

TappyBeatsXR: Using a modern mixed reality approach to develop a virtual music creation environment for non-musicians.

Jarle Hjellvik Wallevik

Master's thesis in Software Engineering at

Department of Computer science, Electrical engineering and Mathematical sciences,
Western Norway University of Applied Sciences

Department of Informatics,
University of Bergen

September 2022



**Western Norway
University of
Applied Sciences**



Abstract

The recent upswing of interest in the domain of visual headset technology has brought with it many generations of impressive peripherals. The HoloLens 2 by Microsoft merges virtual reality and augmented reality to give you an experience called mixed reality. There are many historical instances of researchers and musicians working to combine music creation with technology to create experiences previously thought impossible. Participating in the creation of music is traditionally hard, requiring previous experience or knowledge. The motivation behind this thesis is to create an easy-to-use music creation interface for users of all skill levels, leveraging the interactivity and immersiveness of mixed reality to design a unique experience. This thesis also explores the technologies and engineering methods available for developing for the domain of mixed reality. The application developed for this thesis - TappyBeatsXR - is a prototype that uses the mixed reality headset HoloLens 2 to create a loop-based music creation workstation. The application was tested on 5 unbiased users, and it received positive feedback along with suggestions for user experience improvements.

Acknowledgements

I would like to thank the official HoloLens Slack server for input and feedback as well as debugging assistance both for the HoloLens and the Mixed Reality Toolkit.

I would also like to thank Karsten Specht for fruitful discussions about potential application areas for TappyBeatsXR.

My utmost gratitude also goes to the department of informatics at UiB visualization research group for lending their HoloLens 2 to the development of TappyBeatsXR.

Finally, my sincerest gratitude to my supervisor Noeska Smit for valuable feedback throughout both the development of TappyBeatsXR as well as the writing of this thesis.

Contents

Acronyms	8
1 Introduction	9
2 Background	12
2.1 Extended Reality	12
2.2 The Applications of Extended Reality	15
2.3 The HoloLens	17
2.3.1 General HoloLens Concepts	18
2.3.2 Comparable Headsets	20
3 Related Work	22
3.1 Situated Visualization	22
3.2 Musical XR	23
3.2.1 Music in Extended Realities	23
3.2.2 Sound Sculpting	23
3.2.3 OSC-XR	24
3.3 Digital sequencing	25
3.3.1 The History of Digital Sequencing	25
3.3.2 Digital Music Software and Hardware	26
4 Design and Implementation	28
4.1 Requirement Analysis	28
4.2 Design	29
4.3 Technologies	30
4.3.1 Choice of Technology Stack	30
4.3.2 Unity	31
4.3.3 Microsoft's Mixed Reality Toolkit	31
4.3.4 Holographic Remoting	32
4.3.5 Version control	33
4.4 Project Design	33
4.4.1 Development Method and Debugging Pipeline	33
4.4.2 File Structure	34
4.4.3 Audio Elements	34
4.4.4 Project Structure	35
4.4.5 Expanding and Configuring the Application	35

5	Results	37
5.1	The Application Elements	37
5.1.1	The Sequencer	37
5.1.2	The Mixer	39
5.1.3	The Hand Menu	40
5.2	Controlling The Experience	40
6	Evaluation	43
6.1	User Testing	43
6.1.1	Evaluation Setup	43
6.1.2	Pilot Study	44
6.1.3	Final Study	45
7	Discussion	49
7.1	Use Cases	49
7.2	Limitations and Lessons Learned	50
7.3	Extensability	52
8	Conclusion and Future Work	53
8.1	Summary	53
8.2	Future Work	53
8.2.1	Cooperative Mode	54
8.2.2	Dynamic Instruments	54
8.2.3	Generating Content for Other Application	54

List of Figures

1.1	An organ, a great example of technology used in music. Credit: Duomaxw, distributed under a CC BY-SA 2.5 license.	11
2.1	A Venn-diagram describing the mixed reality paradigm	13
2.2	A person fully immersed in a virtual environment playing a rhythm game called Beat Saber [18]. From Szpak, Michalski, Loetscher, et al.[58].	14
2.3	A person using their phone as as an AR experience in order to capture a Pokémon in the wild whilst playing Pokémon Go [45].	15
2.4	A picture of the mixed reality headset HoloLens 2, designed and manufactured by Microsoft. Credit: Wired	17
2.5	Two people controlling a virtual hologram in the real world, interacting with it using their hands. Credit: Microsoft	18
2.6	The hand controller for the Oculus Quest 2, named Oculus Touch Hand 2. In contrast, the HoloLens has no controller.	19
2.7	An excerpt from the MRTK Examples Hub.	20
2.8	Left: The Varjo XR-3 with it's camera-centric setup. Right: The Magic Leap with it's looking-glass design. Credit: Varjo Technologies [59] & Magic Leap [28]	21
3.1	Three of the constructed environments built using OSC-XR.	24
3.2	A representation of the sound sculpting manipulation action performed on the virtual instrument by the user. Credit: Mulder, Fels, and Mase[44]	25
3.3	A screenshot of the first version of a commonly used DAW called Fruity Loops (more recently rebranded as FL Studio). Credit: www.image-line.com	26
3.4	Left: The Akai Professional Force, a popular and powerful beat machine. Right: The teenage engineering pocket operator. Credit: www.sweetwater.com & www.thirdmanstore.com	27
4.1	The MRTK allows for the user to control and simulate the inputs of a HoloLens in-editor. Shown is a screenshot of TappyBeatsXR running in Unity on a desktop workstation, using the keyboard to simulate the movements of a hand.	32
4.2	A screenshot from the Unity workspace.	35
4.3	Adding a new loop is as simple as duplicating a base prefab and adding an extra Sequencer Bar.	36

5.1	All Sequencers are half-constructed prefabs, with the amount of sequencable buttons being constructed during run-time. Some available parameters can be seen on "Inspector" to the right. . .	39
5.2	The mixer, which allows the users to configure BPM, reverb, echo, flanger and an EQ	39
5.3	The hand-tracked menu gives users easy access to an assortment of options.	41
5.4	The user will be spawned in front of two sub-loops and two transport buttons.	41
5.5	A 2D screenshot of sequencer elements. On the left: a Drum Sequencer Block. On the right: a 1-bar and a 4-bar sub-loop. . .	42
6.1	A person testing out the application for the first time.	45
6.2	Feedback on a questionnaire completed by 5 testers on 18 statements on a 5-point Likert-scale. Response statements are on a scale from Strongly Disagree to Strongly Agree.	46

Acronyms

AR Augmented Reality.

BPM beats per minute.

HMD Head mounted display.

MR Mixed Reality.

MRTK Mixed Reality Toolkit.

VE Virtual Environment.

VR Virtual Reality.

XR Extended Reality.

Chapter 1

Introduction

Modern immersive technology is evolving at an ever-growing rate, making huge leaps by the year. New avenues for the application of these technologies are constantly explored and researched all over the world by companies and researchers alike. One of the most recent examples of innovation amongst modern virtual peripherals is Microsoft's HoloLens 2. This headset combines the concepts of Augmented Reality (AR) and Virtual Reality (VR) to give you an experience called Mixed Reality (MR). MR concepts and technologies present unique opportunities to create spatial experiences in the ways humans interact with their environments and acquire spatial knowledge [11]. With its precise finger tracking, ergonomic fit and completely wireless experience, it gives you a different sensation of immersion: you use your everyday surroundings as your playground and/or workstation. The potential for using your immediate surroundings as a part of your workflow is one of the most important unique factors of mixed reality.

MR technology has previously been restricted to lab research or other specified domains. However, in light of rapid improvement of the technology in recent years, the field of MR has entered the interests of businesses and educational facilities. It has found its use in emergency preparation, civil defense and aviation training, as well as for academic and educational purposes. By infusing information into our environment through spatial computing with AR and MR, we can annotate and redesign our physical world in previously unimaginable ways. These kind of visual or auditory annotations have considerable potential to influence human spatial thinking and learning as well as everyday life [11]. By giving users a sense of space and context, remote cooperation and education can be achieved using a familiar setting and realistic proportions to mirror that of its real-life counterpart.

Music is a part of every person's life. Whether we listen to it occasionally or religiously, it is a part of our lives nonetheless. Most people do not actively partake in the auditory creation experience, and rather passively enjoy the sensation of music. There have been other music-based virtual experiences that have had enormous popularity [18], immersing people in the song itself and making you a part of its ebb and flow.

The HoloLens 2 has the capability for providing a virtual environment in which people can educate and cooperate. It furthermore has an unobtrusive and collaborative design that allows for people to participate through either remote cooperation or in-person spectating and feedback.

Motivation

The entry level for actively participating in music creation using traditional means is quite high, with most instruments or digital software requiring users to have some form of prior knowledge or skill in order to create anything tangible. Aspiring musicians can easily be discouraged by the amount of training and time it takes to get to a stage where they can enjoy themselves. The most popular pieces of music creation software are designed for professional music producers, and can seem both unappealing and tedious to non-musicians (and at times, musicians). A non-musician might not become interested in music creation or even believe that it's possible for them to create anything of substance when presented with what seem like insurpassable hurdles and unintuitive tools.

Furthermore, making music is often a shared process [5]. Music is, like many other subjects, best learned and taught in a social setting. Concepts like playing the piano, learning rhythm or producing songs are often taught by a tutor in a classroom setting, and the instinctive human nature of music makes it cooperative by nature. Most active students engage in learning by interacting with other people and their ideas [70]. Rudolph explains, "technology has long been used to improve musical expression and assist educators. The core principles of instruments such as the organ (see figure 1.1) or the harpsichord are technologically based, and shook the world of musical performing the same way computers have done in more recent times." [54]

The motivation behind this thesis is to design an interactive and engaging music creation solution that is usable by users of any skill level. By leveraging the unique features available through modern mixed reality technology, the goal is to develop an application that mixes the best from both subjects in order to appeal to users by keeping them actively engaged and seeing early results in a fun environment. This is in order to show users that if given the right context, they too can create music that sounds pleasant. The open design of the HoloLens 2 allows for ease of communication between user and the user's surroundings, enabling collaboration and tutoring to happen whilst the user is interacting with the application. This allows the experience and subsequent results to be shared amongst multiple people at the same time.

Contribution

This thesis explores the possibility of using the medium of mixed reality to create an intuitive musical interface that can be used and enjoyed by people of any user, including novice users. To this end, I created a prototype application named **TappyBeatsXR**, a music creation program for the HoloLens 2. TappyBeatsXR offers a workflow based on looping and sequencing that utilizes a controller-free interface with straightforward commands. Components created for TappyBeatsXR include: an audio sequencing program that allows developers to timely queue sounds following a musical tempo, an easy-to-use interface designed from the ground up using elements specifically designed for the MR platform, and additional ready-to-use elements that allow future users and de-

velopers to design their own version of the application with minimal coding. The thesis also explores the design process for developing an experience for the HoloLens using the real-time development platform Unity. Furthermore, the thesis serves to provide an overview of related XR technologies and music creation software.

The application was designed for and tested on the mixed reality headset HoloLens 2, but uses a combination of technologies that allows for the application to be ported onto other XR platforms, hence the name TappyBeats**XR**. The project is fully open-source under the MIT License, and is free to use and further develop by other software engineers.



Figure 1.1: An organ, a great example of technology used in music. Credit: Duomaxw, distributed under a CC BY-SA 2.5 license.

Chapter 2

Background

In this chapter, I will present and elaborate on core concepts my research builds upon. I will elaborate on the subject of various immersive technologies, and give you an understanding of key words used to describe their differences. This theory is important to know as a basis for subsequent chapters.

2.1 Extended Reality

The domain of immersive visual technologies is vast, and can at times be challenging to describe. An umbrella term used for the entire spectrum of different immersive visual technologies is Extended Reality (XR). This encapsulates all forms of visual immersion, such as Virtual Reality (VR), Augmented Reality (AR) and Mixed Reality (MR) (as well as future technologies). Suggested by Milgram and Kishino in 1994, VR and AR exist alongside a continuous spectrum, bridging the gap between complete virtuality and reality [43], see Figure 2.1. Milgram and Kishino It is important to understand the differences between these terms and their meaning in order to distinguish between the different technologies. Other popular names given to describe XR include "X-reality" and "cross-reality".

