

Phrase 3 - Final Report

Mixed Reality – MR DrumKit

Final Year Project 2017
Department of Computer Science
The University of Hong Kong

Wong Ho Sum, Crystal UID: 3035180108

Supervisor
Dr. Yi-King Choi
Research Assistant Professor of Computer Science

13th Apr. 2018

Abstract

Drum kits are large and noisy instruments that are not very accessible to many people. This hinders budding drummers from starting and obstructs regular players from practising. However, with the help of mixed reality technologies, realistic virtual drumming experience can be provided. This paper describes the background, design, implementations and results of MR DrumKit, a mixed reality HoloLens application prototype that utilises Kinect to simulate drumming. The product is largely different from its initial design. Changes are made during implementation due to practicability and technical issues.

Acknowledgment

I would like to thank my supervisor, Dr Y K Choi for her guidance and suggestions in this project, and Dr K L Ho for teaching me how to write academic reports.

Also, I would like to thank my family for supporting and bearing with me when I was under stress due to this FYP. Especially when I have to use the only desktop at home all day for development. Special thanks to my siblings for being testers.

Abbreviations

MR	Mixed reality
VR	Virtual reality

Table of Contents

Abstract	2
Acknowledgment	3
Abbreviations	4
0.1 List of Tables	7
0.2 List of Figures	8
1. Introduction	9
1.1 Background.....	9
1.2 Related Works	9
1.2.1 Aerodrums	9
1.2.2 DrumKit VR	10
1.3 Merging MR technology with Drum Kits	10
1.4 Outline of this paper.....	10
2. Objective	11
2.1 Scope	11
3. Approach.....	12
3.1 HoloLens	12
3.1.1 HoloLens Hardware.....	12
3.1.2 Justifications for choosing the HoloLens.....	13
3.2 Kinect.....	14
3.2.1 Kinect Hardware -- Camera.....	14
3.2.2 Kinect Middleware	15
3.2.3 Justifications for choosing the Kinect.....	15
3.3 Unity	16
3.4 Development Environment.....	17
4. Design	18
4.1 Set up of the MR DrumKit.....	18
4.2 Using the MR DrumKit	19
4.2.1 Standard Air Drums mode	19
4.2.2 Custom Drum Kit mode	19
4.2.3 Application Customisation.....	20
4.2.4 Extra Features for Future.....	20
4.3 Limitations of Hardware	21
4.3.1 Kinect.....	21
4.3.2 HoloLens.....	21

4.3.3 User Environment	21
5. Previous works/Final Status.....	22
5.1 Background Research on Related Technologies.....	22
5.1.1 Motion Capturing Technology	22
5.2 Development Environment Setup.....	23
5.2.1 Initial Setup (before 22 nd October 2017)	23
5.2.2 Second Setup (after 22 nd October 2017)	23
5.3 Logic Design	24
5.3.1 Game Set up.....	24
5.3.2 HoloLens – Kinect Communication	24
5.3.3 Drum hit Detection.....	26
5.3.4 Game Mode	27
6. Result Discussion.....	33
6.1 Development Process	33
6.2 Testing	33
6.3 Use of Kinect.....	34
6.4 Use of HoloLens	34
6.5 Use of HoloLens—Kinect Project	35
6.6 Use of Sender-Receiver-Server System	35
6.7 User position mapping.....	36
6.8 Drum Hit Detection.....	36
6.9 Drum Placement	37
6.10 Easter Egg Mode	37
6.11 General Gameplay	37
6.12 Discussion Conclusion	38
7. Conclusion.....	39
8. References	40
9. Appendices	43
9.1 Original Project Schedule in Project Plan.....	43
9.2 Revised Schedule in Interim Report.....	44
9.3 Development Machine Specifications.....	45
9.4 Hi-hat Mechanism	47

0.1 List of Tables

Table 1: Platforms/Tools used in this project	17
Table 2: Revised Schedule.....	44

0.2 List of Figures

Figure 1: Aerodrums.....	9
Figure 2: Drumkit VR.....	10
Figure 3: Microsoft HoloLens.....	12
Figure 4: Cameras and sensors on Hololens	12
Figure 5: Microsoft Kinect.....	14
Figure 6: Data flow inside Kinect	15
Figure 7: Set up of Application.....	18
Figure 8: Display of Sender	25
Figure 9: Display of Server.....	26
Figure 10: User's palms are in red.....	27
Figure 11: Environment Mesh	28
Figure 12: 3 cubes floating in front of user.....	28
Figure 13: A placed drum	29
Figure 14: User playing the drums	29
Figure 15: Mesh made invisible.....	30
Figure 16: Ball shooting out of Right hand	30
Figure 17: Drum components hit by the balls.....	31
Figure 18: Drum components floating around.....	31
Figure 19: Drum stopped by walls and floor	31
Figure 20: Crash stopped by ceiling.....	32
Figure 21: Kinect Studio.....	34
Figure 22: Hi-hat	47

1. Introduction

1.1 Background

Budding drummers usually face a problem: they wanted to play the drums, but drum sets are not easily available. Drum sets are relatively large instruments, and many people, especially in Hong Kong, cannot afford the space to own a drum set. While people can also go to studios, this option is expensive and inconvenient. To make the instrument more accessible, mixed reality technologies can be used to let drummers play drums regardless of location and availability of physical drum sets.

1.2 Related Works

1.2.1 Aerodrums

Similar ideas have been explored in the field. The commercial air-drumming product Aerodrums [1] utilises high-speed motion capture to let users play drums on the air (see Figure 1(1)). However, this product offers visual feedback only on a computer screen (see Figure 1(2)). It can be difficult to have an immersive experience of drumming. It is also relatively difficult to grasp the location of different drums in relative to the drumsticks, especially for beginners [2].



Figure 1: (1) Users hit the air to hit the invisible drums. (2) Users can see the position of the sticks relative to the drums on the computer screen.

Taken from [3] <https://www.pcmarket.com.hk/>

1.2.2 DrumKit VR

The virtual reality (“VR”) drum kit DrumKit VR [4] offers a visible virtual drumming experience to players (see Figure 2(1)). Players use HTC Vive’s tracked motion controllers as sticks to play with the kit. However, players cannot see their hands and feet (see Figure 2(2)). It is not possible to see if the gestures of the hands are correct. In addition, players cannot use their feet to control the drums like a real drum kit because feet motion is not captured in Vive. There is still some distance between this software and a real drum kit.



Figure 2: (1) A virtual drum set that players can see. (2) Players cannot see their hands.
Taken from [5] <https://www.youtube.com/>

1.3 Merging MR technology with Drum Kits

By using mixed reality (“MR”) technology, a closer simulation of a real drum kit can be achieved. Mixed reality projects virtual projections onto the user’s real environment. Users will be able to view the real and virtual worlds at the same time. Users can interact with the virtual world with hands and feet by motion-capturing peripherals. Merging drums with MR can provide users a more accessible and realistic air-drumming experience.

1.4 Outline of this paper

This paper reports the progress of this MR drums project. First, it introduces MR DrumKit, a mixed reality application for HoloLens. Then, the details of implementation are introduced. To justify the engineering choices, background information of the related software/hardware are provided. Next, the final status and design of progress is reported, and results are discussed. Finally, this paper closes with a conclusion and projections for the future.

2. Objective

This project aims to develop an MR drum set application that allows users to play the drums without needing a real drum set. MR DrumKit aims to provide a feasible, accessible and realistic drumming alternative to users.

