Playing a Virtual Piano with Dynamics

Daniel Salz
Stanford University
dasalz@stanford.edu

Farhan Azam
Stanford University
fazam@stanford.edu

Abstract

This project sought to create an intuitive piano playing experience through VR that allows users a cheaper and mobile alternative to that of real pianos. We used Leap Motion controls for accurate finger tracking and the Unity engine for creating the piano playing scene. We were able to successfully implement finger tracking to press the keys of the piano. User feedback depended on how advanced their piano skills. Novice users actions were easily recreated whereas advanced users could not get the feedback they wanted. The demo currently works better as a prop as opposed to a real piano, though future improvements could lead to a more authentic playing experience.

1. Introduction and Motivation

Virtual Reality (VR) opens the opportunity to interact with real world devices that would otherwise be difficult to obtain or realize. With this in mind, we wish to create an interactive virtual piano that allows users with a Leap Motion device and VR headset to play as though they were in front of a real piano. This project is significant because it would not only allow a new population of people to play the piano, but also make it easier for current piano players to practice and play while on the go. Many people want to learn how to play instruments, but the financial burden of buying and maintaining an instrument can be a significant obstacle. For example, even an already purchased piano requires tuning that costs hundreds of dollars. Mobility is also a significant problem; pianos are large and extremely cumbersome. Instruments like these are difficult to transport without other machinery. Most people can't make space for something like a piano, especially on the move. A virtual piano would allow users to avoid these cost, mobility, and space issues. With the widespread use of VR, we hope to create a simple VR experience in which a growing user base can play a piano almost anywhere. Our work would allow an entire new population access to pay the piano without the need to leave their homes, pay for lessons or rent rooms. Further work could include incorporating a virtual assistant that dictates the user to continue improving, removing the necessity of a piano teacher.

We hope to create a stimulating experience that captures the same realities of sitting in front of a piano and playing. The virtual piano will respond to user's fingers accurately hitting the keys by using hand and finer motions tracked by the Leap Motion device. The Leap Motion device is a hardware sensor that takes these inputs and requires no physical contact. None of the current iterations of virtual pianos accurately track the velocity of finger tracking to dynamic sound of the keys, something extremely important for all songs of all genres. Doing this requires fine tuning of the sounds of notes dependent on the velocity at which users move their fingers to press keys. This would be the largest contribution from this project as current VR pianos do exist, but they miss on this vital aspect of playing the instrument. The end product should be a fully interactive piano that mimics real life without the tactile response aspect. Future work could be down to incorporate tactile feedback which would complete the entire experience of playing a piano, but now in VR.

In summary our contributions are as follows:

1. Incorporate a a piano into a VR unity environment that allows for tracking of finger movements to play.

2. Using velocity of finger movements to introduce dynamics of the notes being pressed.

2. Related Work

Our virtual piano will build upon and improve on work with hand tracking, virtual pianos, and other virtual instruments.

2.1. Hand and Finger Tracking

The most important aspect of the project is precise hand and finger tracking while also removing the need of intrusive hardware. We could have used haptic gloves, but this wouldn't fit all users. The Leap Motion is one-size-fits-all and provides an accurate interface that is only inaccurate up to .5mm [6]. This provides confidence that the Leap Motion will be sufficient to get accurate measurements for playing virtual instruments. A rough idea of the Leap Motion device and its usage is shown in Figure 1. How to utilize these measurements is also essential. A previous EE267 project integrated the leap motion for several movements such as hand translation, rotation, key tapping, and dynamic movement of the fingers [3]. These are all essential movements for playing the piano and will be heavily utilized in our own implementation of the leap motion device.

Figure 1: The Leap Motion is a small USB periphery device that captures hand motion using IR lights and LED's.

2.2. Current VR Musical Instrument Implementations

Currently most instrument implementations in VR are simple string or tapping based instruments. These are beneficial because they allow a tipping off point for the beginning of musical experiences in VR, however, they do not contain many of the complexities that are involved with transferring a real musical instrument to VR. These projects though do give a framework on how to approach bringing music to VR [2, 4, 7]

2.3. Current implementations of VR Piano

The best implementation of a viewable VR piano project currently is from the Beijing Institute of Technology. They also used the Leap Motion for finger tracking and successfully implemented the piano playing features. Their main issues with the final product were: no force feedback making it difficult for users to locate their hands when playing quickly and no real collision making certain actions difficult to interpret in the game engine. Our implementation hopes to do the same as this project, but also add dynamics to the keys [5].