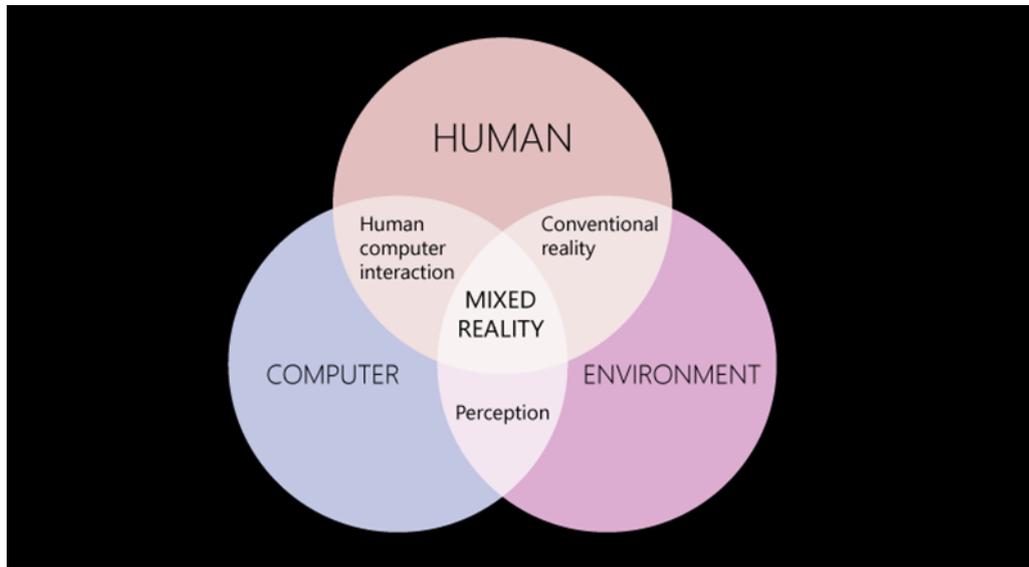


Figure 2.1: A Venn-diagram describing the mixed reality paradigm

A Virtual Environment (VE) allows the user to control a digital experience by adjusting the viewpoint and/or allowing a user to interact with different parts of the scene. Thus, movies and TV shows would not be considered in this description. According to the work by MacEachren et al. on virtual environments, four key factors have been utilized to differentiate between VR displays and other mediums such as regular MP4 videos and "flat-screen" experiences: *Immersion*, *Interactivity*, *Information intensity*, and *Intelligence of objects* [31]. Çöltekin et al. furthermore describe a defining feature of VR is the ability to fully immerse a user in a digital environment by occluding and replacing their real-life surroundings most commonly through a Head mounted display (HMD) [11]. An HMD is a wearable piece of technology that commonly includes a multitude of sensors that tracks the user's position and rotation in order for the user's viewpoint to change in correlation with their head. This is done through low-latency communication with the HMD's processing unit, allowing for new frames to render in an instant. As the name suggests, an HMD also includes a high-refresh rate display placed a few centimeters from the user's eyes. The refresh rate of an HMD is crucial for a user to avoid motion sickness or other types of discomfort. A prominent example of a popular musical VR application is the rhythm game 'Beat Saber' [18], in which you use your controllers as light sabers to cut oncoming blocks in time with a piece of music, see Figure 2.2.



Figure 2.2: A person fully immersed in a virtual environment playing a rhythm game called Beat Saber [18]. From Szpak, Michalski, Loetscher, et al.[58].

AR experiences directly use your immediate surroundings as part of the experience. By continuously tracking the user's environment, an AR experience can make objects appear as they are in the real world. As such, phones have played a massive role in increasing the awareness of the technology, by using a phone's camera and sensors to act as a video see-through display. One of the most widespread applications of this is through the massively popular mobile game "Pokémon Go", see Figure 2.3. Whilst roaming around the world, the user has to catch Pokémon around them in their physical environment, using their mobile phones as a window into an alternate reality. Due to the combination of accessibility of phones and the relatively low computing requirements of technology, AR has become the most common of the XR technologies amongst the general public. The widely adopted use of face filters in camera and messaging apps are also an example of augmented reality. Whilst VR headsets are the most common today's society, more and more headsets incorporating the full XR spectrum are showing up as technology improves by the year.



Figure 2.3: A person using their phone as an AR experience in order to capture a Pokémon in the wild whilst playing Pokémon Go [45].

Defining MR based on these distinctions can be difficult due to its wide range of interpretations and the novelty of the technology. The concept of MR has been interpreted differently by multiple authors, defining it incongruously and referring to the concept as "X-reality" [9], "cross-reality" [48] or "spatial augmented reality" [29]. According to Çöltekin et al., recent discourse regarding MR usually implies the presence of spatial referencing happening in real-time, allowing for the amalgamation of both virtual and real objects within the same spatial reference frame [11]. This allows for entities from the two mediums to interact with one another, as opposed to superimposing information in the real world (navigation arrows, warning signs), which would be considered AR [11]. Given this distinction, MR in a dark room will in principle behave like VR [3]. Also like VR, MR technology relies on a HMD in order to bridge interactions between a virtual environment and the real world. Though there is no distinct consensus on the correct use of the term "mixed reality", this is the definition that will be applied in this thesis, as it fits nicely with the technology used.

2.2 The Applications of Extended Reality

Applications using XR have been seen in different fields of education. There have been XR applications developed for the training of people in fields such as chemical engineering [1], construction [55] and flight training [68]. A common application of XR technology in education is to simulate and situate users in an immersive environment in which one can repetitively practice and explore dangerous activities in a safe, digital replica of a real-world scenario. The use of a digital workspace also gives educators the ability to access otherwise limited resources. In a virtual environment, the access to any form of tool or set of surroundings is practically limitless. These resources can include not only an expensive lab setting or airplane cockpit, but also the costly equipment commonly found in these settings [25]. XR applications also commonly facilitate some sort of input or reaction, forcing the user to be active in the learning scenario and

encouraging active engagement. In their paper on virtual reality in education, Pantelidis discuss the impact of keeping students engaged whilst learning and to avoid *passivites* [47]. Some important positive factors that the papers present include student motivation and favourable attitude towards virtual reality, [42] and the ability of virtual reality to accurately illustrate certain features or processes better than other means. Virtual reality can furthermore change the way a student learns about and relates to a subject matter. By forcing interaction between the material and the student and encouraging the student to continuously explore their surroundings, the student has the opportunity to find new perspectives and insight that would not be realized in a regular 2D visualization. The paper concludes that virtual reality does indeed have a place in education and training, and proposes a model for when virtual reality might be a good fit for a subject.

To the point of virtually replicating potentially dangerous situations, XR has found a footing in the field of medicine. Andrews et al. discuss the benefits of applying XR technology in the context of medical applications [4]. The traits of an XR application lend themselves well to work in the field of medicine, seeing as XR devices are capable of hosting programs running touch-free interfaces in sterile environments. As importantly, spatial information such as the anatomy of a patient and the position of medical instruments are valuable for training new doctors and pre-operative planning. Examples of applications applying these principles include the Stanford Virtual Heart [32] - a VR project used to explain complex congenital heart defects - and [64] - an MR application developed for teaching anatomy to medical students. The latter was in fact developed using the HoloLens 1.

Kwok and Koh discuss the idea of an increased usage of immersive technologies in tourism [26]. Whilst tourism is historically reliant on an individual's mobility and location access [10], the unprecedented global scale of the COVID-19 pandemic has the possibility of setting new norms for global travel and tourism. Due to the high contagiousness of the virus, governments around the world had to enforce inbound and outbound travel restrictions, which exposed a massive vulnerability in the tourism industry [19]. Innovation in the tourism industry is critical for its survival. With modern XR technology's ability to offer a heightened sense of immersiveness and presence, a focus on the innovation of digital tourism with the use of immersive technologies (3D virtual worlds, location-based AR experiences) seems inevitable.

In summary, XR has seen a wide variety of adaptations in various fields of both commercial, industrial and educational facilities.

2.3 The HoloLens



Figure 2.4: A picture of the mixed reality headset HoloLens 2, designed and manufactured by Microsoft. Credit: Wired

The HoloLens 2 is a second generation tether-less HMD developed by Microsoft, shown in Figure 2.4. *This thesis will hereon out refer to the headset as the HoloLens, referring to the first generation HMD as the HoloLens 1.* The HoloLens uses a combination of wave-guide and laser-based stereoscopic technology to deliver a full-color mixed reality experience. It is the technical successor to the previous HoloLens 1, also developed by Microsoft. The headset is completely controller-free, relying on its hand and eye tracking sensors, as well as modern voice recognition engine to process input from users. The headset is also not reliant on any external processing power, solely relying on its Holographic Processing Unit to perform all the calculations and rendering. This makes the headset completely free of any third party accessories, unlike many other modern XR headsets. The HoloLens bring a set of significant improvements over its predecessor, with some notables being a larger field of view, fully articulated hand and eye tracking, and a boost in processing power. The headset sports an upgraded custom Holographic Processing Unit (HPU 2.0) which specializes in delivering real-time computer vision with high energy efficiency. All computer vision algorithms (eye and hand tracking, eye tracking and spatial mapping) are ran by the HPU on-device, and is located in the front part of the headset. Applications ran on the device are powered by a Qualcomm Snapdragon 850 CPU, and is wholly available to developers. Additional features of the headset includes an Internal Measurement Unit, an RGB and a depth camera, 5-channel audio recording and spatial audio. The product first started shipping in September 2019.

The majority of the HoloLens' marketing has been focused around office and industrial work, which is reflected in the on-board applications and online demos [65]. Microsoft has designed hypothetical tutorials for the HoloLens where a user is guided through fixing a car engine, with visual indicators of the relevant area and tools for the specific issue. Microsoft has collaborated with Trimble to create a hardhat implementation of the HoloLens, specifically designed for industrial workers [21]. The accessory - nicknamed *XR10* - allows for construc-

tion work to be visualized on-site, allowing for collaborative planning and work done with the eventual construction being visualized at the location of work.

Despite the product's marketing being heavily focused around it being an MR experience, there are still misconceptions online with multiple sources referring to the HMD as an AR headset. The HoloLens allows you to operate with a VR-like interface with head-tracking interfaces and intractability, whilst still being able to have your direct environment affect your experience, like an AR experience.

2.3.1 General HoloLens Concepts

What is a hologram?



Figure 2.5: Two people controlling a virtual hologram in the real world, interacting with it using their hands. Credit: Microsoft

Microsoft refers to models being placed in the real-life world as "holograms". The HoloLens lets you view these three-dimensional objects made of light and sound that fit into your real-world environment just like everyday objects, as seen in Figure 2.5. They can be manipulated and controlled through a variety of inputs, such as voice commands, hand gesturing and eye gazing [40]. A hologram can both be a part of the environment around you, as well as interact with it. A hologram can emit sound based on its location in the world, and can change its appearance based on the lighting around it. Using the HoloLens' many sensors, a hologram can remain stationary when out of view, or be configured to follow the user around. For example, a user can place a bouncy ball over a table, and watch it bounce around the room, producing sound from whatever source it hits. As a result of the HoloLens' spatial awareness and depth perception, holograms can also appear both in front of or behind real world objects, allowing for real objects or people to occlude holograms.

What is a hologram?

With its large amount of visual, auditory and positional sensors, the HoloLens allows for a wide variety of motion and speech based input to control and explore different applications. Holograms glow white when your hand draws near them, giving you a visual indicator of your proximity to the object. You select objects



Figure 2.6: The hand controller for the Oculus Quest 2, named Oculus Touch Hand 2. In contrast, the HoloLens has no controller.

using your index finger, and can interact with various objects such as buttons and check-boxes. Using a grabbing gesture by pinching your thumb and index finger, you can move, resize and rotate 3D objects. For objects farther away, a ray is cast from the palm of your hand, appropriately named "hand ray". This allows you to manipulate and interact with holograms similarly to the grab gesture but from a further distance. This is achieved by a gesture known as an "air tap". By aiming your hand ray at an object and swiftly tapping your index finger on your thumb, you can interact with a hologram. By keeping your fingers pinched, you can manipulate is like you would with a grab gesture. For all general HoloLens navigation, speech controls are also implemented. This can be used to control certain holograms or navigate through the HoloLens' menus [35].

Through a combination of these controls, you can perform tasks such as surfing the web, resizing a solar system in your living room or controlling a cat running on your desktop.

The use of hands and joints as controller is relatively unique to the HoloLens in context of the XR domain. The vast majority of market-leading [12] headsets (e.g DPVR [15], Oculus Quest 2 [33]) use custom controllers designed for a singular headset, see Figure 2.6. Albeit having full control over the input scheme and presentation of a device-specific controller is a huge boon to controlling the experience for the user, the advancement in spatial computing technologies has enabled companies to develop interfaces that rely solely on a user's body to control the experience. This eliminates the dependency on any secondary or tertiary objects to navigate the headset, and allows for more intuitive control schemes by designing experiences around a person's motor functions.

MRTK Examples Hub



Figure 2.7: An excerpt from the MRTK Examples Hub.

The 'MRTK Examples Hub' [51] is a dedicated HoloLens 2 app available from the Microsoft Store that shows of the capabilities of the Mixed Reality Toolkit (MRTK), see Figure 2.7. The application highlights a lot of the unique features in the MRTK as well as serving as a good example of what the HoloLens is capable of. It has different dedicated experiences surrounding types of input, how to design and interact with a HoloLens 2 application, as well as providing examples and ideas like playing a keyboard in MR. This was one of the first applications that were tested on the HoloLens as part of development, and a lot of the core principles on how to design an application specifically for the HoloLens came from this program. It was used as a guideline in terms of how one expects holograms to operate, different applicable animations used to visual clarity as well as the overall scale and positioning of holograms.

2.3.2 Comparable Headsets

The HoloLens is not the only headset striving to bridge the gap between the human and machine with the use of head mounted displays. One headset that strives to deliver a comparable experience to that of the HoloLens through different means is the Varjo XR-3 [60], see Figure 2.8. Both headsets utilize the user's hands and voice as their main controls, and despite its misleading name, the XR-3 is a mixed reality headset. The XR-3 uses a 70 PPD video camera with 115 degrees field of view, capturing and configuring video at low latency before delivering the image to the user in real-time. This differs from the HoloLens's see-through virtual-based retinal display where you are able to see your surroundings directly, and objects are created and put into the scene through light technology. The XR-3 has seen use in clinical and commercial research due to its high resolution display, whilst the HoloLens lends itself more towards use cases in which interacting with your surroundings and keeping you grounded in your environment is more important [52].