2.1 Scope

The intermediate goal of the project is to develop a prototype application. This prototype includes 2 drumming modes: the standard air drums mode and custom drum kit mode.

In the standard air drums mode, a drum set hologram is projected in front of the user. The user hit the drums with sticks to produce corresponding sounds. Users can also step on the floor to hit the pedals.

In the custom drum kit mode, users can choose different flat surfaces as drum pads. Users hit the selected surfaces to produce corresponding sounds.

The final goal is to develop a more advanced version based on the prototype. More customisation is introduced, such as allowing users to customise the position of different drums in the standard air drums mode or having more choices of drum kits. Music can also be played during the session. A short tutorial will also be included to let beginners learn the basics of drumming.

As the focus of this application is on practicability, this application does not include beautiful graphics. In addition, due to technical constraints of peripherals, this application does not provide very precise differentiation on the intensity of forces used to hit the drums.

3. Approach

3.1 HoloLens

3.1.1 HoloLens Hardware

MR DrumKit is a HoloLens application. The Microsoft HoloLens (“HoloLens”) (Figure 3) is a standalone mixed reality headset. HoloLens allows users to see 3D holograms projected over the real environment. Sensor units comprised of depth cameras and light sensors are positioned on the two sides of the headset to sense the environment and the user’s movements (see Figure 4). Accelerometers, gyroscopes and magnetometers are used to detect the movement of the head. 3D images are rendered differently according to the location of the head to the environment, which gives the user a sense that the image is a real 3D object. The HoloLens runs on Windows 10. [6] [7] [8]



Figure 3: The Microsoft HoloLens
Taken from [9] <https://www.microsoft.com/>

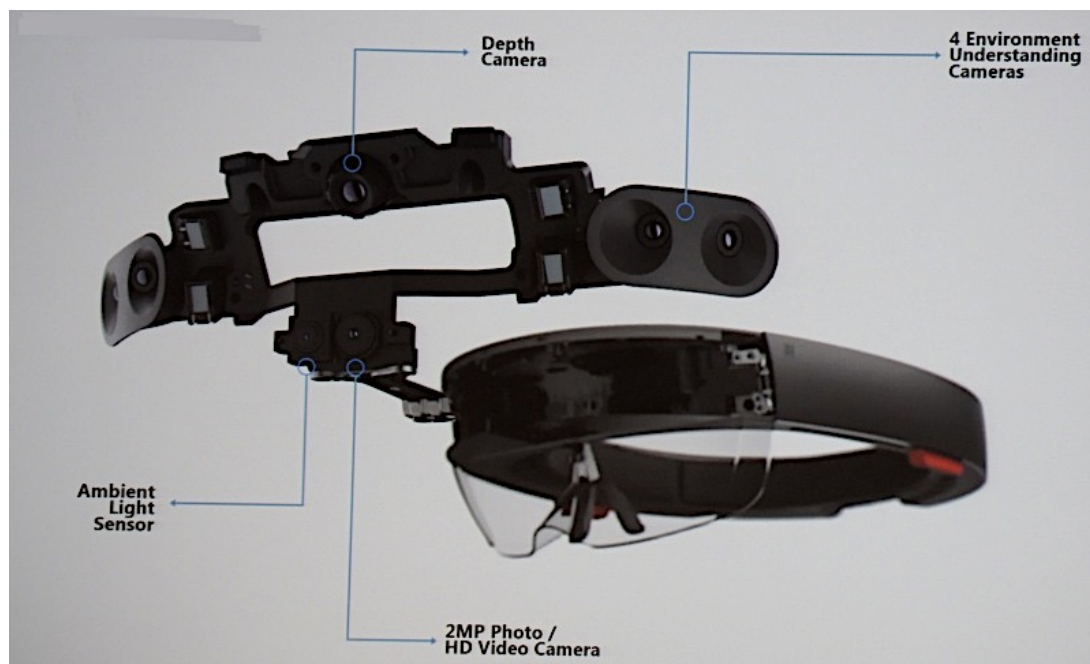


Figure 4: The position of cameras and sensors on HoloLens
Taken from [10] <http://image-sensors-world.blogspot.hk/>

3.1.2 Justifications for choosing the Hololens

Hololens is used for this project because it is the only MR equipment that does not require any tethering to a computer. Users might have body and head movements when playing the drums. They can feel freer to move along with the rhythm.

Hololens' scanning capability is also utilised to detect the user's environment. The Hololens scans for the floor and flat surfaces. The Hololens scans for the floor to project the drum set on the floor in the standard air drums mode. The Hololens also scans for flat surfaces for the user to choose as different drums in the custom drum kit mode.

Hololens' scanning capability is also used to detect user gesture input. Users can set some options using hand gestures, such as setting the volume and switching between drum sets.

3.2 Kinect

Kinect (Figure 5) is used for capturing user's hand and feet movements. The technology Structured Light is used to compute a depth map, and machine learning is used to analyse body position [11].



*Figure 5: The Microsoft Kinect
Taken from [12] <https://www.xbox.com/>*

3.2.1 Kinect Hardware -- Camera

Unlike older software programs that uses difference in colour and texture to distinguish objects from the background, the Kinect uses the Time of flight technology [13]. The Kinect transmits near-infrared light and measures the time it needs to travel back after reflecting off objects. This technology can eliminate some false positives because it solves the problem of ambient light. The Kinect can distinguish depth of objects within 1 cm and width and height within 3mm [13].

3.2.2 Kinect Middleware

The Kinect uses an on-board processor to map raw data to render three dimensional images. The middleware uses a randomised decision forest to distinguish objects from backgrounds, human body parts, joints, faces and movement [11] [13]. The algorithm was trained from over 1 million training examples [11]. By using machine learning, the Kinect is able to model raw data into 3D avatar shapes. The combine use for hardware and software allows the Kinect to detect and track 48 different human joints on up to 6 human bodies and repeats 30 times every second [14]. Figure 6 shows the flow of data in Kinect: the data is captured by the camera, then processed into information by the middleware and finally passed into end-use software to create useful output.

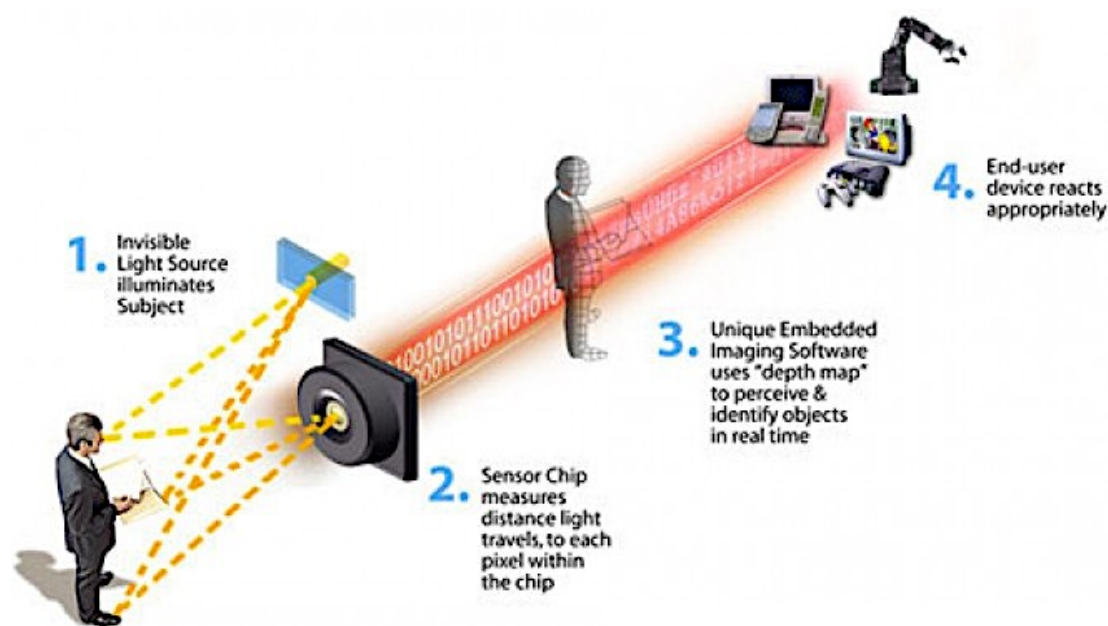


Figure 6: How data is passed from hardware to middleware and finally to end-user software inside the Kinect. Taken from [13] <https://www.wired.com/>

3.2.3 Justifications for choosing the Kinect

The Kinect provides precise detection of human joints which is crucial to the implementation of this Application. More on the design is explained in Design section below.