3. Implementation

3.1. Unity and Piano Integration

While we could create our own interactable piano object for Unity, this wouldn't be practical or time-effective. Therefore, we were limited by the assets offered in the Unity Asset Store. Fortunately, we found a musical keyboard asset complete with animations and musical notes corresponding to the individual piano keys. Originally, the animations and notes were triggered by pointer clicks, so we had to modify the scripts to trigger the notes upon collision with other objects.

3.2. Unity and Leap Motion Integration

The integration of Leap Motion and Unity was difficult at first because the Leap Motion is only compatible with certain VR headsets like the Oculus Rift and HTC Vive. Also, the Leap Motion Unity Asset only works for Windows, so we had to develop on the EE 267 Lab Computers. Originally, we were planning to use the Oculus Go to create a more seamless VR experience, but the Oculus Go doesn't support Leap Motion integration. Therefore, we have to fall-back on the Custom View Master Head Mounted Device (HMD) built for EE 267. More details and specifications of this can be found on the EE 267 website. The experience was run on a Windows PC in Unity, and the LCD display within the HMD was connected as an external display. Using concepts that we learned in EE 267 and the scripts provided by the TA's, we could use stereo rendering as picture below.

Figure 2: There are two images rendered, one for each eye in the View Master. The discrepancies between these two images give the illusion of depth.

3.3. Finger Tracking and Collisions

There are a total of 4 hand models in each scene. There is a rendered hand model that you can see for each hand, and a physical interaction based model that is hidden from the user. The physical based model contains collider shapes, outlined in green in the image below. These collider shapes are used to demarcate when objects collide with one another.

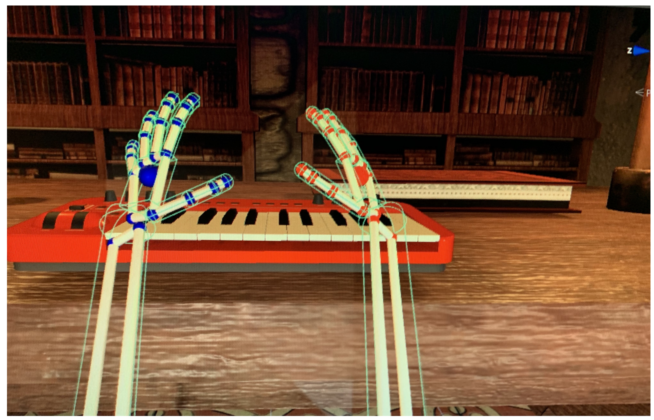

Figure 3: Each of the individual bones of the fingers have colliders objects around them, as well as the wrists and palms.

Once the Leap Motion hand models were integrated with Unity, it was time to finally test the collision between the keyboard buttons and hand models. While the key pressed animations were correct, the sound dynamics were incorrect. Specifically, some of the notes didn't stop playing once collision between the piano key and hand ended. Our initial guess was that the Leap Motion hand models used a sweep-based continuous collision detection algorithm [8]. However, the Unity implementation of this algorithm doesn't account for the angular momentum of objects, and the large performance overhead associated with this algorithm may have caused it to miss some collision enter/exits. As a solution, we called a function that performed sound fading shortly after the note was played, simulating correct sound dynamics.

3.4. Sound Fade Dynamics

To create realistic sounding piano notes, sound fade must be incorporated after the key is no longer pressed. There are different curves used for sound fade, including linear, logarithmic, exponential and s-curves. In this experience, a straightforward linear sound fade is used. The algorithm for the linear sound fade is presented below:

```
IEnumerator SoundFade(AudioSource source) //sound fade after the button gets unpressed
{
    float progress = 0;
    while (progress < 1)
    {
        progress += 0.75f * Time.deltaTime;
        if (source != null)
            source.volume = volume * 1 - progress;
        yield return null;
    }
    Destroy(source);
    yield return null;
}
```

Figure 4: In this algorithm, the volume of the audio source is linearly decreased over time. Time.deltaTime records the time elapsed between the previous frame and the current frame.