Another competitor for the HoloLens is the Magic Leap 2 [23], see Figure 2.8. Unlike the two HoloLens editions and the Varjo XR-3's MR focused marketing, the Magic Leap presents itself as a pair of AR glasses. It sports a 70 degree field of view and has a focus on fidelity of objects, being able to render 3D objects at a higher definition than the HoloLens. It also sports a "*dynamic dimming*" feature, allowing for the headset to dim any ambient light to increase the clarity of the generated 3D objects. It sports a "goggles-like" design with a

tethered computing box that the user has to carry around and handle as they put on or take off the headset. As opposed to the "visor-like", self-contained flip-up design on the HoloLens, which integrates all hardware in the headset itself as well as giving you the ability to remove the projected imagery simply by flipping up the headset [27]. Furthermore, the goggle design of the Magic Leap blocks a portion of your peripheral vision in order to grant an increased sense of immersiveness. The HoloLens, in contrast, gives you an almost completely unobscured view of your surroundings. These two design decisions on the Magic Leap decrease a user's awareness and reactivity to their environment. As both headsets market themselves to sectors like manufacturing and healthcare, the HoloLens definitely has an advantage for ease of use with its uncomplicated consumer-centric design. Whilst the design of Magic Leap might not be as light and open as that of the HoloLens, its higher quality image and increased immersiveness might be useful for certain use cases.



Figure 2.8: Left: The Varjo XR-3 with its camera-centric setup. Right: The Magic Leap with its looking-glass design. Credit: Varjo Technologies [59] & Magic Leap [28]

Chapter 3

Related Work

This section explores related work, both in the realm of music applications as well as general applications for the HoloLens. As mentioned in Chapter 2, the area of musical applications for the HoloLens remains largely unexplored, and the relevant scientific literature and related works in the XR domain will therefore be discussed at a more general level.

3.1 Situated Visualization

Most types of visualizations are typically displayed outside their relevant context, on a stand-alone device outside the environment in which the data is applicable. The area in which the data is displayed will usually serve no purpose as to increase the clarity and readability of the visualization itself. In the work done by Bressa et al. on exploring the research field of *situated visualization*, they describe the term as encapsulating the idea of bringing data into context by utilizing the real world environment or physical counterparts [6]. For instance, having a spatial visualization of a colored cloud representing the amount of pollution recorded at certain real-world locations would be considered "situated visualization". In the context of MR, the mixing of virtual and real life environments and objects could be interpreted as part of this field of research. By designing experiences around marrying digital assets with physical environments, one develops a specialized situated context for any application, forming a two-way dependant relationship between the real and digital world.

White and Feiner simply explain the term situated visualization as "is related to and displayed in its environment". [6] However, due to the wide adaptation and appropriation of the term, a consensus on the definition of the term remains unclear. The term has been used for fields such as data visualization, AR and urban computing, as well as various other fields such as physicalization.

TappyBeatsXR uses situated visualization by incorporating certain elements into the environment with the use of *spatial mapping* [39]. Spatial mapping provides a detailed representation of the environment surrounding the HoloLens, allowing for developers to design holograms that can be placed in and interact with the real world, allowing for users to truly experience mixed reality. The

spatial mapping provided by the HoloLens is furthermore dynamic, providing a low-latency stream of updates to any changed mesh regions. Spatial mapping allows for holograms to have the ability for more realistic real world interactions that align with a user’s expectations. One might be carrying a virtual coffee cup that when dropped and affected by (virtual) gravity, will hit the table and fall over. TappyBeatsXR uses spatial mapping to incorporate a virtual mixer into the environment by giving users the ability to anchor the hologram on any surface. This infuses the application with a sense of both virtual and real-world spatial sense, allowing for users to move freely whilst interacting with their virtual dashboard but returning to the mixer in the real-world location to change the sound of the music generated.

3.2 Musical XR

There have been other instances of researchers proposing the marriage between music and XR, and a term coined for this is *Musical XR*.

3.2.1 Music in Extended Realities

In their work on music in extended realities, Turchet, Hamilton, and Çamci discuss both the history of and the potential behind the interconnection of musical interaction and extended realities [61]. Extended reality applications can allow for both artists and audiences to actively participate in shaping musical processes in virtual, immersive environments. One of the key concepts behind Musical XR is the requirement for applications to be audio-first and drive the experience through engagement by the user, as opposed to playing a more simple or passive role. The user should also be situated in the same 3-dimensional space as the musical elements in order to maintain spatial relationship between the virtual elements and the user. In their work on researching virtual reality instruments, Serafin et al. also discuss the importance of leveraging the medium of XR to explore new opportunities for virtual instruments, as opposed to attempting to duplicate already existing real-world instruments [56]. Additionally, other authors have proposed the idea of using XR-based tools to support compositional processes [7, 8]. Actively participating with your environment whilst combining both familiar sounds and re-imagined visual surroundings offer a range of potential for creative and musical expression. Atherton and Wang elaborates on good principles behind artful XR design, and stresses the importance of immersion, presence and agency [5]. TappyBeatsXR incorporates many of the same principles brought up by these authors, focusing on the user being present within the experience and allowing for them to interact with their composition as it is being created.

3.2.2 Sound Sculpting

Mulder, Fels, and Mase proposes as a form of musical expression in an XR environment [44] called ”sound sculpting”. Sounds are created and manipulated by a user virtually sculpting sounds in real-time, adjusting the properties of the sound by manipulating shapes, position and orientation. In their implementation of the concept, they constructed a virtual surface that the sound composer

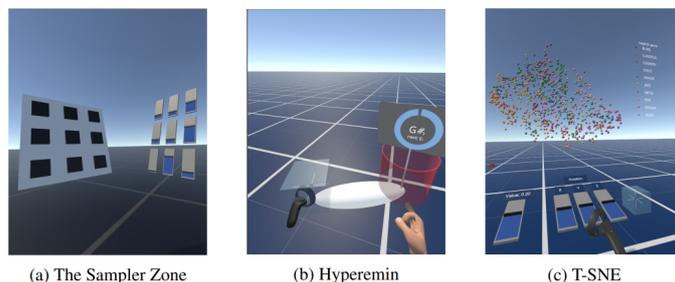


Figure 3.1: Three of the constructed environments built using OSC-XR.

manipulates. To do this, they used a pair of gloves called Cybergloves that track the hands shape, as well as a sensor for tracking position and orientation, see Figure 3.2. Two virtual objects were constructed - a rubber sheet and a balloon - that the user was able to manipulate. One of the findings from a study performed on 15 participants was that the lack of tactile and force feedback made correctly manipulating these virtual objects difficult on a precise level. The lack of proper depth clues in particular made precise manipulation hard to achieve. This too is a persistent issue on the HoloLens that will be discussed in later chapters. Furthermore, another conclusion drawn from the study is the difficulty of adapting the behavior of such objects in a way that is natural to correctly manipulate. Although a variety of implementations were possible, significant technical expertise was required to reliably control these virtual instruments. Although not fully incorporating "sound sculpting", TappyBeatsXR focuses on the user's hands being their musical conduits. Both through the act of sequencing notes, as well as adjusting the sonic qualities of the audio produced.

3.2.3 OSC-XR

OSC-XR is a prototyping toolkit developed by Johnson, Damian, and Tzane-takis for designing music controllers in XR environments [24]. The toolkit consists of a set of prefabs, a collection of objects that are ready-to-use, designed to operate as virtual music controllers. Some of these include a virtual gyroscope, a drum pad prefab and a configurable slider, see Figure 3.1. Included in these prefabs are a multitude of scripts that enable the developer to transmit the state of the prefabs to other objects. Although Unity - the real-time development platform used in this thesis - has an audio engine, it does not support audio synthesis. The authors therefore also incorporate an audio synthesis engine called Chunity in order to dynamically generate sounds by sending the controller parameters to the synthesis engine. The authors also designed a set of prototypes using these tools. One of these prototypes is the Sonic Objects zone, an environment in which the user can manipulate 3D objects in order to change the characteristics of a sound, including pitch and volume. A demo of the toolkit's capabilities can be seen in a video demo on the projects Github page ¹.

¹OSC-XR Github page: <https://github.com/davidsjohnson/OSC-XR>

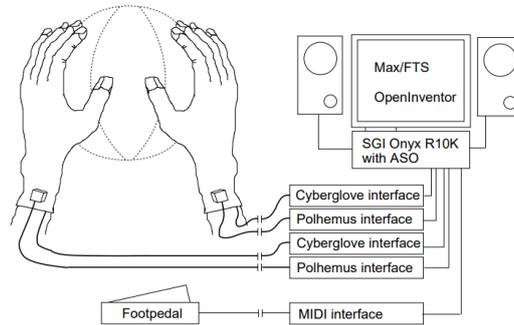


Figure 3.2: A representation of the sound sculpting manipulation action performed on the virtual instrument by the user. Credit: Mulder, Fels, and Mase[44]

3.3 Digital sequencing

As mentioned in the introduction ??, technology has always played a huge part in advancing the field of music. The idea behind this thesis is to develop a functional music workstation with the use of XR. The design of the application is based off of a workflow called "sequencing", and this section will cover its history and modern implementations of the concept.

3.3.1 The History of Digital Sequencing

The core principle behind TappyBeatsXR is a short-form looping mechanism, utilizing a grid-like structure for inputting notes and beats in a sequence over time. This way of working is a fundamental idea in modern digital music creation, and is known as "sequencing". Digital sequencing of music did not become popular until the invention of *musical instrument digital interfaces*, more commonly known as *MIDI*. In 1981, the founder of the Roland Corporation - Ikutaro Kakehashi - proposed the standardization between different companies' communication protocols which could also communicate with computers. Previously, hardware would commonly be controlled by two different electrical signals: a **control voltage** to specify the note (CV), and a second boolean electrical signal to play the note (gate). These signals would be carried across thin patch cables to a synthesizer, that would in cooperation with its own synthesizing engine produce a sound, with the pitch controlled by the CV. This method of synthesis is known as **analog synthesis**. Despite vast advancements in technology, analog synthesis has remained quite prevalent in the music industry, and has even recently seen a resurgence in popularity [14]. The first digital MIDI sequencer was the Roland MSQ-7000, released by Roland in 1983. The following widespread adoption of the MIDI protocol spawned many computer-based MIDI sequencers, and converters were made to allow CV and gate controlled synthesizers to be controlled by this new digital protocol. MIDI remains the industry standard for controlling musical instruments to this day.

3.3.2 Digital Music Software and Hardware

Dedicated tools for processing and designing audio in form of loops or songs are called **digital audio workstations** (DAWs). Many of the popular audio workstations today like Fruity Loops [22] and Ableton Live [2] started off as simple sequencers or loop machines, allowing for users to control small chunks of audio or sequence different notes and drum sounds, see Figure 3.3. [22], see Figure 3.3. This high-level and low-cost entry into the world of music production was the start of today’s ever-rising popularity of ”bedroom production”. Artists that started started off their laptop with a laptop and a microphone are today winning Grammys next to famous industry musicians, proving that deep pockets and corporate capital are not prerequisites to generating creative output [67].



Figure 3.3: A screenshot of the first version of a commonly used DAW called Fruity Loops (more recently rebranded as FL Studio). Credit: www.image-line.com

A callback to this era of loop-based music development has also sparked other companies to make dedicated hardware that follow the design pattern of early DAWs. Notably, the reputable synthesizer company **teenage engineering** have in recent years designed a plethora of pocket-sized synthesizers with integrated looping functionality called **pocket operators**. These small synthesizers have a very inviting and nonthreatening appeal, as it’s design reminisces a toy or a calculator. In contrast, bigger and more powerful ”beat machines” usually have a vast array of knobs and buttons that takes days or even weeks of training to be able to use properly, see Figure 3.4. Albeit being designed for different purposes, easy access to a portable form factor with great potential of creative and mellifluous output has inspired a vastness of internet users to cover compositions dear to their heart and post them online.



Figure 3.4: Left: The Akai Professional Force, a popular and powerful beat machine. Right: The teenage engineering pocket operator. Credit: www.sweetwater.com & www.thirdmanstore.com

Chapter 4

Design and Implementation

This section will elaborate on the design choices and decisions that went into making the final application. I will discuss the different technologies used and implemented; the program used to make the application, the technologies applied when designing and prototyping as well as the structure of the back-end.

4.1 Requirement Analysis

Before starting development, a set of requirements were drafted. These were based off of a combination of factors. When initially testing the HoloLens, I invited several students to test it and see what their experience was. They were shown the main menus of the HoloLens, as well as some applications downloaded from the store. Notes were taken down regarding what they found difficult with operating the HoloLens. Previous experience with showing acquaintances different VR applications also played a role. Lastly, my own experience in learning different musical subjects such as the piano, music theory and the use of digital music software played a role in deciding these requirements. Some key pillars that were central for the development of this thesis were:

- R1 The controls should be simple enough so that a user without any prior XR experience can properly sequence notes and make music.
- R2 Interaction between holograms should be controlled by the user as opposed to automated, to force interaction with the application.
- R3 The number of elements in the interface should be as low as possible to promote ease-of-use.
- R4 Any additional visual flavor interacting with the interface should be informative in nature as to avoid visual clutter and distractions.
- R5 A non-musician should be understand the purpose of the sequencer and looping interface and how they co-exist.
- R6 A non-musician should have the opportunity to pick notes by intuition and personal preference, as opposed to knowledge of music theory.