The Kinect is a motion sensing device that is also developed by Microsoft, so it has good compatibility with the HoloLens [6].

In addition, there are several open source projects that utilises the Kinect as inputting devices for HoloLens uses e.g. Project-Infrared [15]/Cloud Compare [16]. They can be used in development/as reference in this project.

3.3 Unity

Unity is used to develop the Hololens application. Unity is a cross-platform game engine that is mainly used for game development. Unity supports the development of 2D and 3D graphics. C# is mainly used for scripting. Unity is the recommended engine for creating 3D apps for Hololens [17].

The application is developed in Unity in C#. Unity is suitable because 3D graphics is used in this project. Unity also has a comprehensive asset store, which offers ready-to-use 3D drum kit models [18].

3.4 Development Environment

Microsoft HoloLens applications are developed in Visual Studio with Windows 10 SDK. Unity is used as the game engine. The Microsoft HoloLens and the Microsoft HoloLens Emulator is used for testing. The Emulator is mainly used because the HoloLens is not readily available throughout the development process.

The following is a standard work flow in HoloLens application development [19]:

1. Build the 3D models and code the application logic in Unity.
2. Build and compile the Unity app as a Unity C# project under a Windows Store/HoloLens setting.
3. Load the compiled project into Visual Studio as a Solution.
4. Compile and deploy the project assembly to the HoloLens or the Emulator.

The following platform/tools are used in this project:

Table 1: Platforms/Tools used in this project

Platforms/Tools	Notes
Windows 10 Education Operating System	Hyper-V is enabled.
Windows 10 SDK (10.0.16299.15)	
Visual Studio 2017	The following workloads are installed <ul style="list-style-type: none">- Universal Windows Platform development- Game Development with Unity
HoloLens Emulator and Holographic Templates (build 10.0.14393.1358)	The Emulator requires Hyper-V support in the system. Only Windows 10 Pro, Enterprise, or Education versions support Hyper-V.
Unity 2017.1	Windows Store .NET Scripting Backend is installed to allow building into a Visual Studio Solution.
MixedRealityToolkit - Unity (aka HoloToolKit)	Open source
HoloLens-Kinect	Open source

4. Design

4.1 Set up of the MR DrumKit

Users should sit down while using the application. In front of the user, a Kinect module should be set up such that the whole body of the user can be captured (see Figure 7) i.e. the user should be at least 0.9 m and not further than 6 m from the Kinect camera.

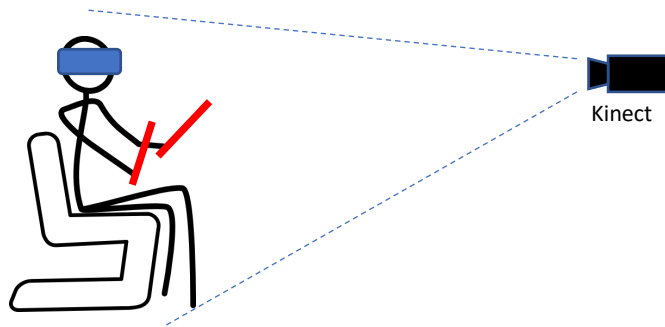


Figure 7: Set up of the application. Kinect should be set up such that the whole body of user can be captured. Stickman figure taken from [20] <http://www.clker.com/>

4.2 Using the MR DrumKit

After opening the application, users can choose between the 2 drumming modes. The Kinect captures the position and movement of the user's hands and feet. The processor inside Kinect models the movement and turn it into a 3D human avatar. The information is sent to the Hololens through the wireless network. Then the Hololens calculates if the user has hit a certain drum, and produces the corresponding sound through its speakers.

There are two methods of implementing the drum "hit" algorithm. One is position-based detection and the other one is velocity-based detection. Due to insufficient information on the hands-on device performance experience of the Hololens and Kinect, the method is decided during the implementation stage. The final design is discussed in Part 5.3.3 and the results are discussed in Part 6.8.

Programs are used for turning data collected in Kinect to Unity-compatible information. Open source programs such as Project-Infrared and Cloud Compare are examined for the adaptability for the project during the implementation stage. The final design is discussed in Part 5.3.2 and the results are discussed in Part 6.5.

4.2.1 Standard Air Drums mode

In the standard air drums mode, a hologram of a drum kit is placed on the ground in front of the user. Users interact with the hologram like a real physical drum. The sounds and position of the drum kit can be customised.

4.2.2 Custom Drum Kit mode

In the custom drum kit mode, the Hololens uses its scanning ability to recognise flat surfaces designated by the users as drums. Users can place drums on any flat surfaces. For example, a user can choose a table surface as drum #1, and his chair armrest as drum #2. Whenever he hits the designated surfaces, the corresponding drum sound is produced.

4.2.3 Application Customisation

User experiences can be customised in Options. Option features are listed as below:

- Volume adjustment
- Switch between drum sounds/drum kits
- Switch between drum skins
- Customise position of the drums

4.2.4 Extra Features for Future

If development is smooth, extra features is added to the application to enhance user experience.

4.2.4.1 Tutorial Mode

A tutorial mode can be added. Music scores of simple beats is shown beside the drum kit. When the score is played, the way of playing the beat is shown on the drums via visual aid. Users can follow the instructions and the visual aids to learn the beat. Harder drum scores can be unlocked as user progresses.

4.2.4.2 Musical Score display

Users can import musical scores into the application. A virtual score can be displayed during a drum play session. Space can be further saved when physical space for scores are not needed. Score storage and management can also be facilitated.

4.2.4.3 Music Import Support

Users can import music into the application. Imported tracks can be played during a drum play session. Users can play drums along music they like.

4.3 Limitations of Hardware

4.3.1 Kinect

Due to the fixed location of Kinect, users can only remain in the area of vision of Kinect during playing. Users must face the camera in order to let Kinect capture users' motions.

4.3.2 Hololens

The field of vision is limited to 100 degrees from the users' point of view [21]. This is because of the limited size of the transparent screen and limited computational power on the Hololens. Holographic images are not displayed outside the screen area.

4.3.3 User Environment

MR DrumKit must be used in an environment where wireless network is steadily available. The Wi-Fi technology is utilised for communication between the Kinect and the Hololens.