3.5. Visual Feedback

An extra feature added was visual color feedback. When the keys are pressed by the user, they light up as a visual feedback. Our purpose for doing this is that both visual, haptic, and auditory feedback are a important components of commercial VR systems [9]. While we couldn't implement haptic feedback, we thought that colored keys could be a feasible and worthwhile additional feature. You can observe this implementation below.

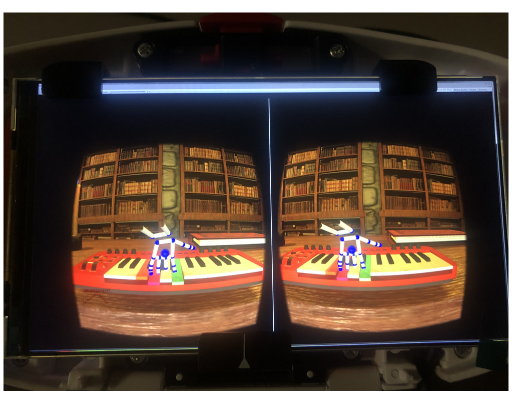

Figure 5: The individual piano keys light up with different colors when the user presses them.

Additionally, this visual component provides a nice segue into for future work in creating tutorials by lighting up the keys that users should press when playing a new song.

4. Experiments

4.1. User Feedback

Our experiments were limited to our group and the few people who tried our experience during the EE 267 Final Project Demo Session. Most, if not all, users were novice piano players. Many people were pleased with our demo. However, there were some hit-or-miss experiences depending on how users position themselves.

The Leap-Motion controls require users to be positioned accurately above the sensors. If users do not place themselves correctly, the demo feels strange due to the visual feedback of finger movement being different than their own. Further work could employ strategies in Unity to mask the finickiness of the Leap Motion interaction with the piano. Another option could be to add an interactive tutorial at the beginning of a session that demonstrates the best way to position their hands or even have some kind of object in the scene that helps users with the positioning of their hands while playing.

Novice users in general were happy with the demo to play a few keys with their fingers. This is where the demo excels since it can very easily track the independent fingers and the collision with keys works very well when only one finger is used to play. This was not the same though for advanced players. They reported normal hand and finger movements they usually employ when playing piano did not work as well as they would hope. The main issue it seemed is that it was difficult to perceive how close to the piano their hands were and when simply lowering a finger to play a key it would either not reach or they would put their hands too close causing several unwanted keys to play. Another issue they reported was the inability to rest their hands on the keys since the colliders immediately start playing the sound when a finger pressed on the keyboard. A solution to this in the future could be to incorporate the velocity of the finger as a threshold before the finger is pressed.

From this user feedback we evaluated that our demo as it stands now works more as a prop piano rather than an alternative to real piano playing. Novice users would enjoy it significantly more as the demo works fine with how they typically employ their fingers, for example playing the song Chopsticks works very well as opposed to anything that requires more than one finger on each hand.

4.2. Qualitative Evaluation

The robustness of the experience depended a lot on how users placed their forearms/wrist. If users placed their forearms/wrists relatively high above the Leap Motion, the Leap Motion was able to accurately tracked their hands and fingers. However, if their forearms/wrists right above the Leap Motion, there hands would be placed at awkward and unnatural positions in the VR experience. We believe this is reasonable, since when people are taught to play the piano, they are often instructed to keep their wrists raised. You can see both of these positions in Figure 6 below.

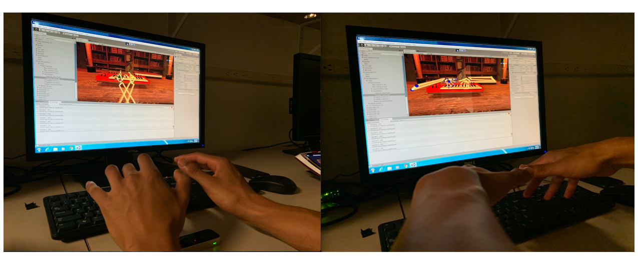

Figure 6: One the left, the user has the wrists closer to the Leap Motion device, resulting in an inaccurate depiction of the hand models. On the right, wrists are raised, so the Leap Motion is able to get a full view of both hands.