- R7 The user should be able to have some control over the sound character of their constructed piece of music, either through sound selection or audio effects.

4.2 Design

Doing complicated tasks in MR can be quite tedious and difficult to perform by users of any skill level. With the controls on the HoloLens being solely based on hand tracking, gaze tracking and voice controls, they are easy to pick up on initially. However, there are a lot of minute details in the interfacing that does indeed pose a slight learning curve for newer users. Albeit the general understanding of the control scheme is quite natural, the HoloLens is a tad finicky in how exactly it wants your hands to operate. To perform more complex actions, your fingers are required to be at a certain depth with a specific stiffness, and using its more elaborate transformational commands require a bit of experience.

With these concerns in mind, it was important to keep the main tasks of the application simple in terms of control schemes. As sequencing notes is the most consistently repeating action in the application, it was crucial to design it in a way that was easily comprehensible and understandable. Initial ideas revolved around using the position of the user's hand in order to select certain notes, using a three-dimensional space to select different sounds. Despite this being a good way to display the quality of the HoloLens' hand tracking, it took away from the core concept of making a functional musical application in mixed reality. Ultimately, an important inspiration for the layout and workflow in this part of the application were the pocket-sized synthesizers referenced in ??, the pocket operators. The pocket operators serve a very functional sequencing workflow into a very small form factor, with a set of buttons on a grid being the main operating tools. The proven effectiveness of this primitive layout led to the decision of relying on a matrix for sequencing notes.

Another major concern for the application was timing issues in regards to audio. As the HoloLens is a completely tether-less headset, all its calculations for hand tracking, face tracking, spatial mapping et cetera happen in the on-board processor. The amount of available resources is completely dependent on multiple factors, such as amount of polygons currently being projected, amount of new spatial information in queue for processing, and any application-specific code being run on the headset. As TappyBeatsXR is a musical application, temporal accuracy is of the utmost importance. Adding all these factors together, designing the application to be as optimized as possible was necessary for a smooth user experience. As such, a lot of initial visual ideas such as generating and controlling holograms in time with the beat had to be scrapped.

Design Inspiration

Designer Daniel Escudero and artist Martin Wettig have a captivating article on Microsoft's pages describing their experience with designing an experience for the HoloLens [16]. The process of having to scale back initial ideas to make room for what was feasible to represent in any given environment, as well as appropriately applying artistic touches to their project to increase clarity, inspired the later stages of development. Specifically, concepts that seemed

clear through development were hard to grasp for some users. Applying a precise amount of visual clarity within the frames of what's possible on the HoloLens made the flow of the application way better, and as a result yielded a much better experience for users.

4.3 Technologies

One of the primary objectives of the application developed for this thesis was to explore the feasibility of introducing a musical workstation into a mixed reality setting. To this end, the HoloLens was the preferred choice due to its user friendliness, compared to its competitors discussed in Section 2.3.2. After the decision of hardware, extensive research was done on possible development paths. Multiple combinations of software are possible when developing for the HoloLens, including different engines, IDEs and additional toolsets and standards. This section will cover the choices made and potential alternatives.

4.3.1 Choice of Technology Stack

The HoloLens is supported by many common real-time AR and VR development platforms. A few of the most popular platforms for immersive experiences include Unity, Unreal Engine and Vuforia, and choice of platform was an important problem that had to be solved early in development. Being a relatively new piece of technology, I found it best to follow the suggestions by the developers of the HoloLens - Microsoft. Key points that were heavily weighted for the decision were available documentation for MR development on the HoloLens, as well as the platform's main back-end scripting language.

Following the official HoloLens documentation by Microsoft [34], the two main engines that are suggested by the developers and are most extensively documented are Unreal Engine and Unity. Amongst these two, Unity has an advantage over Unreal Engine in additional relevant features. For instance, Unity has support for world locking tools on the HoloLens such as Azure Spatial Anchors. Spatial Anchors allows a user to save or fetch real-world coordinates by communicating with Azure's servers in real-time. This enables a user to save an instance of a digital workspace or exhibit in a real world location, and for other users to load a replica of said instance at a later date. This was a feature considered for the application, to allow a user to save their loops and for other users to load them back up in the same location at a later date to continue work. Unfortunately, this feature was scrapped due to time constraints, but played a role in the choice of engine.

Additionally, Unity's native scripting language is C#, in contrast to Unreal Engine relying on C++. A lot of back-end code had to be written for the sequencer engine, so a lot of time during development had to be spent creating scripts in the specified language. Additional interaction scripts would also have to be written in the same language in order for the application to behave correctly. Having extensive knowledge and experience in Java, picking the language that was most similar to Java's workflow and structure was an important factor. In that context, C# is far closer to Java than C++. Performance optimization was important for this application, and though it is safe to assume that an ap-

plication written in C++ runs faster than its C# counterpart, the trade-off in increased development time would not be worth the sacrifice. Moreover, Unity has specially designed optimization tools that simplify the process of optimizing for a HoloLens application.

The aforementioned points ultimately led to Unity being decided on as the platform and development engine for the application. Furthermore, the application uses Unity’s support for the Mixed Reality Toolkit and OpenXR. OpenXR[46] is a free and open standard for XR platforms, allowing for cross-platform development and working to solve the fragmentation in XR by simplifying development and allowing access to XR device runtimes on a multitude of platforms. These were both crucial tools extensively used in the development of the application, and will together with additional notable features be covered more in-depth in later sections.

4.3.2 Unity

Unity[63] is a cross-platform real-time development game engine available for PC and Mac. Unity can be utilized to create 2D and 3D games and experiences, and supports a variety of platforms. It is one of the most cutting edge engines for developing XR applications, and was used to develop Beat Saber and Pokémon Go, two of the video games referenced in 2. Real-time changes can be applied during development with the engine’s inspector, allowing you to assign references to objects manually in-editor as opposed to fetching objects through scripting. It includes a plethora of convenient features such as a package manger for additional add-ons, as well as an online asset store.

The engine’s ease-of-use combined with it’s extended set of tools and features has made Unity one of the most popular game engines on the market. As of 2018, Unity powers about fifty percent of all mobile apps, and approximately ninety percent of all applications on newer augmented reality focused platforms such as the HoloLens [13].

An important piece of documentation that could be overlooked by future developers is the choice of the Unity version used. At the time of writing, Microsoft’s recommended version Unity configuration for HoloLens and Windows Mixed Reality development is Unity 2020.3 LTS with the Mixed Reality OpenXR plugin. However, as mentioned in other parts of the documentation, the 2020.3.21f, 2020.3.22f and 2020.3.23f versions are to be avoided, as they have been officially recognized to cause flashing and strobing issues on the HoloLens [49].

4.3.3 Microsoft’s Mixed Reality Toolkit

The Mixed Reality Toolkit (MRTK)[41] is a project by Microsoft that aims to provide important sets of tools and features to speed up cross-platform development of mixed reality applications. The MRTK provides a multi-platform input system for a multitude of platforms such as Oculus, OpenVR, OpenXR and Windows Mixed Reality. The toolkit includes custom designed shaders and UI spatial interaction elements designed for optimal performance in mixed reality. The toolkit furthermore provides an array of *observers*. Observers function as a bridge between your development platform and the data collectors on the

headset. They allow you to use an API to communicate with the sensors of the headset to fetch and configure data such as scene understanding (spatial meshing), camera footage and audio.

MRTK ships with input simulation for devices, and allows for the developer to reproduce the input for any supported device in-editor. This enables the possibility to test the application with the intended controls in a desktop setting without reliance on the direct access to the headset. Moreover, it allows instant testing of the interaction with UI elements and between objects in the scene without need for compilation or building.

The toolkit's support for OpenXR was extensively used in this project, as it enabled for holographic remoting..

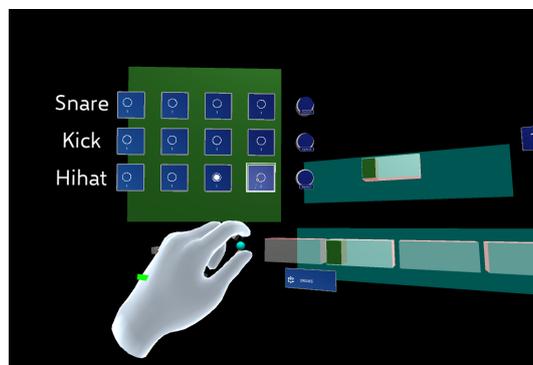


Figure 4.1: The MRTK allows for the user to control and simulate the inputs of a HoloLens in-editor. Shown is a screenshot of TappyBeatsXR running in Unity on a desktop workstation, using the keyboard to simulate the movements of a hand.

4.3.4 Holographic Remoting

OpenXR provides a useful tool for debugging apps for the HoloLens - *Holographic Remoting*[36]. This allows you to stream holographic content directly from an application on your PC to your HoloLens over WiFi. This can be a timesaver during development, as it lets you playtest your application at a moment's notice, skipping the 2-step build phase. Holographic remoting also uses the host PC's processor to serve the application, allowing for bigger and more powerful applications to be run. Holographic Remoting uses a companion app on the HoloLens by the same name, along with the HoloLens' IP on the network. A connection is formed between Unity and the HoloLens, and data is transferred between the two parties with very little latency. The HoloLens feeds its sensor data to the application, allowing for Unity to handle different observers like gaze tracking and spatial mapping, whilst Unity returns the holograms to be rendered on the HoloLens.

There are two versions of holographic remoting; one can either build the project as an executable that is hosted in fullscreen and streamed from your PC, or one can run the program straight from the Unity editor. The former is more useful for process-heavy applications that should be powered by a desktop computer,

as it streams the complete field of view with very little latency. The latter is optimal for debugging, as it gives you a fraction of the field of view, but takes mere seconds to get running on your HoloLens when correctly configured.

4.3.5 Version control

Version control was crucial for the project, due to the many breaking iterations and constantly changing initial sketches. The files needed to be accessed from multiple locations due to the constantly changing regulations of the Covid pandemic, so the choice of a reliable, cloud-hosted version control system was crucial. Although Unity Collaborate was considered due to its close integration with Unity, the interface and its ongoing deprecation made it a poor choice. The use of the Git version control system along with Github for remote hosting was chosen as the version control system for the application, with Github Desktop chosen as the interface.

4.4 Project Design

4.4.1 Development Method and Debugging Pipeline

For this project, a sprint style engineering method was applied. Different functionality of the application was broken up and the required development artifacts was separated to their own section. Each artifact was then tested through a 3-step process. Each part was then tested using a combination of PC interfacing, holographic remoting and finally a complete deployment over Wi-Fi.

The first debugging step is using Unity's play mode, interacting with the application through a keyboard and mouse. The MRTK has built-in keybindings and different configurations for Unity's play mode, allowing the developer to simulate the input of the HoloLens from the convenience of their computer. The include all basic functionality such as controlling the joints of a hand, the transform of the entire hand, air-tapping, as well as rotating and dragging object. The input is however limited when compared to the actual HoloLens, and one can not appropriately test the experience of using the application designed for a wearable headset. Furthermore, functionality tied to the AR aspect of the headset can also not be appropriately tested in a fully digital environment.

The next step in the pipeline is making sure the controls and holograms behave appropriately in the context that they were designed for. The most efficient way of testing the application for its intended environment is with holographic remoting, as discussed in section 4.3.4. This greatly speeds up the "PC-to-hardware" process, allowing for an online HoloLens to be ready to debug in mere seconds. One of the drawbacks of this method is the profiling and performance testing. When running the application through holographic remoting, the performance of the application might be outrageously better than reality, as a PC is performing all the difficult calculations, as opposed to the HoloLens' portable battery-driven processor.

Another drawback discovered in the development process is that, at least in Unity 2020.3.17f1, if holographic remoting is enabled and the HoloLens is not

discovered on the assigned IP, the entire program will halt and crash, giving no indicator of the HoloLens not being reached.

The final step in the debugging pipeline is a full deployment. This process starts with making sure all configurations are appropriately assigned in the editor, and that there are no current breaking errors in the project. Any breaking run-time bugs will have been discovered in either step 1 or 2. After Unity's build settings are appropriately configured for the correct deployment platform, Unity builds your project to a folder containing a Visual Studio solution file. At this stage, developer mode should have been enabled for the HoloLens. After every unique build, the Visual Studio deployment settings need to be re-checked, albeit building to the same solution folder reduces the build time. The application is then built in Visual Studio, and sent to the HoloLens as a runnable application. This will be the actual experience of your application - the performance, controls, holograms and audio. Some notable drawbacks with this debugging step is the time it takes for the entire process to complete. Both Unity and Visual Studio requires a great deal of time to build the project, and for debugging your application during development, this step is simply not feasible to do with every change.

Step 2 and 3 can also be completed using a USB-C cable connected from the HoloLens to the computer, but this was unfortunately not a possibility during this thesis, as discussed in Section 7.2.

4.4.2 File Structure

The sounds are imported directly from the project's Resources folder, and the sequencers will name each different sample according to the filename. Basing the sound import system around Unity's Resources folder allows for different samples to be loaded during runtime, making it trivial to design a system to swap out the initial sounds with a more fitting set at the user's convenience. This opens up for the potential of "sound packs", giving the user a choice in which type of music they'd like to create.