5. Previous works/Final Status

5.1 Background Research on Related Technologies

Research has been done to understand how the HoloLens and Kinect work. We have also looked at demos of applications and projects with a similar objective to know the performance and limitations of HoloLens and Kinect. From the demos, it is expected that the HoloLens cannot perform complicated calculations, or else the system will be slowed. In addition, motion capturing of Kinect might have time lag. Therefore, design of MR DrumKit must be careful and well-engineered to ensure a smooth and real time user experience. [22] [23]

5.1.1 Motion Capturing Technology

For the motion capturing, there are 2 ways of deciding whether a drum is hit. The first one is to simply count it as a “hit” whenever the hand is detected at a specific position. The second one is to take in also the velocity of the hand as well as the position to determine if a drum is hit. The drum only produces sound when it is hit with a certain speed. Further research and experiments were done in the implementation stage to get the balance between realistic and smooth user experience.

5.2 Development Environment Setup

For Hololens application development, Visual Studio and Unity running on Windows 10 is required.

5.2.1 Initial Setup (before 22nd October 2017)

For app development in this project, such an environment was set up using VMWare in a MacBook Pro. As Hololens applications are still not very common in the field, support for development is lacking. Compatibility issues arise between old and new versions of required software. Different versions were tried before the working combination is found.

The Hololens Emulator provided by Microsoft is used to test the application on a computer without a physical Hololens. The Emulator was set up in the said development environment. However, as the development environment is in a virtual machine, the speed and performance are very slow but workable.

5.2.2 Second Setup (after 22nd October 2017)

After further testing on the Initial Setup environment, it is decided that the slow speed and performance of the above setup is not a desirable environment. Although the environment is workable for test applications, it is prone to having worse performance issues when the application is more complex. A new development environment is set up in a Windows 10 desktop computer. All development onwards will be done in the new environment. For the specifications of the machine, please refer to Appendix 9.3.

Programs for testing the development environment and the capabilities of the Hololens have been written (e.g. scripts for placing objects on the ground). However, because of the limited availability of hardware devices (the Hololens device and the Kinect), the actual development progress was largely hindered.

5.3 Logic Design

The logic design is done along with the implementation stage. Changes to the initial design are done on the fly due to practicability issues and technological limitations. This part examines the final design in a factual manner. The consideration and justifications to the changes are discussed in the Part 6 Result Discussion.

5.3.1 Game Set up

The basic setup for the game remains the same as described in Part 4.1 “Set up of the MR DrumKit”. The Kinect is set up in a high place such that the whole body of the user can be captured. The user puts on the HoloLens and faces the Kinect during gameplay. The user can stand up or sit down. The user should stand from at least 1m and not more than 3 m from the Kinect.

5.3.2 HoloLens – Kinect Communication

The open source project HoloLens-Kinect [24] is used as a reference for enabling communication between the HoloLens and the Kinect. This project uses Unity scripts to send Kinect data to the HoloLens. This project’s scripts are derived from the HoloLens Sharing example in the HoloToolkit and the KinectView example from the Kinect Tool Unity Pro Packages. Minor changes are applied to the scrips for the purpose of this app. The project utilises 3 main components: Sender, Receiver and Server. The components are discussed as follows:

Sender

The Sender is a Unity app that reads in Kinect skeletal data and broadcasts it to other devices through the Server. It runs on the Windows machine which is connected with Kinect. This program captures the human skeleton by detecting the joints in the human body. There are 25 joints per skeleton. Users can see the detected skeleton(s) on the sender's display (Figure 8).

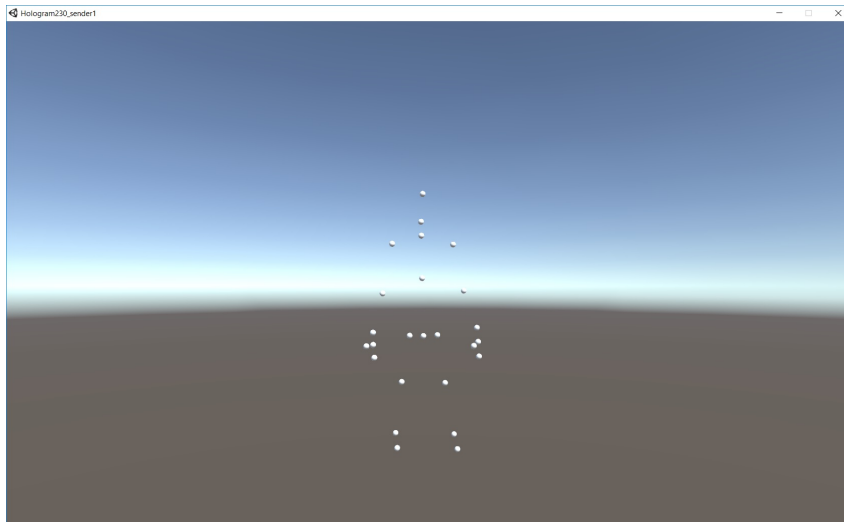


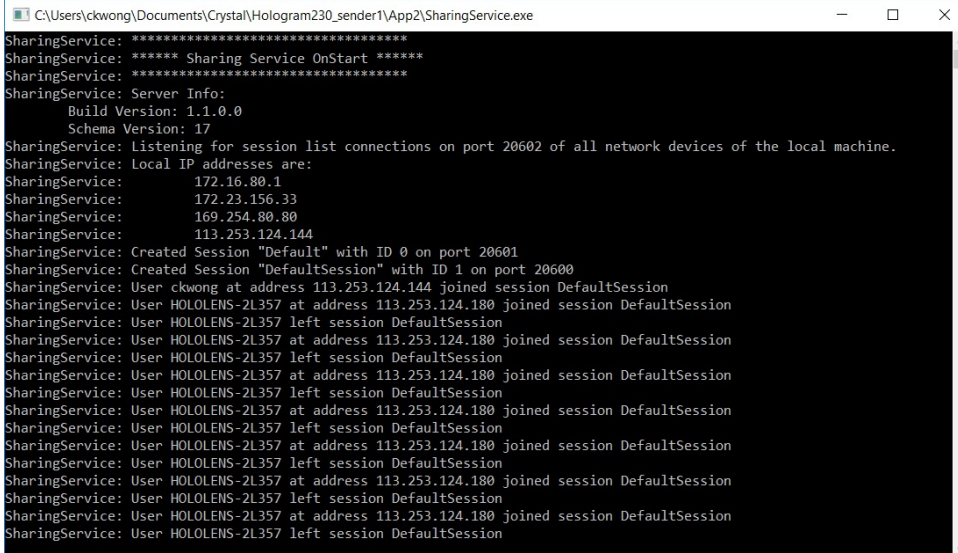
Figure 8: Display of Sender

Receiver

The Receiver is a Unity app that can be deployed on the HoloLens that listens for messages about Kinect skeletal data and renders it to screen. The skeleton is displayed as 25 spheres – 1 sphere per joint. The Head of the skeleton is mapped to the position of the user (technically the HoloLens), such that when the user moves, he can see the skeleton moves in the same place. The Receiver also renders the drum holograms to the screen. This app detects if there are any collisions between the skeleton and the drum holograms. If yes, a corresponding sound is played.