5. Conclusion

Our finished product resulted in a demonstrable prototype of a VR piano experience. Users may take some time to adjust to the hand placement at first, but after a few minutes of playing with it, they were capable of playing beginner songs with reasonable results. Advanced users though were on the opposite spectrum and did not enjoy the demo nearly as much. The lack of tactile feedback and inaccuracies in the Leap Motion to track the much faster motion of advanced players left them wanting more. Our motivation for this project was to provide an alternative to a real piano, as it stands this is not the case. Small improvements though in future work though could create a more realistic experience and provide users a mobile substitute for real pianos. What we learned through this implementation could be applied to creating other non-wind type instruments in VR. Creating a VR drums experience not to be too far off from the same methods employed for the piano and we expect it to be more realistic as playing the drums already requires the use of a controller like interface.

6. Future Work

In the short term, we could definitely make some modifications to our prototype to create a more robust experience. For starters, the View Master HMD requires the use of one's hand at all times. Therefore, the user isn't able to simultaneously play the piano with both hands and use the View Master rendering. To solve this issue we would need to create a head strap similar to other commercial VR headsets, allowing for a more immersive experience.

Additionally, we would also like to make the hand tracking more robust. To do that, we would need to provide the Leap Motion device with an unobstructed view of both of the user's hands. To do this, we could attach the Leap Motion device to the View Master HMD, as some other commercial VR headsets do. That way, the user can place their wrists/forearms as high or low as they please. This would require more calibration within Unity, but users would be happy to do this at the beginning of a session if it meant more natural movement from increased finger tracking accuracy.

To create a more realistic sound fade, we could consider implementing a logarithmic curve sound fade. This more closely mimics how human ears perceive sound. This would be a nice touch for audiophiles and musicians who wish to use this experience in a way that more accurately imitates real pianos [10].

Adding tutorials is a simple addition that could be easily implemented by utilizing the lights on keys we already have. The addition of tutorials could provide a cheap experience for users to learn as opposed to paying for real lessons or learning from videos which is not nearly as intuitive.

References

[1] Play Music with AR Technology! Retrieved April 26, 2019, from https://www.music-everywhere.co/ (n.d).

[2] Peterson, C. Video Series: Taking VR Guitar to a Whole New Depth. Retrieved from http://blog.leapmotion.com/video-series-taking-vr-guitar-whole-new-depth/ 2017.

[3] Lin Shao. Hand movement and gesture recognition using Leap Motion Controller. Retrieved from http://stanford.edu/class/ee267/Spring2016/report_lin.pdf (2016).

[4] Vestergaard, J. Building a virtual reality music instrument. Retrieved from https://medium.com/@jespervega/building-a-virtual-reality-music-instrument-d97bdf9fc5d4 2015.

[5] W. Qiao, R. Wei, S. Zhao, D. Huo and F. Li, A real-time virtual piano based on gesture capture data, 2017 12th International Conference on Computer Science and Education (ICCSE), Houston, TX, 2017, pp. 740-743. doi: 10.1109/ICCSE.2017.8085592, 2017.

[6] Guna, Joe & Jakus, Grega & Pogacnik, Matevz & Tomazic, Saso & Sodnik, J. An Analysis of the Precision and Reliability of the Leap Motion Sensor and Its Suitability for Static and Dynamic Tracking. Sensors (Basel, Switzerland). 14. 3702-20. 10.3390/s140203702, 2014.

[7] J. Fillwalk, ChromaChord: A virtual musical instrument, 2015 IEEE Symposium on 3D User Interfaces (3DUI), Arles, 2015, pp. 201-202, 2015. doi: 10.1109/3DUI.2015.7131770

[8] Larsen, E. Collision detection for real-time simulation. Case study, Reseach and Development Department, SCEA, 1999.

[9] Wendy J. Adams, Iona S. Kerrigan, Erich W. Graf, Efficient Visual Recalibration from Either Visual or Haptic Feedback: The Importance of Being Wrong Journal of Neuroscience 3 November 2010, 30 (44) 14745-14749; DOI: 10.1523/JNEUROSCI.2749-10.2010

[10] Langford, S. Digital Audio Editing. Burlington: Focal Press. pp. 47-57, 2014.