4.4.3 Audio Elements

All samples for the application were designed and exported from Ableton Live 11 [2] using stock effects and plugins. Both the bass and lead were created using the 'Operator' synth plugin, a relatively simple synth with parameters that are easy to control. The lead was created by combining a saw wave and a sine wave with a short tail to create a stabby and plucky sound with a percussive feel. The bass was created by adding a lo-pass filter to a dark bass note with a long tail, adding a much needed foundation to the selection of sounds. The drum sounds are a set of stock hi-hat, kick and snare sounds from the stock Ableton packs.

It was important for the small selection of sounds to be instantly pleasing to the users, due to the target audience potentially not being well versed in music. Therefore, it was decided that all notes available for the bass and lead instruments would from the C pentatonic scale. The 5-tone pentatonic scale is largely considered to be the most innate and universal scale, with songs in the

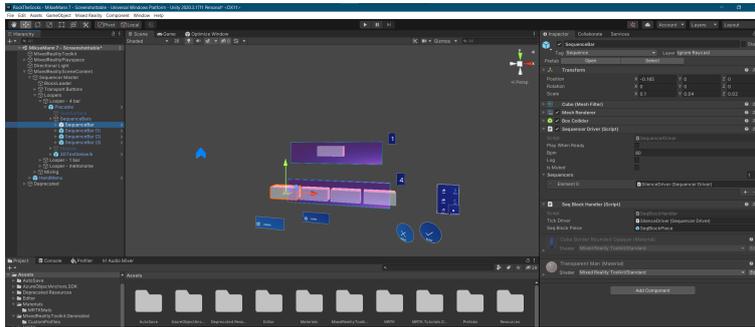


Figure 4.2: A screenshot from the Unity workspace.

pentatonic scale showing up specifically in folk music from countries all around the globe. Work by Van Khe on the universality of the pentatonic scale describes how both their own research as well as that of ethnomusicologists have discovered this exact scale in separate places on different continents [66]. Bobby McFerrin also brilliantly demonstrated the power of the pentatonic scale at the World Science Festival in 2009 [20]. By giving the audience two notes, a random selection of people were able to ascertain the next note innately to continue the melody. The C major pentatonic scale is also one of the most common in modern pop music, as it is one of the simplest to perform on a piano. Therefore, the notes C, D, E and G were chosen as the notes for the application.

4.4.4 Project Structure

TappyBeatsXR was developed using Unity as the platform and Visual Studio Code for C# writing and debugging. The project's assets include a multitude of scripts and objects made for the sequencer engine, as well as all custom sound elements. It further includes a set of reusable prefabs that are easily configurable to allow future developers to use the Unity project to create their version of TappyBeatsXR. The prefabs and Unity project structure have furthermore been constructed in a way so that minimal additional coding would be required to adjust the existing version. For example, a teacher can solely use Unity to add additional sub-loops, adjust the layout, change the sounds used or remove options from the interface. The Unity objects (GameObjects) are organized by category and appropriately named according to their function. Multiple parameters of the GameObjects' scripts are also publicly exposed to the editor, allowing for adjustment of e.g default BPM to be customized. A screenshot of the project layout can be seen in Figure 4.2.

4.4.5 Expanding and Configuring the Application

The majority of functionality in the application is compiled during run-time, such as Sequencer Blocks, loops and even sounds. That is to say, all interactive and changeable elements are created based on a set of parameters, and are then spawned into the application following these parameters. This opens up for the possibility of developers or users with limited coding knowledge to design the application to meet their needs. A lot of work has been done in

order to make the application as flexible and configurable as possible to open up for future work – at both a high and low level. Furthermore, as stated in Section 4.4.2, the samples used in the application are embedded in Unity’s ”Resources” folder, and are loaded into memory as the application starts. As the back-end audio engine designed for the application fetches all samples from this folder, this design open ups for the possibility of swapping between samples or ”sound packs” while in the experience. This feature has unfortunately not been included in the current iteration, but the code and scripts have been designed around the implementation of this feature.

The current way of configuring the application is through the Unity editor. By duplicating the Loop prefab, adding or removing sequencer bars will create a new, fully-functioning looper (as demonstrated in Figure 4.3).

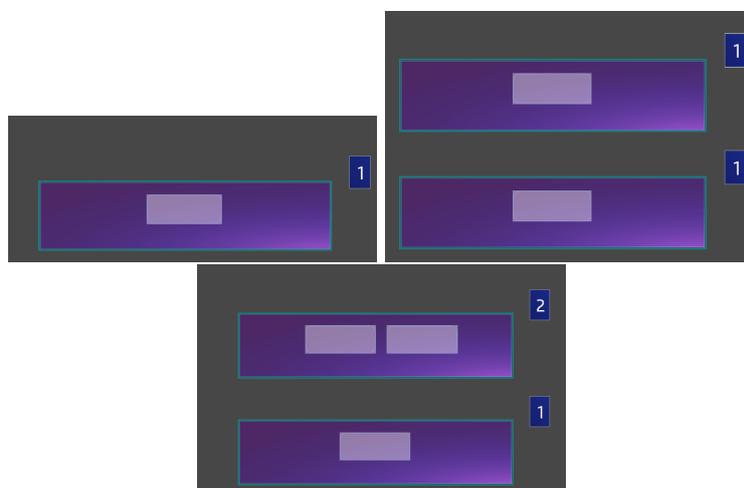


Figure 4.3: Adding a new loop is as simple as duplicating a base prefab and adding an extra Sequencer Bar.

Chapter 5

Results

The application developed during this thesis - TappyBeatsXR - is a novel and fully-functioning music sequencing and looping software designed for XR environments, using the mixed reality headset HoloLens 2 along with its movement tracking and spatial interfacing to deliver a unique and immersive musical experience. The controller-less interface is designed for non-musicians and people with no previous XR experience. This is done by relying on a simple "click-and-drag" control scheme to operate the application. TappyBeatsXR can create programmable sequencer blocks using instrument samples, and the application currently has 3 main sounds: drums, lead and bass. The resolution at which these notes can be sequenced is completely independent, allowing users to combine any type of rhythm-pattern using the different instruments. These individual elements - called Sequencer Blocks - can be created and subsequently programmed using clear visual indicators in a grid-structure. These blocks are used in conjunction with a custom-made audio sequencer engine built in C#. The Sequencer Blocks are added to the audio sequencer by the use of a drag-and-drop system, using multiple Sequencer Blocks to produce a looping piece of music. A pop-up hand-tracking menu allows for users to create Sequencer Blocks from the list of available instruments, In addition, a surface tracked mixer allows for the user to place and control the sound character of their loop from anywhere in a room. The mixer exists of 4 different slider that affect the sound of the music produced from the loop, as well as the speed of the loop, and can be adjusted and interacted with either before or during run-time.

5.1 The Application Elements

TappyBeatsXR consists of an assortment of elements designed for specific purposes, and this section will elaborate on how they work and how they were designed.

5.1.1 The Sequencer

For this application, a custom Unity audio sequencer was built, based on an existing codebase [30]. A **sequencer** is a grid-based interface that allows you to

sequence independent timesteps using different samples and instruments. The core of the sequencer is based on Unity's *AudioSettings.dspTime* metronome example [62]. It was important for the audio sequencing engine to be based on a measure from Unity's audio engine, as the time values obtained through the *.dspTime* function are based on the actual number of samples the audio system processes, and is therefore much more precise than the time obtained via the *Time.time* property. At its core, the sequencer is a boolean array that will queue sounds for playback. Each sequencer has one dedicated sample such as a kick drum or a piano key, and for each sequencer block, multiple different sequencers are attached. The sequencing system is fully abstractable, with all sequencers being an implementation of the *SequencerBase* class. Derived from this is the *Sequencer* class, which behaves as previously explained. Any number of sequencers can then be simultaneously controlled through a *SequencerDriver*, and due to its inheritance, any number of *SequencerDrivers* can be added to and controlled by a separate *SequencerDriver*. Acting as a good starting point, the sequencer subsequently had to be severely modified in order to work for TappyBeatsXR. Further explanation regarding sequencers will surround the modified TappyBeatsXR sequencers that is in the final application.

All the aforementioned classes are played and controlled by a *Sequential Sequencer*. This class controls all *Sequencer Drivers* and allows for them to be played and configured individually in succession as opposed to simultaneously. The *Sequential Sequencer* keeps track of all relevant *Sequencer Driver* changes during run-time, allowing for any part of the loop to be modified and changed as the loop is playing and progressing.

A *Sequencer Block* is a hologram with clickable buttons for each of the sequencers. Interacting with the *Sequencer Block* will allow users to decide at what point a sound is to be played, and there are multiple different *Sequencer Blocks*:

- **Bass** - a green backplate with the different notes displayed to the left of the sequencer.
- **Drums** - a red backplate with the specific sound name displayed on the left (kick, snare, etc).
- **Lead** - a blue backplate similar to bass, with the different notes displayed on the left of the sequencer.

A drum sequencer can be seen in Figure 5.5. Each *Sequencer Block* has a single word describing the sound available for sequencing, with melodic *Sequencer Blocks* having assigned notes whilst percussive *Sequencer Blocks* are assigned the name of the percussive element. Each individual sound can be previewed by playing it from the *Sequencer Block*.

Each *Sequencer Block* is queued up and controlled asynchronously by a *Sequential Sequencer*, see Figure 5.5. For intelligibility, a *Sequential Sequencer* will from here on out be referred to as a *sub-loop*. A *sub-loop* serves to break up a single loop into different lengths, allowing for the user to control smaller portions of the loop as opposed to adjusting the entirety at once. Each *sub-loop* is broken up into multiple *Sequencer Bars*, as will be elaborated on in Section 5.2. The *sub-loop's* asynchronicity allows for variable tempo playback on each

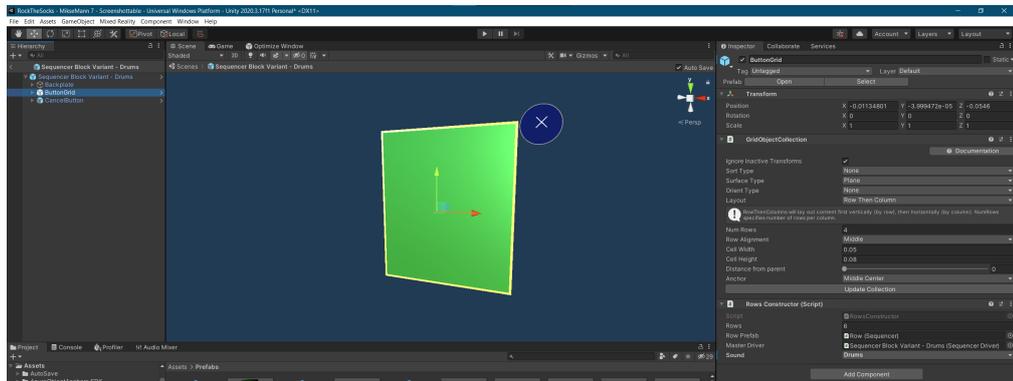


Figure 5.1: All Sequencers are half-constructed prefabs, with the amount of sequencable buttons being constructed during run-time. Some available parameters can be seen on "Inspector" to the right.

different Sequencer Block, enabling for potential interesting rhythms such as 5/4 hi-hats on a 4/4 kick loop. Furthermore, the sequencer is fully modular, allowing for any number of sequences to be played concurrently from different sub-loops.

5.1.2 The Mixer

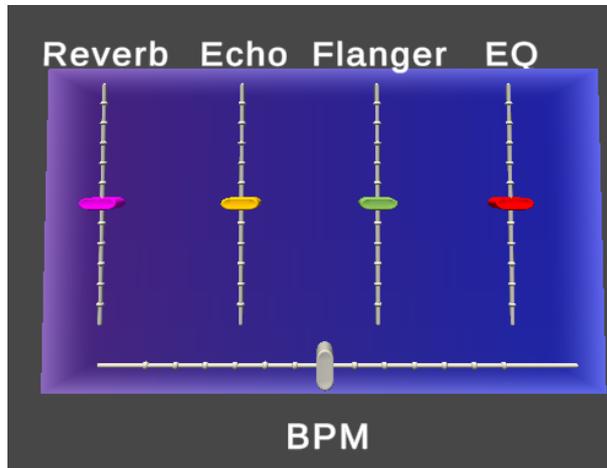


Figure 5.2: The mixer, which allows the users to configure BPM, reverb, echo, flanger and an EQ

Along with the aforementioned looping based interaction elements, a configurable mixer is also added to TappyBeatsXR, see Figure 5.2. The mixer has a distinctly different design from the rest of the blocks as to avoid confusion between Sequencer Blocks and the Mixer, and is spawn-able through a button below the loopers. The mixer specifically makes use of the HoloLens' built-in mesh mapping, enabling it to be placed on any surface in the room by the pro-

cess of air-tapping. It will follow the rotation of the surface at which it is placed, allowing for it to be placed either flat on a table or horizontally on a wall. This allows for a user to reduce the spatial clutter when designing sequences, and allows for the users to return to the mixer by position whenever they see fit. The mixer uses an array of sliders to affect the sound of the music produced through the loopers, allowing users to shape and form their track by mixing and matching different effects together. The effects can be configured during playback or between. All effects are derived from Unity's built-in Audio Mixer, and will affect all sounds except for the metronome. At the bottom of the mixer is a horizontal slider used to control the tempo (beats per minute (BPM)) of the loop, and it can additionally be adjusted either before or during playback. The BPM will affect all loops including the metronome, and will not affect a sound's pitch. The effects included in the mixer are:

- **Reverb** - adds reverberation to the audio signal, as to replicate the sound of being in a large and reflective room such as a church. The slider controls the level of the reverb, how loud the reverb will be.
- **Echo** - the audio signal gets replicated at short intervals with decaying level. The slider controls level of the echo.
- **Flanger** - duplicates the audio signal at a 20 ms interval, adding a "weirdness" to the sound.
- **EQ** - a low-pass filter, that removes the high frequencies of the sounds.