Server

The Server is an exe program that runs on Command Prompt on the Windows machine. This program is provided as the Sharing Session Manager by the HoloToolkit. This program accepts connections from devices and helps broadcasting messages across the devices (Figure 9).



```
C:\Users\ckwong\Documents\Crystal\Hologram230_sender1\App2\SharingService.exe
SharingService: *****
SharingService: ***** Sharing Service OnStart *****
SharingService: *****
SharingService: Server Info:
Build Version: 1.1.0.0
Schema Version: 17
SharingService: Listening for session list connections on port 20602 of all network devices of the local machine.
SharingService: Local IP addresses are:
SharingService: 172.16.80.1
SharingService: 172.23.156.33
SharingService: 169.254.80.80
SharingService: 113.253.124.144
SharingService: Created Session "Default" with ID 0 on port 20601
SharingService: Created Session "DefaultSession" with ID 1 on port 20600
SharingService: User ckwong at address 113.253.124.144 joined session DefaultSession
SharingService: User HOLOLENS-2L357 at address 113.253.124.180 joined session DefaultSession
SharingService: User HOLOLENS-2L357 left session DefaultSession
SharingService: User HOLOLENS-2L357 at address 113.253.124.180 joined session DefaultSession
SharingService: User HOLOLENS-2L357 left session DefaultSession
SharingService: User HOLOLENS-2L357 at address 113.253.124.180 joined session DefaultSession
SharingService: User HOLOLENS-2L357 left session DefaultSession
SharingService: User HOLOLENS-2L357 at address 113.253.124.180 joined session DefaultSession
SharingService: User HOLOLENS-2L357 left session DefaultSession
SharingService: User HOLOLENS-2L357 at address 113.253.124.180 joined session DefaultSession
SharingService: User HOLOLENS-2L357 left session DefaultSession
SharingService: User HOLOLENS-2L357 at address 113.253.124.180 joined session DefaultSession
SharingService: User HOLOLENS-2L357 left session DefaultSession
SharingService: User HOLOLENS-2L357 at address 113.253.124.180 joined session DefaultSession
SharingService: User HOLOLENS-2L357 left session DefaultSession
```

Figure 9: Display of Server

The 3 components should be run in the order: Server, Sender and Receiver.

5.3.3 Drum hit Detection

This application uses a position-based collision detection. A drum sound is played when the palm of the hand collides with the surface of the drum components. Different components output a different sound, mimicking real drums i.e. the snare produces a sharp, high-frequency staccato sound, the floor tom produces a low-frequency sound, the crash cymbal produces a loud sharp crash sound. The user can play the drums with any hand.

As for drums that are played with a foot pedal i.e. the bass drum, the displacement of the knee is measured to detect the motion of striking the floor of the right foot. The component that is played with the left foot i.e. the pedal for the adjustment of looseness of the hi-hat, is omitted in this stage of the project. This is because this component involves high mechanical complexity and it cannot be easily implemented given the circumstances of the technology at this stage. For further technical discussions on the complexity of the hi-hat, please refer to Appendix 9.4.

5.3.4 Game Mode

The final product only has the Standard Air Drumming mode (as mentioned in Part 4.2.1). The Custom Drum Kit mode (Part 4.2.2) is omitted at this stage because of technical and time constraints. However, the intention of having the second mode is incorporated into the first mode as a function. Further discussions on such changes are in Part 6 Result Discussion. There are also no settings and customisation available.

When the user launches the app, he should stand still from about 1 m from the Kinect to let the HoloLens maps his skeleton with his position. When success, the user can see the skeleton in the position of his body. Most noticeably the user should see the skeleton's hands moving in sync with his hands. The other parts of the skeleton are not as noticeable because of HoloLens's clipping plane. The spheres that represent the user's palm are coloured red for easy distinction (Figure 10).

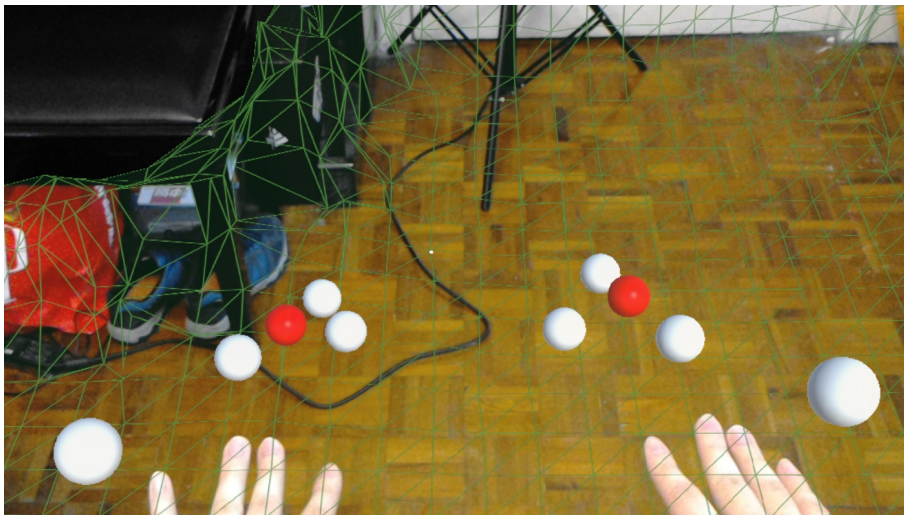


Figure 10: User's palms are in red.

A green mesh is visible on the environment (Figure 11). This mesh is the result of the Spatial Mapping from the HoloLens. The mesh should match closely to the shape of the environment, and it is right above the surface. Such mesh is rendered for the environment within 4-5 meters from the user.



Figure 11: Environment Mesh

There are 3 cubes floating in front of the user (Figure 12). They are buttons for activating different functions.



Figure 12: The 3 cubes floating in front of user

On the top left, there is a cyan cube. This button is for drum placement. When air tapped, the program starts to record the position of user's palms (Air tap is a HoloLens gesture equivalent to a click). When the palms stop moving for a few seconds, a drum set is placed between the user's hands and at the level of his feet (Figure 13). The size of the drum set is scaled approximately to the distance between the palms. The user can start playing with the drum set immediately (Figure 14). The user can reposition the drum again by air tapping the cyan cube. The cube rotates when activated and stops when the placement is complete.



Figure 13: A placed drum



Figure 14: User playing the drums

On the top middle, there is a grey cube. This button is for toggling the visibility of the green environment mesh (Mesh disabled in Figure 15).



Figure 15: Mesh made invisible

On the top right, there is a pink cube. This button is for activating an “Easter Egg Mode”. This mode makes the drum components gravity-less. The user starts shooting balls out from his right hand (Figure 16). The user can hit the drum components with the balls. The balls collide with the drums and they start floating away (Figure 17). The drum components and the balls behave as if they are in the room under no gravity (Figure 18) e.g. their motion is stopped when hitting a wall, floors etc (Figure 19, Figure 20). Drum components stop moving when the cube is tapped again. The cube rotates on activation and stops when deactivated.

