fig. 1-2 Search Result

Figure 1. Track Information Display

2.2. 3D Content Display

Second is extending the current music experience by adding 3D visual components. There are two features in this category: 1. Showing sound spectrum bars matching the current song being played. (Figure 2-1). 2. Showing an animated character on the record player, as if the artist is performing live on the spinning miniature stage .

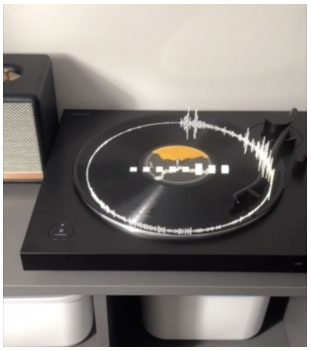

Fig. 2-1 Sound Spectrum Band

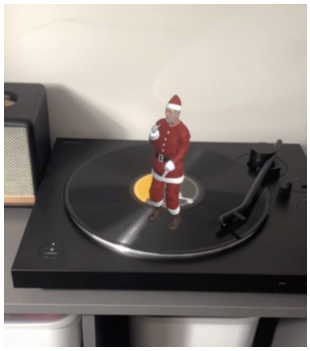

Fig. 2-2 Animated Character

Figure 2. 3D Contents Display

3. Implementation

3.1. Record Player Detection

The application detects the record player to instantiate prefabs in the scene. This is achieved through AR Tracked Object Manager in the AR Foundation framework in Unity. To use the trackable manager, we first need to save our target object in a reference library. For this implementation, the ARKit object scanning application is utilized to record object features as ar object file.

3.2. Track Information Display

When a user presses the song search button, the application records the microphone input for 5 seconds and saves it as a wav file to send the recorded input to the music recognition API. In this project, AudD API is used to fetch song title, artist name, and link to album art. When the result of POST API request is successfully received, it will fetch the album image using the link received, and show song-related information in the search box as a result.

3.3. 3D Content Display

When a user presses the "Change Overlay" button, the application switches to different features such as Sound Spectrum Band Display and Animated Character Display. In Sound Spectrum Band Display, the application listens to microphone input and converts the audio input to spectrum data every frame. The spectrum data is divided into 512 frequency bands for scaling the height of 512 spectrum bars listed in a circle around the vinyl record. The positioning of spectrum bars in a circle is based on the center of the detected record player. X and Y position is calculated based on the radius (distance from the record player center) and angle for each bar. There are also 8 aggregated audio frequency spectrum bars in the center of the vinyl record which represent the 512 divided frequencies grouped into 8 bands, so that it show smoother visualization of the sound input. In Animated Character Display, the application utilizes an animated character prefab and displays it at the center of the record player.

4. Discussion

4.1. Object Detection

The current version of the application uses prerecorded target information to detect the record player in the camera scene. Therefore, the application's use case is limited to a preassigned record player. For general uses beyond this limitation, detecting the vinyl record in a camera scene is more desirable than detecting a preassigned record player. A possible method can be the implementation of a custom machine learning model (a model trained with Faster R-CNN for instance) for detecting any vinyl record. Having not many features both in shape and color, vinyl records can be a challenging object to be recognized by a detection model, which will require a huge amount of images for training. For this challenge, combining real-world images and synthetic image data can be a solution and this process can be sped up using automatic labeled data generation provided in Unity Perception package. [4] Also, binding the model to AR space can be accomplished by the Unity Barracuda package, which is a lightweight cross-platform neural network inference library that can run neural networks on both the GPU and CPU. [2]

4.2. Interactive 3D Contents

Another possible improvement is for the interactive 3D prefabs used in the 3D Content Display feature. Audio spectrum bars can be improved for smoother visuals by testing with different scale values when converting audio frequency to 3D cube's height as well as by adding buffer value when grouping small frequency parts into eight center bars. Furthermore, to make animated characters more interactive with music, there are two possible approaches: 1. Using keywords such as genre or mood in song search results and 2. Detecting the tempo or BPM (beats per minute) to determine the song speed. Using either keyword or detected speed, we can control character animation movement or speed so that it can create a perception of the character performing or interacting with the song. For BPM detection, there are existing Unity assets or projects that can be referred to such as RhythmTool that can analyze beats or pitch of a song [3], and Unity-Beat-Detection project which touches on beat detection and audio spectrum analysis. [1]

References

[1] Gihub User: allanpichardo. Github Repository: Unity-Beat-Detection. https://github.com/allanpichardo/Unity-Beat-Detection, 2014. 2

[2] Unity Documentation. Introduction to Barracuda. https://docs.unity3d.com/Packages/com.unity.barracuda@3.0/manual/index.html, 2019. 2

[3] Unity User: HelloMeow. Unity Asset: RhythmTool - Music Analysis for Unity . https://forum.unity.com/threads/rhythmtool-music-analysis-for-unity.270866, 2014. 2

[4] You-Cyuan Jhang, Adam Palmar, Bowen Li, Saurav Dhakad, Sanjay Kumar Vishwakarma, Jonathan Hogins, Adam Crespi, Chris Kerr, sharmila chockalingam, Alex Romero, Cesar abd Thaman, and Sujoy Ganguly. Training a performant object detection ML model on synthetic data using Unity computer vision tools. https://blog.unity.com/engine-platform/training-a-performant-object-detection-ml-model-on-synthetic-data-using-unity, 2020. 2