5.1.3 The Hand Menu

TappyBeatsXR utilizes a hand-tracked menu, see Figure 5.3. The menu is initially invisible, and is accessed by the user flipping their left hand upwards. The menu will only be visible and accessible if the user's hand is pointing upwards, and is hidden otherwise as to not cause additional interface clutter. From this menu, the user can spawn Sequencer Blocks and configure them using this hand menu. The "DRUMS", "BASS" and "LEAD" buttons will spawn a Sequencer Block containing its respective sound. The "MORE" and "LESS" button allows the user to configure the amount of notes in each spawned Sequencer Block, giving them either more or less sequencable notes to work with, see Figure 5.1. Lastly, the "METRONOME" button will activate a metronome. This will add a ticking sound at the chosen BPM whenever the loop is playing, giving the user an accurate feel for how fast their loop is playing,

5.2 Controlling The Experience

This section covers how the user operates and controls the TappyBeatsXR experience.

After loading TappyBeatsXR, the user is presented with a hub consisting of simple transport controls as well as a set of sub-loops, see Figure 5.4. The base setup in the application consists of a 1-bar and a 4-bar sub-loop vertically stacked, as this design allows for users to construct a short drum sub-loop whilst experimenting with the melodic elements in the longer sub-loop. To start interacting with the experience, the user has a hide-able hand-tracked



Figure 5.3: The hand-tracked menu gives users easy access to an assortment of options.

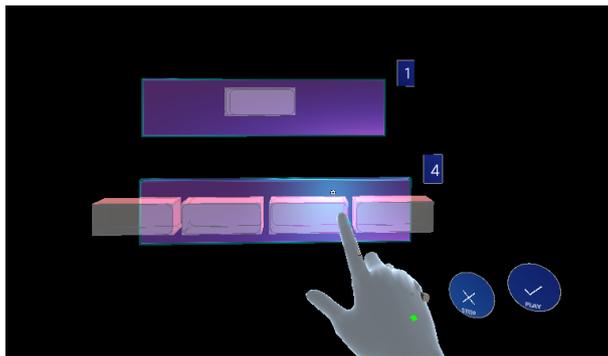


Figure 5.4: The user will be spawned in front of two sub-loops and two transport buttons.

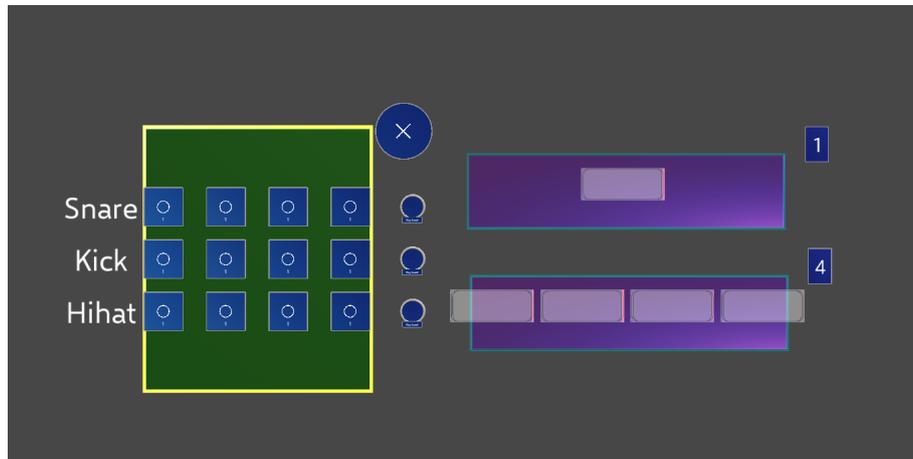


Figure 5.5: A 2D screenshot of sequencer elements. On the left: a Drum Sequencer Block. On the right: a 1-bar and a 4-bar sub-loop.

menu. This menu serves a multitude of functions. Foremost, the menu allows the user to spawn the different sets of Sequencer Blocks currently enabled in order to begin designing their loop, as well as controlling the length of the spawned Sequencer Blocks. Moreover, there are two sliders attached to the menu, allowing for the user to control the tempo of the song as well as adjusting audio effects.

The sounds in a Sequencer Block are sequenced by toggling the sounds in the order the user wishes them to play. A Sequencer Block can be re-positioned and resized by the user. In order to add your sequence to a sub-loop, the Sequencer Block is dragged and dropped into a sub-loop in the specific Sequence Bar the users wants them to be played. Any number of Sequencer Blocks can be placed in a Sequence Bar, stacking the sounds on top of each other. Every sub-loop is loaded and compiled at run-time, enabling for modifications or additions during playback.

Every sub-loop is controlled by the transport controls. This includes a "PLAY" and "STOP" button placed to the right of the Sequencer for easy playback control.

Chapter 6

Evaluation

A video demo of the application is available here:

<https://y.at/%E2%9B%B0%EF%B8%8F%F0%9F%8E%B5%F0%9F%92%BB%F0%9F%8D%BA>

6.1 User Testing

The application was tested on multiple users and feedback data was collected. The design of the questionnaire was structured around important pieces of information laid out in the book "Real World Research" by Robson and McCartan [53]. A trivial error when designing surveys is to phrase questions around the answer you want. People offering answers for a survey often tend to provide answers through a mixture of boredom, politeness and as to be seen in a good light rather than express their true feelings [53]. Thus, multiple questions were repeated and phrased both positively and negatively, as to give the respondents of the survey multiple opportunities to express their true feelings.

Due to the requirement of wireless connection to the HoloLens and necessity of being in the same room at the same time as the tester with the hardware, it was a bit difficult to do extensive testing on a wide audience, as opposed to an online application.

6.1.1 Evaluation Setup

To be able to convey the steps correctly, it was important for me to observe the testers' field of view in order for the tutorial to be efficient and to appropriately convey the necessary steps. It is impossible to observe what a user sees through the HoloLens' display from the outside, and thus the Live Preview feature from the HoloLens' Mixed Reality Capture was utilized. By logging the HoloLens onto a nearby wi-fi, a secondary device like a phone or computer can be used to access a live-stream with a 3 second delay from the user's point of view, combining the camera and HoloLens display to output a 2D video. This was used to help users with movement controls, guide users through the tutorial and testing, and to have the testers point out specific issues they had with "the application".

The testing was designed to consist of in three parts:

- First, users were given a short vocal tutorial on how to operate the HoloLens by navigating through menus and interacting with the base interface. This would start by having the users navigate to the "Calibration" section in order to calibrate the headset to the user. Users were then told to do simple tasks that required hopping between different menus, using the in-air HoloLens keyboard, as well as searching for and locating programs through the UI.
- Second, the user was told to track down and open TappyBeats. I guided the users through how the interface was set up, including how to spawn Sequencer Blocks, how to sequence and configure a loop, as well as other UI elements such as the mixer.
- Finally, the users were guided through a session of making a loop in the application. I'd show the users an example of a kind of beat, before allowing them to make their own melody at their own pace and in their own tempo.

All guidance and communication throughout the testing was done vocally. The HoloLens does not prevent users from hearing or seeing the world, allowing for in-person assistance and feedback to be quickly and effectively conveyed through voice and physical demonstration. The ability to replicate the movements that were required by the testers proved extremely valuable in the interest of time. The HoloLens has certain vital input commands that are very easily conveyed by having the tester replicate the movements. By showing users in-person how to perform certain interactions such as air-tapping, pinching, how to correctly click buttons and use the wrist-menu for both the regular interface and the application, any temporary interfacing hurdles in the testing were quickly overcome.

After completing the tour of the application, the testers were given the self-completion form (via Google Forms), and were instructed to answer honestly. I then asked questions particularly about the negative responses to the survey, as will be discussed below. Data was then collected in Google Sheets for presentation in this thesis.

6.1.2 Pilot Study

An initial pilot study was conducted on a set of biased parties consisting of 3 acquaintances. A photo of the first tester of TappyBeatsXR can be seen in Figure 6.1. Initial testing showed some interfacing issues that had not popped up under development testing. This was due to a combination of developer familiarity with the HoloLens leading to a certain set of interface elements that were not immediately usable by new users, due to them being unfamiliar with the control scheme on the HoloLens. In addition, the most striking issue was with the 'Tap to Place' functionality. This functionality allows for users to air-tap Sequencer Bars and place them on surfaces like walls and other objects mapped by the spatial observers. This feature was intended to make good use of the AR side of the application, allowing for people to play back and configure loops at different nearby locations. Unfortunately, as air tapping is among the more



Figure 6.1: A person testing out the application for the first time.

finicky actions on the HoloLens, users would accidentally air-tap the Sequencer Bars and cause the Sequencer Bars to track to their hand before adding any elements to them. For the conclusive user testing, these problems were worked on to have as little impact as possible.

6.1.3 Final Study

For the final study, **five** unbiased participants volunteered to take part in a session of user testing. The user pool consisted of 4 (PhD) students as well as 1 master student at the High Technology Center at the University of Bergen. The testing lasted for around 20 minutes per person, and was followed by a survey regarding the HoloLens and application, with most answers given on a 5-point Likert-scale. The user survey was conducted without supervision or any input other than helping the user understand a question.

Background									RESULTS
A1	I have experience with XR technology (VR headsets like Oculus or <u>Vive</u> , AR/MR headsets)	2	3	1	1	2			1.75
A2	I have tried other musical XR applications before.	No	No	No	No	No			0
A3	I have a musical background (played piano/guitar, made music with a computer program/synthesizer).	5	1	5	1	1			3
HoloLens									
B1	It is intuitive to use my hand as a controller.	4	4	4	4	4			4
B2	I had a hard time navigating the HoloLens' menus.	2	2	3	4	3			2.75
B3	The HoloLens' see-through display was disorienting (objects appeared blurry, difficult to get a sense of depth).	3	4	2	4	1			3.25
B4	Answer only if you have previous experience with XR: I prefer regular controllers (<u>Vive</u> , Oculus) over hand tracked controllers.	2	3			4			2.5
The Application									
C1	It was easy to understand the controls and navigate the experience (accessing hand menu, drag & dropping sequencer blocks etc).	3	4	4	5	2			4
C2	I understood the mechanics of the application. (<u>different</u> loops with different lengths, transport controls etc).	4	5	3	4	5			4
C3	Using the hand menu, I could find and sequence the sound I wanted.	4	4	4	4	5			4
C4	It was difficult to sequence notes using my index finger in the Sequencer Block.	4	4	1	1	1			2.5
C5	I did not find the mixer to be useful or entertaining.	1	1	1	1	2			1
C6	After initial guidance, I could figure out how to work with the application.	4	5	5	5	4			4.75
C7	The layout and positioning of the Sequencing Bars (loops) were unintuitive/obtrusive and hard to manage.	3	1	3	4	2			2.75
C8	The shaders and visuals of the application were distracting to the experience.	2	2	3	4	2			2.75
Overall Experience									
D1	I enjoyed testing out a HoloLens application.	5	5	5	5	5			5
D2	The HoloLens controls (using your fingers) was not a good fit for a music application.	1	3	3	1	3			2
D3	Having voice controls for the interactions would be helpful to me.	4	5	1	3	5			3.25

Figure 6.2: Feedback on a questionnaire completed by 5 testers on 18 statements on a 5-point Likert-scale. Response statements are on a scale from Strongly Disagree to Strongly Agree.

Evaluation results

The evaluation results are summarized in Figure 6.2.

Albeit having a limited pool of feedback, there are some interesting findings in the feedback data. The testers were generally positive about using their hand for controlling the HoloLens (B1, C1), but usually had issues with the specific interfaces (B2). This was also noted down during testing, and seems to be as a result of a combination of a depth tracking issue and personal experience. Data also shows the users generally had a good time and easily understood the mechanics of the application (C2). Interestingly, the users seem to be very split on the difficulty of sequencing notes (C4). Whether this was due to personal

preference or misunderstanding the negatively phrased question, it seems users either found it simple or quite hard.

Further feedback

During testing, vocal user feedback was written down whilst the user was undergoing first time testing of TappyBeatsXR. Moreover, in the final part of the survey, the testers had the voluntary option of elaborating on their opinions and add any personal feedback regarding the application.

In the voluntary feedback section, one tester stated that there were *"some UX issues mostly because I am left-handed and the controls were definitely less sensitive to the angle I had placed for my left hand, but worked fine with the right (eg the equalizer board)"*. This was the only left-handed user that took part in the testing. Seeing as I am right-handed and mostly used my right hand whilst testing the application, it is understandable that the left-handed experience suffered. One tester commented that *"the menus were also a bit far away from me sometimes, meaning that I had to extend my arm as far as I could in order to click something"*. Interestingly, most testers that struggled with depth perception had issues with attempting to click behind the hologram, but this tester experienced too much of a gap in the interface. Further feedback point out that the experience was interesting and enjoyable, although multiple people struggled with depth perception of the holograms.

Observations: Through a combination of user-supplied feedback and personal observations, some key findings were discovered over the course of the testing.