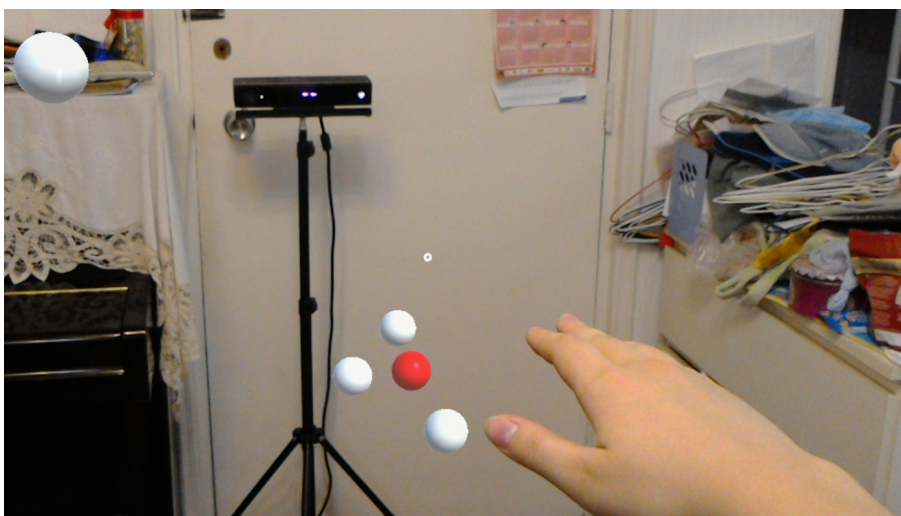


Figure 16: Ball shooting out of Right hand



Figure 17: Drum components hit by the balls



Figure 18: Drum components floating around

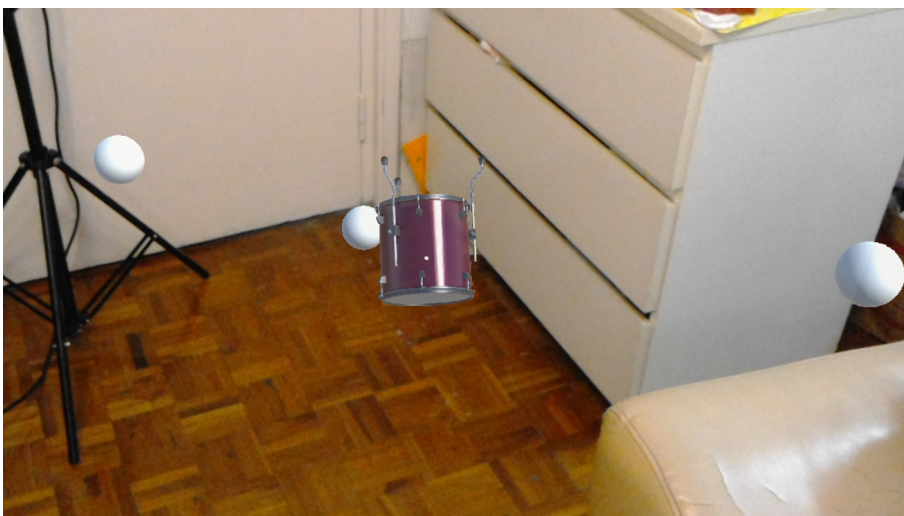


Figure 19: Drum stopped by walls and floor



Figure 20: Crash stopped by ceiling

6. Result Discussion

This part discusses the considerations and justifications of the implementation and also compares the results with expectations.

6.1 Development Process

The substantial development was faster than we expected. The programming started later than planned, but because of the built-in engine of Unity and the availability of open source programs and Development Kits, the process of coding is largely facilitated. However, the HoloLens app development environment is still complicated. The modelling and programming are done within Unity, then the app should be built into a Visual Studio solution, be deployed in the HoloLens and only then it could be tested. Many steps are involved, and time is needed to get familiarised with the process. The building and deploying are also time-consuming.

In addition, HoloLens app development is a rather new topic, so there are not a lot of support and resources on the Internet. Often, we cannot find solutions to a certain technical problem online, so we have to resort to testing things out by ourselves.

6.2 Testing

There were 3 ways of testing the HoloLens app: The Unity Editor Player, the HoloLens Emulator and the HoloLens itself. Initially, we opted to use the Emulator because the HoloLens might not be always available. However, there were some technical problems and we could not utilise the Emulator properly. We could not find the solution on the net, so we were forced to give up on it.

Most application-level logic could be tested with the in-built Unity Editor Player. It was significantly more convenient to do so than deploying to the HoloLens every time. However, the Unity Player cannot fully simulate the HoloLens. We still have to test the app on the HoloLens when testing functions involving dimensions and cameras. Also, there are bugs in the HoloLens version that are not present in the Unity player version, despite that they run the exact same source code. HoloLens apps should be tested thoroughly to prevent undiscovered bugs.

6.3 Use of Kinect

It is convenient to use the Kinect on a Windows machine. The Kinect can be used right after related SDKs are installed. The SDK includes software to monitor the status of the Kinect (Figure 21). The outcome of the motion capture can be shown and we can easily check whether the Kinect is working normally or not.

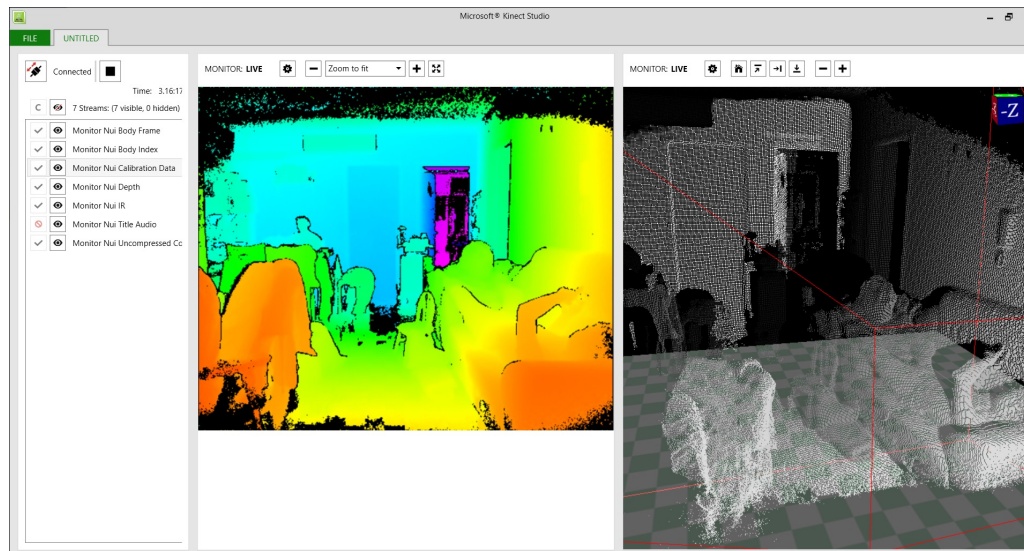


Figure 21: Using Kinect Studio to monitor the Kinect

However, there are some limitations to using the Kinect. The play area is limited to a few metres in front of the camera. Players cannot walk around freely. It was quite counterintuitive, as users naturally want to walk around especially when they see holograms on the other side of the room. When they leave the Kinect play area, they cannot interact with the holograms with their bodies. It is rather difficult to solve this problem because it is an inbuilt limitation of the Kinect. We might need to change the choice of hardware if we want to alleviate this problem.

6.4 Use of HoloLens

The performance of the HoloLens is better than expected. At the beginning of this project, we were told that the HoloLens's computational power is limited. In addition, we saw some demo videos that show the functionality of the HoloLens was rather primitive. Therefore, during the initial stage of development, we thought that we have to keep the design very simple and the performance might be slow. However, after playing with other HoloLens apps and getting hands-on development experience, we found that most of the time the performance is quite smooth and natural.

6.5 Use of HoloLens—Kinect Project

HoloLens—Kinect is chosen because it has a brief introduction on the functionality of the scripts. It is also intended that developers use it as an add-on function. There is more freedom on customising the scripts for the application's own use. On the other hand, Project Infrared is a finished Unity app solely with the intent to demonstrate the hardware's ability to project user movement to an avatar. It is not as flexible. Therefore, HoloLens—Kinect is chosen.

This open source program is of primitive minimal functionality. It extracts the raw data points from the Kinect and relays them to the HoloLens app. There is no processing of the data so it has only a primitive appearance, out of scale, have a lot of noise, and not very fault tolerant.