- **Controls** An interesting conclusion that can be drawn from the user testing is the importance of the element of practice and familiarity. During testing, every user immediately understood the HoloLens' controls and what they were *supposed* to do. However, most users had problems correctly interfacing on the headset. Interestingly, the problems that were most prevalent were both present whilst navigating the standard HoloLens menus as well as the application itself.
- **User practice** A surprising but in hindsight obvious part of the testing of TappyBeatsXR was the importance of user practice, both in terms of familiarity with the headset as well as the application itself. Although the layout, behavior of holograms and input methods were rigorously iterated on before testing, the element of user practice was something that had an enormous impact on the experience of the testers. As previously stated, users quickly picked up on how they were intended to navigate and interface on the HoloLens, but the process of actually performing operations proved more difficult than expected. A positive to this was however that the testers would usually get a grasp of it after a few minutes in the headset, although unfortunately this usually happened near the end of the testing session. A few more runs in the application would definitely had resulted in a smoother and better experience for the user. Different testers also adapted on a differing scale to this new environment and user interface.
- **Ease of communication** As proposed, the see-through display on the HoloLens was vital in teaching users and guiding them through the differ-

ent sections of the tests. With regular controllers and hardware, there will usually be a set of buttons that whose layout can be simply memorized. However, every step of the testing required some combination of spatially locating elements and operating them in a specific manner. Solely relying on auditory feedback to and from the user would have made this task significantly slower, having to explain not only the position, but the specific arm movements required to operate the element. However, being physically present in the room with the tester, it was trivial to help them locate any object of interest in 3D and give them feedback on any incorrect hand movements. Together with the Live Preview function, it was as simple as showing them a position in the room and presenting a specific hand movement, and the user would understand the assignment.

- **Stream-induced lag** As mentioned in the Setup section, the HoloLens' "Live Preview" was utilized in order to make testing as efficient and problem-free as possible. This clearly highlighted the benefit of having the HoloLens be see-through. However, due to the headset having to dedicate a sizable portion of its CPU to live-stream the headset feed, this induced unexpected lag during testing. Holograms would occasionally behave irrationally, and there were timing issues with the loop playback. This will either have to be iterated on in future versions of the application, or rely more on the Holographic Remoting discussed in 4.3.4.
- **The effect of light** The mixer element relies on the spatial mapping features of the HoloLens to properly function, by enabling users to place the mixer on any trackable surface in the room. This is also an important element to bridge the gap between interacting with virtual objects and the user's physical surroundings, a core element of the MR experience. However, something that had a substantial impact on the quality of the tracking was the light intensity of the room in which testing took place. Both low- and intense-light scenarios have a negative impact on the headset's mapping capabilities. When testing occurred during development, the spatial mapping would behave as intended, being able to map out the area as the application ran. Nevertheless, albeit the user testing occurring in a standardly lit conference room, the spatial mapping seems to have substantial problems correctly mapping out the room. The official documentation suggests 500-1000 lux [37] for optimal use, but a light intensity meter was not a tool available for use at the time of testing.

Chapter 7

Discussion

This chapter covers different potential use cases for TappyBeatsXR, as well as reflections on the choices made during development. The chapter also elaborates on the limitations of both the current iteration on the software and the technology chosen for developing TappyBeatsXR.

7.1 Use Cases

During the development of TappyBeats XR, some specific use cases were in mind.

A possible area of use for TappyBeatsXR is within educational facilities, specifically within primary and middle schools. The intuitiveness of the interface can be used as a method to educate children and teenagers about musical concepts like melodies, rhythm patterns and chord progressions. Initial discussion regarding these ideas were had with Karsten Spetch, a professor from the Faculty of Psychology at the University in Bergen. Despite the absence of an online cooperative mode, the HoloLens has support for live-streaming footage from the headset real-time over Wifi to a nearby computer screen. This enables the teacher to give feedback both through seeing the student's visual interface and through the streamed audio. Furthermore, it is trivial to communicate with your direct surroundings in the headset, as the non-obscuring visor and open ear design allows for full eye contact and communication. Stosic talks about the importance of incorporating educational technology, stating that it is crucial to take into account the disproportionate distribution of knowledge and skill in a classroom setting, as kids at a lower level of knowledge and skill aren't always able to contribute, and get left behind [57]. TappyBeatsXR presents an uncomplicated way of manipulating and playing with basic musical elements through an intuitive interface that doesn't require any high-level knowledge of music theory or previous rigorous practice, allowing for people of all ages and sizes to express themselves musically through an interesting device.

Throughout the evolution of this project, keeping the interface uncomplicated was a major design concept. Accessibility has been in the forefront of this application from it's adolescence, and the final interface is a response to this.

The simplicity of the interface bears multiple fruits: it is easy to understand and get into, and as will be discussed in 8.2, this allows for the access to an experience of music that might be unachievable for a certain crowd through other mediums. Integrating a fully speech controlled version of the application with the current design is very achievable, and this might be helpful for users with reduced motor functions.

The user friendly design of both the application as well as the project setup itself also allows for TappyBeatsXR to have potential future use cases that were not thought of during development.

7.2 Limitations and Lessons Learned

Tracking Accuracy and Visual Constraints

When one considers the possibilities of a music-creation extended reality software, being able to physically control instruments seems like one of the core ideas. Such as having a personal kinetic impact on a set of drums or keys with the use of hand controllers, having the sound alter and distort depending on intended personal input. However, albeit the hand tracking on the HoloLens being remarkable at doing transitional and navigational tasks, tracking the speed and minute details of a user's hand movements is simply something it is not capable of. Consequently, this caused the interface to be boiled down to a simple "click-and-drag" interface. This causes difficulty of immersion in regards to "feeling" the music, as one has to statically input sequences and play them back in a two step process. Being able to live record parts of the loop would be a huge improvement, but did not seem feasible during deployment of the application. This was due to two factors: people would have to be quite familiar with the controls and how to accurately operate them, as well as the aforementioned temporal detail loss on the hand tracking.

Debugging on the HoloLens

A major problem during the development of this application was testing using Microsoft's dedicated tools, as well as deploying the application for appropriate testing. The preferred program for testing a HoloLens application on your computer is the **HoloLens Emulator**, developed by Microsoft. The emulator allows you to test the entire HoloLens experience, from start screen to application, simulating inputs from the HoloLens sensors using either your keyboard and mouse, or an Xbox controller. The emulator also sports features like allowing you to easily choose between different gestures and actions, as well as allowing you to log into your Microsoft account or change the network configuration. This emulator also allows for simulated input pipelines, scripted with C# scripts. But unfortunately, this emulator uses a "Hyper-V virtual machine", which is not available for the most common Windows distribution, Windows Home. This feature is locked behind Windows Pro or above, which was not available for the computer used throughout the development of TappyBeatsXR.

Furthermore, as briefly mentioned in 4.4.1, most of the debugging tools for the HoloLens are also readily available over USB, removing the need to deal with network configuration when setting up the HoloLens. For the single HoloLens

that was used during the entire development, any USB connectivity except for charging proved impossible to get to appropriately function. After multiple attempts on multiple computers and consultation with my supervisor, relying on USB deployment for the rest of the project seemed unachievable. As the start of this project was developed on university ground, achieving connection between the development PC and the HoloLens also proved unfeasible due to the strict firewall settings. The only reliable solution ended up being developing the application at my apartment, using my own WiFi for HoloLens debugging and deployment.

Performance

To combat performance issues, a pack of shaders optimized specifically for the HoloLens were also integrated on every single object [38]. Furthermore, due to its large reliance on scripts to control the looping aspect of the application, early test builds of TappyBeatsXR showed severe performance issues. Thusly, rewrites of both the base sequencer engine as well as any additional hook code affecting the time precision of the music had to be performed. It was important to make sure the code was highly optimized in order to reduce latency between loops.

The Mixed Reality Toolkit

The Mixed Reality Toolkit is very much a work-in-progress, and certain parts of the core functionality are either entirely undocumented or have an implementation that does not correspond with the official documentation. A suggestion for future developers would be to stay up-to-date on new releases of the documentation, and to seek out forums or chat applications (such as the HoloLens Slack server).

The Novelty of Mixed Reality

Another important lesson learned in developing for mixed reality is to start testing at an early stage. As mentioned in Section 6.1.3, one of the main findings over the course of the user testing was the importance of being familiar with the technology. Throughout development, I would design the interface on a 2D screen and quickly test it solely on myself. This over time gave me experience and a sense of familiarity with what the HoloLens *expected* me to do, and the interface was designed around this. When introducing TappyBeatsXR to completely new users, they would behave differently to one another, using the tapping and grabbing functionality to different degrees of success. As the intention of TappyBeatsXR is to appeal to newer users, an idea for future developers seeking the same goal would be to periodically test on fresh users and note down how they interact with your interface.

Suggestions for the Sequencer Engine

In the final application, all Sequential Sequencers are driven in parallel, and each Sequencer Bar is queued up for playback by the preceding Sequencer Bar. This was a purposeful design choice, as it allows for free expansion of any loop, whether in number of parallel loops or the length of a single one. However, although compensated for, this gives the room for error in terms of timing between the different Sequential Sequencers, and can sometimes be affected by performance. An alternative approach would be to expand the sequencer

engine and compile all individual Sequencer Bars into one common Sequential Sequencer. This would remove the ability to create syncopated sequences, but be a more sturdy approach for simpler straight-forward loops.

7.3 Extensability

The majority of code written and element compositions in the Unity editor was created to enable as many additions and configurations as possible. Multiple such ideas have been touched on in the thesis, such as adding sound packs that are changeable during run-time and adding more interesting spatial interaction visuals with the use of spatial mapping. The entire code base and engine created for the application is available **open source** on GitHub for anyone to continue furthering the work done by me during this master's thesis. Seeing as the idea and execution of the development was quite novel in terms of similar applications being created with modern MR technology, it was not realistic to be able to achieve all the ideas that constituted the initial scope. Instead, a concentrated effort was made in order to make the code as trivial to work with in the future as possible. This included multiple considerations. One of the most important ones are proper naming schemes for methods and classes, allowing for future developers to understand the intended function behind a method. Secondly, a focus on keeping scripts modular and easy to manage by utilizing events and delegates. This allows for additions to snippets of code to not break links between objects, as they are managed separately (single responsibility). Thirdly, a focus on keeping a proper file structure in order to easily obtain the required elements for creating and expanding the play-space using the Unity editor.

Chapter 8

Conclusion and Future Work

This chapter summarizes the contributions of the thesis along with some ideas for future iterations of the software.

8.1 Summary

This thesis provides an overview of research done in the field of extended and mixed realities, and its impact on different fields of commerce and science. The thesis focuses on the marriage between music and technology, and how the two concepts have long been interlinked. The thesis presents the work done by others in applying music to XR and the thesis also presents the design choices that were made during development, as well as giving an overview of how the final application was constructed. Furthermore, the thesis explores a lot of the difficulties and challenges developing an application for a newly released peripheral with a small development community.

The application developed for this thesis - TappyBeatsXR - provides an immersive musical experience designed for non-musicians with the help of the mixed reality headset HoloLens 2 and the real-time development platform Unity. This application serves as a novel attempt at presenting a possible solution to creating an environment in which non-musicians can enjoy and learn the concepts of music creation. The code and project files as well as any prefabs produced for the application, are openly hosted on Github under the MIT license for any other researchers or programmers to further development.

The Github repository can be found here: <https://github.com/Zylvian/TappyBeatsXR>

8.2 Future Work

This section covers some potential ideas for future developers of TappyBeatsXR.

8.2.1 Cooperative Mode

As discussed previously in the thesis, MR technology has a large focus on remote collaboration, allowing groups of people huge distances apart to work together to wirelessly solve issues in a familiar environment. When developing the application, this was always something that was kept in mind. Unfortunately, by being limited to only one available HoloLens, this was not something that was feasible to implement within the scope of this thesis. I however strongly encourage future work to focus on using the available tools such as Azure Spatial Anchors and Photon SDK [50] to expand the experience to a multitude of users.

8.2.2 Dynamic Instruments

Any samples can be changed at run-time, but it would be interesting to use C# to create sounds that can be sequenced. One approach is to implement a version of the the sound sculpting proposed by Mulder, Fels, and Mase [44] as described in Section 3.2.2, although this would require arduous development. Another approach would be to use a selection of tools from the MRTK together with a audio synthesis engine designed in C, by compiling it to a Unity plugin with the help of, e.g the functional programming language Faust. [17]. Alternatively, an idea would be to implement OSC-XR in conjunction with Chunity in order to dynamically create sounds, and save the created state to be used as a sample in sequencing of the loop. OSC-XR's own implementation of their tools could very well serve as an inspiration for future versions of TappyBeatsXR.

8.2.3 Generating Content for Other Application

A potential idea is to use TappyBeatsXR to construct and generate content for other applications or games, as the array-based layout of the final result can easily be extracted and saved. One hypothetical implementation of this would be to generate so called "beat maps", which are the names given to the levels for the VR game Beat Saber [18] discussed earlier in the thesis. By exporting the produced loop and assigning a unique position to each different sound in the game, one could convert a loop made in TappyBeatsXR into a map playable in Beat Saber with the click of a button and run of a script.

Bibliography

- [1] Elin Abdul Rahim et al. “A desktop virtual reality application for chemical and process engineering education.” In: *Proceedings of the 24th Australian Computer-Human Interaction Conference*. 2012, pp. 1–8.
- [2] Ableton. *Ableton Live*. URL: <https://www.ableton.com/en/shop/live/>. (accessed: 31.08.2022).
- [3] Victor Agulhon. *What is spatial computing?* URL: <https://medium.com/@victoragulhon/what-is-spatial-computing-777fae84a499>. (accessed: 22.06.2022).
- [4] Christopher Andrews et al. “Extended reality in medical practice.” In: *Current treatment options in cardiovascular medicine* 21.4 (2019), pp. 1–12.
- [5] Jack Atherton and Ge Wang. “Doing vs. Being: A philosophy of design for artful VR.” In: *Journal of New Music Research* 49.1 (2020), pp. 35–59.
- [6] Nathalie Bressa et al. “What’s the Situation with Situated Visualization? A Survey and Perspectives on Situatedness.” In: *IEEE Transactions on Visualization and Computer Graphics* 28.1 (2021), pp. 107–117.
- [7] Zach Buckley and Kristin Carlson. “Towards a framework for composition design for music-led virtual reality experiences.” In: *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*. IEEE. 2019, pp. 1497–1499.
- [8] Marko Ciciliani. “Virtual 3D environments as composition and performance spaces.” In: *Journal of New Music Research* 49.1 (2020), pp. 104–113.
- [9] Beth Coleman. “Using sensor inputs to affect virtual and real environments.” In: *IEEE Pervasive Computing* 8.3 (2009), pp. 16–23.
- [10] Tim Coles and Michael Hall. “The geography of tourism is dead. Long live geographies of tourism and mobility.” In: *Current Issues in Tourism* 9.4-5 (2006), pp. 289–292.
- [11] Arzu Çöltekin et al. “Extended Reality in Spatial Sciences: A Review of Research Challenges and Future Directions.” In: *ISPRS International Journal of Geo-Information* 9.7 (2020). ISSN: 2220-9964. DOI: 10.3390/ijgi9070439. URL: <https://www.mdpi.com/2220-9964/9/7/439>.
- [12] TEAM COUNTERPOINT. *Global XR (AR & VR Headsets) Shipments Market Share: By Quarter*. URL: <https://docs.microsoft.com/en-us/windows/mixed-reality/mrta-unity/mrta2/?view=mrtaunity-2022-05>. (accessed: 12.08.2022).
- [13] Tech Crunch. *With new realities to build, Unity positioned to become tech giant*. URL: <https://techcrunch.com/2017/05/25/with-new->

- realities-to-build-unity-positioned-to-become-tech-giant/. (accessed: 04.05.2022).
- [14] Marc Doty. *What's Behind The Resurgence of Analog Synthesizers?* URL: <https://performermag.com/best-instruments/best-music-keyboards-synth/resurgence-of-analog-synthesizers/>. (accessed: 20.04.2022).
- [15] DPVR. *DPVR*. URL: <https://www.dpvr.com/en/>. (accessed: 30.08.2022).
- [16] Daniel Escudero and Martin Wettig. *Designing holograms - mixed reality*. Mar. 2022. URL: <https://docs.microsoft.com/en-us/windows/mixed-reality/discover/designing-holograms>.
- [17] Dominique Fober, Stéphane Letz, et al. *FAUST: an efficient functional approach to DSP programming*. 2009.
- [18] Beat Games. *Beat Saber*. URL: <https://beatsaber.com/>. (accessed: 05.04.2022).
- [19] Stefan Gössling, Daniel Scott, and C Michael Hall. "Pandemics, tourism and global change: a rapid assessment of COVID-19." In: *Journal of sustainable tourism* 29.1 (2020), pp. 1–20.
- [20] The Guardian. *The power of the Pentatonic Scale*. URL: <https://www.theguardian.com/science/punctuated-equilibrium/2010/dec/17/1>. (accessed: 22.06.2022).
- [21] Brian Heater. *Microsoft and Trimble made a hard hat with HoloLens built-in*. URL: <https://techcrunch.com/2019/02/24/microsoft-and-trimble-made-a-hardhat-with-hololens-built-in/?guccounter=1>. (accessed: 20.04.2022).
- [22] Image-Line. *Fruity Loops*. URL: <https://www.image-line.com/>. (accessed: 31.08.2022).
- [23] Magic Leap Inc. *Magic Leap 2*. URL: <https://www.magicleap.com/en-us/>. (accessed: 05.07.2022).
- [24] David Johnson, Daniela Damian, and George Tzanetakis. "Osc-xr: A toolkit for extended reality immersive music interfaces." In: *Proc. Sound Music Comput. Conf.* 2019, pp. 202–209.
- [25] Sam Kavanagh et al. "A systematic review of virtual reality in education." In: *Themes in Science and Technology Education* 10.2 (2017), pp. 85–119.
- [26] Andrei OJ Kwok and Sharon GM Koh. "COVID-19 and extended reality (XR)." In: *Current Issues in Tourism* 24.14 (2021), pp. 1935–1940.
- [27] Ben Lang. *Hands-on: Magic Leap 2 Shows Clear Improvements, But HoloLens 2 Retains Some Advantages*. URL: <https://www.roadtovr.com/magic-leap-2-hands-on-vs-hololens-2-impressions/>. (accessed: 05.07.2022).
- [28] Magic Leap. *Magic Leap Presskit*. URL: <https://magic-leap.presskite.com/>. (accessed: 23.08.2022).
- [29] Sandra Maria Correia Loureiro, Joao Guerreiro, and Faizan Ali. "20 years of research on virtual reality and augmented reality in tourism context: A text-mining approach." In: *Tourism management* 77 (2020), p. 104028.
- [30] Ludomancer. *Unity Audio Sequencer*. URL: <https://github.com/Ludomancer/Unity-Audio-Sequencer>. (accessed: 19.04.2022).
- [31] Alan MacEachren et al. "Virtual Environments for Geographic Visualization: Potential and Challenges." In: Jan. 1999, pp. 35–40. DOI: 10.1145/331770.331781.
- [32] Stanford Medicine. *The Stanford Virtual Heart – Revolutionizing Education on Congenital Heart Defects*. URL: <https://www.stanfordchildrens>.

- org/en/innovation/virtual-reality/stanford-virtual-heart. (accessed: 23.08.2022).
- [33] Meta. *Meta Quest 2*. URL: https://store.facebook.com/gb/en/quest/products/quest-2/?utm_source=www.oculus.com&utm_medium=oculusredirect. (accessed: 30.08.2022).
- [34] Microsoft. *Choosing an engine [for HoloLens development]*. URL: <https://docs.microsoft.com/en-us/windows/mixed-reality/develop/choosing-an-engine?tabs=unity>. (accessed: 08.07.2022).
- [35] Microsoft. *Getting around HoloLens 2*. URL: <https://docs.microsoft.com/en-us/hololens/hololens2-basic-usage>. (accessed: 21.04.2022).
- [36] Microsoft. *Holographic Remoting*. URL: <https://docs.microsoft.com/en-us/windows/mixed-reality/develop/native/holographic-remoting-overview>. (accessed: 05.07.2022).
- [37] Microsoft. *Hololens environment considerations*. URL: <https://docs.microsoft.com/en-us/hololens/hololens-environment-considerations>. (accessed: 05.07.2022).
- [38] Microsoft. *MRTK Shaders for Unity*. URL: <https://docs.microsoft.com/en-us/windows/mixed-reality/mrtk-unity/mrtk2/features/rendering/mrtk-standard-shader?view=mrtkunity-2022-05>. (accessed: 08.07.2022).
- [39] Microsoft. *Spatial mapping*. URL: <https://docs.microsoft.com/en-us/windows/mixed-reality/design/spatial-mapping>. (accessed: 05.07.2022).
- [40] Microsoft. *What is a hologram?* URL: <https://docs.microsoft.com/en-us/windows/mixed-reality/discover/hologram>. (accessed: 19.04.2022).
- [41] Microsoft. *What is the Mixed Reality Toolkit?* URL: <https://docs.microsoft.com/en-us/windows/mixed-reality/mrtk-unity/mrtk2/?view=mrtkunity-2022-05>. (accessed: 12.08.2022).
- [42] Tassos A Mikropoulos et al. “Students’ attitudes towards educational virtual environments.” In: *Education and Information Technologies* 3.2 (1998), pp. 137–148.
- [43] Paul Milgram and Fumio Kishino. “A taxonomy of mixed reality visual displays.” In: *IEICE TRANSACTIONS on Information and Systems* 77.12 (1994), pp. 1321–1329.
- [44] Axel GE Mulder, Sidney S Fels, and Kenji Mase. “Design of virtual 3D instruments for musical interaction.” In: *Graphics Interface*. Vol. 99. Cite-seer. 1999, pp. 76–83.
- [45] Niantic. *Pokémon Go*. URL: <https://pokemongolive.com/en/>. (accessed: 05.04.2022).
- [46] OpenXR. *OpenXR - Unifying Reality*. URL: <https://www.khronos.org/openxr/>. (accessed: 11.07.2022).
- [47] Veronica S Pantelidis. “Reasons to use virtual reality in education and training courses and a model to determine when to use virtual reality.” In: *Themes in Science and Technology Education* 2.1-2 (2010), pp. 59–70.
- [48] Joseph A Paradiso and James A Landay. “Guest editors’ introduction: Cross-reality environments.” In: *IEEE Pervasive Computing* 8.3 (2009), pp. 14–15.
- [49] philc_nity. *Flickering issue affecting HoloLens 2 projects in 2020.3.21 and later*. URL: <https://forum.unity.com/threads/flickering-issue->

- affecting-hololens-2-projects-in-2020-3-21-and-later-1200199/. (accessed: 20.04.2022).
- [50] Photon. *Photon SDK*. URL: <https://doc.photonengine.com/en-us/arvr/current/hololens/hololens-intro>. (accessed: 15.08.2022).
- [51] polar-kev. *MRTK Examples Hub - Mixed Reality Toolkit*. en-us. URL: <https://docs.microsoft.com/en-us/windows/mixed-reality/mrtk-unity/features/example-scenes/example-hub> (visited on Apr. 28, 2022).
- [52] Johnny Reid. URL: <https://www.mixyourreality.com//insights/mixed-reality-headsets-available-in-the-us>. (accessed: 04.05.2022).
- [53] Colin Robson and Kieran McCartan. *Real world research: a resource for users of social research methods in applied settings*. Wiley, 2016.
- [54] Thomas E Rudolph. *Teaching music with technology*. GIA Publications, 2004.
- [55] A. Zita Sampaio and Luís Viana. “Virtual Reality used as a learning technology: Visual simulation of the construction of a bridge deck.” In: *2013 8th Iberian Conference on Information Systems and Technologies (CISTI)*. 2013, pp. 1–5.
- [56] Stefania Serafin et al. “Virtual reality musical instruments: State of the art, design principles, and future directions.” In: *Computer Music Journal* 40.3 (2016), pp. 22–40.
- [57] Lazar Stosic. “The importance of educational technology in teaching.” In: *International Journal of Cognitive Research in Science, Engineering and Education (IJCRSEE)* 3 (June 2015), pp. 111–114. DOI: 10.23947/2334-8496-2015-3-1-111-114.
- [58] Ancret Szpak, Stefan Carlo Michalski, Tobias Loetscher, et al. “Exergaming with beat saber: an Investigation of virtual reality aftereffects.” In: *Journal of Medical Internet Research* 22.10 (2020), e19840.
- [59] Varjo Technologies. *Varjo Technologies*. URL: <https://store.varjo.com/>. (accessed: 23.08.2022).
- [60] Varjo Technologies. *Varjo XR-3*. URL: <https://varjo.com/products/xr-3/>. (accessed: 04.05.2022).
- [61] Luca Turchet, Rob Hamilton, and Anil Çamci. “Music in extended realities.” In: *IEEE Access* 9 (2021), pp. 15810–15832.
- [62] Unity. *Unity Metronome*. URL: <https://docs.unity3d.com/ScriptReference/AudioSettings-dspTime.html>. (accessed: 19.04.2022).
- [63] Unity. *Unity Real-Time Development Platform*. URL: <https://unity.com/>. (accessed: 11.07.2022).
- [64] Case Western Reserve University. *HoloAnatomy app wins top honors*. URL: <https://engineering.case.edu/HoloAnatomy-honors>. (accessed: 23.08.2022).
- [65] UploadVR. *HoloLens 2 AR Headset: On Stage Live Demonstration*. URL: <https://www.youtube.com/watch?v=uIHPPtPBgHk>. (accessed: 30.06.2022).
- [66] Tran Van Khe. “Is the pentatonic universal? A few reflections on pentatonicism.” In: *The world of music* 19.1/2 (1977), pp. 76–84.
- [67] Daniel A Walzer. “Independent music production: how individuality, technology and creative entrepreneurship influence contemporary music industry practices.” In: *Creative Industries Journal* 10.1 (2017), pp. 21–39.
- [68] Wang Wei, Li Dongsheng, and Liu Chun. “Fixed-wing aircraft interactive flight simulation and training system based on XNA.” In: *2013 Inter-*

- national Conference on Virtual Reality and Visualization*. IEEE. 2013, pp. 191–198.
- [69] Sean White and Steven Feiner. “SiteLens: situated visualization techniques for urban site visits.” In: *Proceedings of the SIGCHI conference on human factors in computing systems*. 2009, pp. 1117–1120.
- [70] Jackie Wiggins. “Teaching music with a social constructivist vision of learning.” In: *Teaching general music: Approaches, issues, and viewpoints* (2016), pp. 49–72.