The motion capture is not very accurate. The spheres that represent the joints shifts around and vibrates occasionally even when the person is not moving. This phenomenon is especially severe at the fingers. The hand is represented by 3 spheres: 1 for the palm and 2 for the fingers. The 2 spheres that represent the fingers vibrate violently all the time. This leads to detection of false positives of drum hits, so we decided to use only the palm sphere for drum-hit detection. However, this does not feel natural as users expect that they can hit the drums with their fingers too. Users need to reach out their arms to hit a drum with their palms, which is not intuitive. For future development, algorithms can be added to smooth the shaking of the fingers.

The motion capture is also not very accurate when there are multiple persons, especially when they are physically close to each other. Their limbs can get jumbled up or they are detected as one person. Occasionally a chair with armrests is detected as a human being. This inaccuracy is hard to fix, because the recognition of human skeleton is done by the Kinect. Developers can only take this into consideration when choosing motion capturing devices.

6.6 Use of Sender-Receiver-Server System

This app requires 3 components running at the same time in order to work. It is cumbersome and inconvenient to set up: this app needs a windows machine and a Kinect, while most HoloLens app can be run independently. However, there are only a few methods that allow the capturing of whole-body data and this is the only conveniently available method at the time of development.

This method also requires Wi-Fi communication, which could be a double-edged sword. Users can move around freely without having cables attached to the headset, but it requires a reasonably fast Wi-Fi network.

6.7 User position mapping

There were difficulties when trying to map the skeleton to the physical location of the user. This is because the Kinect does not know the position of the user with respect to the physical space. In this project, the head sphere of the skeleton is mapped forcefully to the in-app Camera object (which is the HoloLens). However, this method is not accurate. The hand of the skeleton and the user's hand might be 20-30 cm apart.

In addition, the skeleton is not of the right scale. The skeleton can go up to 2-3 meters tall when displayed. This is due to that the Kinect does not know about real-world scales and the HoloLens does not know what scale the skeleton is measured in. As a workaround, a coefficient is hardcoded into the program to scale down the skeleton. This might not be the best fitting number for every user. In the future, it is expected that a dynamical method can be implemented to make the skeleton human-sized.

6.8 Drum Hit Detection

This application recognises drum hits when the hand collides with the drums, as opposed to the use of drumsticks in real life. This is because the Kinect can only detect the human skeleton. We could also have used algorithms to project a length from the hand as sticks, but this is rather difficult as the Kinect does not detect the rotation of the joints. In reality, the sticks point to different directions as the wrist rotates. It would be difficult to realise such fine detail.

This application also uses a position-based collision detection, as opposed to a velocity-based method. The first method is adopted because it is easier to implement. It is planned that when the first method is successfully implemented and other aspects of the app are mature, we can build the second method on top of the first one. However, this is yet to be realised because of time constraints. This could be left for future development, where the user can control the amplitude of the hit with his velocity.

6.9 Drum Placement

This function is implemented to allow users customise the location and the size of the drum set. The users might also be standing or sitting down, so they might want to recalibrate the drum. The drum set is put between the palms and on the level of the feet. We could also have opted to place the drum on the floor by utilising the Spatial Mapping module i.e. placing the drum on the floor mesh. However, the first method is chosen because it is technically easier to get the coordinates of the feet than the floor. Nevertheless, the outcome should also be similar.

6.10 Easter Egg Mode

The Easter egg mode is added in the place of the Custom Drum Kit mode. During the implementation of the Custom Drum Kit mode, technical difficulties are not easily overcome. The mode in question involves techniques such as using the Spatial Understanding module, World Anchors and possibly advanced gestures such as dragging. The original intent of having such a mode is to demonstrate the “environment awareness” capability of the HoloLens: the HoloLens does not only put a 3D graphic on top of reality; its holograms can interact with the reality. In addition, this mode offers an interesting change to the standard drumkit model.

After rounds of tries to implement the Custom Drum Kit mode, we decided to first implement a function that achieves the same intention. The Easter Egg mode is a function added to the Standard Air drumming mode. It is so named because it is a minor function with no direct connection to the main function and it is for fun.

The environment scanning function is provided by the Spatial Mapping module in the HoloToolkit. It is easy to use and rather accurate. It also excludes humans from the environment scanning.

6.11 General Gameplay

In general, the gameplay is not immersive enough. One feedback from testers is that the position of the skeleton is too far away from his hands, so he does not feel like he is hitting the drums. In addition, there is no physical rebound when hitting the drums, which further reduces the realism. It also does not have a UI. On the technical side, the application is still buggy and sometimes has bad performance issues. For example, skeletons are not removed when the user leaves the play field, hindering the detection of a new user. Also, when the drum is scaled to be

wider than 1 metre, the frame rate is unplayably slow. The application has a large room for improvement.

6.12 Discussion Conclusion

Although the core idea remains the same, the design of this application is largely different from the initial design. This is because we have further understood the capacity and performance of the Kinect and the HoloLens, and changed the design and implementation method in the course of development. The drumming function performed at around expectation, but the addition of the gravity-less mode was not foreseen.

This application is far from a commercial standard, but satisfies for a prototype. It is expected that further development can be done on fixing bugs and smoothing the performance, as well as adding UI and adding customisations.

7. Conclusion

We have presented the final status of the development of a mixed reality drum kit application. This application allows users to play visible virtual drum sets regardless of the availability of a real physical drum kit.

This paper introduces and describes the implementation and the discussions of results of MR DrumKit. This application does not reach a commercial standard but is a workable prototype, as designed in the project's Objective. It is expected that further development to be done to turn the prototype into a fully playable application.

8. References

- [1] Aerodrums, "Aerodrums," [Online]. Available: <http://aerodrums.com/aerodrums-product-page/>. [Accessed 22 October 2017].
- [2] K. Schiano, "Aerodrums - A Virtual Drum Kit!!!," 17 December 2016. [Online]. Available: <https://www.youtube.com/watch?v=wgptHLBI0e4>. [Accessed 23 October 2017].
- [3] Mickey, "【正宗打空氣】 aerodrums-虛擬空氣鼓夜晚打唔怕嘈/," 20 01 2016. [Online]. Available: <https://www.pcmarket.com.hk/>. [Accessed 28 November 2017].
- [4] OoO, "DrumKit VR - Play drum kit in the world of VR," OoO, 6 July 2016. [Online]. Available: http://store.steampowered.com/app/496910/DrumKit_VR__Play_drum_kit_in_the_world_of_VR/. [Accessed 22 October 2017].
- [5] S. Yu, "DrumKit VR - Play drum kit in the world of VR," 3 6 2016. [Online]. Available: https://www.youtube.com/watch?v=q9d8kJf_c1c. [Accessed 28 November 2017].
- [6] J. Callahan, "Merging the Microsoft HoloLens and Kinect could get us one step closer to the holodeck," Windows Central, 28 June 2016. [Online]. Available: <https://www.windowscentral.com/merging-microsoft-hololens-kinect-could-create-holodeck>. [Accessed 22 October 2017].
- [7] S. Colaner, "What's Inside Microsoft's HoloLens And How It Works," tom's Hardware, 23 August 2016. [Online]. Available: <http://www.tomshardware.com/news/microsoft-hololens-components-hpu-28nm,32546.html>. [Accessed 22 October 2017].
- [8] A. Dachis, "What's the Difference Between HoloLens, Meta & Magic Leap?," Next Reality, 15 October 2016. [Online]. Available: <https://mixed.reality.news/news/whats-difference-between-hololens-meta-magic-leap-0171361/>. [Accessed 22 October 2017].
- [9] Microsoft, "Microsoft HoloLens," 28 11 2017. [Online]. Available: <https://www.microsoft.com/en-us/hololens>. [Accessed 28 November 2017].
- [10] V. Koifman, "Microsoft HoloLens Cameras," 26 5 2016. [Online]. Available: <http://image-sensors-world.blogspot.hk/2016/05/microsoft-hololens-cameras.html>. [Accessed 28 November 2017].
- [11] J. MacCormick, "How does the Kinect work?," 2011. [Online]. Available: <https://users.dickinson.edu/~jmac/selected-talks/kinect.pdf>. [Accessed 22 October 2017].
- [12] Microsoft, "Kinect for Xbox One," 28 11 2017. [Online]. Available: <https://www.xbox.com/en-US/xbox-one/accessories/kinect>. [Accessed 28 November 2017].
- [13] T. Carmody, "HOW MOTION DETECTION WORKS IN XBOX KINECT," Wired, 3 11 2010. [Online]. Available: <https://www.wired.com/2010/11/tonights-release-xbox-kinect-how-does-it-work/>. [Accessed 28 November 2017].
- [14] R. W. Robert Cong, "How Does The Xbox Kinect Work," Jameco, 28 11 2017. [Online]. Available: <https://www.jameco.com/jameco/workshop/howitworks/xboxkinect.html>. [Accessed 28 November 2017].

- [15] WavelengthLLC, "Project-Infrared," 2016. [Online]. Available: <https://github.com/WavelengthLLC/Project-Infrared>. [Accessed 28 November 2017].
- [16] CloudCompare, "CloudCompare," [Online]. Available: <http://cloudcompare.org/>. [Accessed 28 November 2017].
- [17] Microsoft, "Development overview," [Online]. Available: https://developer.microsoft.com/en-us/windows/mixed-reality/development_overview. [Accessed 28 September 2017].
- [18] Unity Asset Store, "Drum kit for your band," 30 March 2017. [Online]. Available: <https://www.assetstore.unity3d.com/en/#!/content/84693>. [Accessed 22 October 2017].
- [19] J. Odom, "How to Build a Basic HoloLens App in Minutes," Next Reality, 10 12 2016. [Online]. Available: <https://hololens.reality.news/how-to/hololens-dev-101-build-basic-hololens-app-minutes-0175021/>. [Accessed 28 November 2017].
- [20] Clker, "SITTING STICKMAN CLIP ART," 28 11 2017. [Online]. Available: <http://www.clker.com/clipart-26132.html>. [Accessed 28 November 2017].
- [21] SeanL@WCC, "Analyzing HoloLens Field of View (FOV) – Updated," 25 6 2015. [Online]. Available: <http://windowscomments.com/?p=660>. [Accessed 28 November 2017].
- [22] A. Sharma, "Air-Drumming with Kinect," 15 September 2013. [Online]. Available: https://www.youtube.com/watch?v=jMNOCa_jz-o. [Accessed 1 October 2017].
- [23] D. Herrero, "Air Drums with Kinect and Unity," 7 April 2016. [Online]. Available: <https://www.youtube.com/watch?v=oZHwNFxANfs>. [Accessed 1 October 2017].
- [24] michell3, "Hololens-Kinect," 23 Mar 2017. [Online]. Available: <https://github.com/michell3/Hololens-Kinect>. [Accessed 14 04 2018].
- [25] C. Myers, "Hi Hat Parts – How Does A Hi Hat Work?," [Online]. Available: <https://www.musicskanner.com/hi-hat-parts-hi-hat-work/>. [Accessed 14 04 2018].
- [26] E. Limer, "How Microsoft's HoloLens Works Its Holographic Magic," Popular Mechanics, 1 May 2015. [Online]. Available: <http://www.popularmechanics.com/technology/gadgets/a15324/how-microsofts-hololens-works/>. [Accessed 27 September 2017].
- [27] S. Colaner, "What's Inside Microsoft's HoloLens And How It Works," tom's Hardware, 23 August 2016. [Online]. Available: <http://www.tomshardware.com/news/microsoft-hololens-components-hpu-28nm,32546.html>. [Accessed 27 September 2017].
- [28] A. Dachis, "What's the Difference Between HoloLens, Meta & Magic Leap?," Next Reality, 15 October 2016. [Online]. Available: <https://mixed.reality.news/news/whats-difference-between-hololens-meta-magic-leap-0171361/>. [Accessed 27 September 2017].
- [29] J. MacCormick, "How does the Kinect work?," 2011. [Online]. Available: <https://users.dickinson.edu/~jmac/selected-talks/kinect.pdf>. [Accessed 27 September 2017].
- [30] Aerodrums, "Aerodrums," [Online]. Available: <http://aerodrums.com/aerodrums-product-page/>. [Accessed 29 September 2017].

- [31] D. Lewis, "Aerodrums Review," 2 July 2015. [Online]. Available: https://www.youtube.com/watch?v=8EFFcy_vaQk. [Accessed 29 September 2017].
- [32] K. Schiano, "Aerodrums - A Virtual Drum Kit!!!," 17 December 2016. [Online]. Available: <https://www.youtube.com/watch?v=wgptHLBI0e4>. [Accessed 29 September 2017].
- [33] JPhoenix19, "In case you were wondering... (An honest review of Aerodrums)," 30 November 2014. [Online]. Available: <http://www.sevenstring.org/threads/in-case-you-were-wondering-an-honest-review-of-aerodrums.284473/>. [Accessed 29 September 2017].
- [34] J. Callaham, "Merging the Microsoft HoloLens and Kinect could get us one step closer to the holodeck," 28 June 2016. [Online]. Available: <https://www.windowscentral.com/merging-microsoft-hololens-kinect-could-create-holodeck>. [Accessed 1 October 2017].

9. Appendices

9.1 Original Project Schedule in Project Plan

Sept	Get familiar with Unity
Oct 1	Project plan Project website
Oct	Collect user requirements Analysis and Design program
Nov - Dec	<u>Achieving the intermediate goal</u> Development with Hololens (w/ emulator) <ul style="list-style-type: none"> - Displaying with Hololens - Scanning with Hololens Development with Kinect <ul style="list-style-type: none"> - Motion capturing with Kinect
Jan - Feb	Implementation of communication between devices Implementation of two drum modes
Jan 8-12	First Presentation
Jan 21	Intermediate Report
Feb - Mar	<u>Achieving the final goal</u> Fine tuning <ul style="list-style-type: none"> - Improving user interface - Adding customisation features - Sounds and graphics enhancement
Apr	Testing and optimisation
Apr 16-20	Final Presentation
Apr 15	Final Report
May 2	Project Exhibition

9.2 Revised Schedule in Interim Report

Table 2: Revised Schedule

Month	Progress	Status
Sept	Get familiar with Unity	Completed.
Oct	<ol style="list-style-type: none"> 1. Collect user requirements 2. Analysis and Design program 3. Do background research on related technologies 	Partially completed; application logic design is not completed.
Nov	<ol style="list-style-type: none"> 1. Do research on implementation details of related technologies 2. Set up the Development environment 	Completed.
Dec - Feb	<p><u>Achieving the intermediate goal</u></p> <ol style="list-style-type: none"> 1. Development with Kinect <ul style="list-style-type: none"> - Motion capturing with Kinect - Experimenting on drum-hit detection - Experimenting on data modelling 2. Development with Hololens (w/ emulator) <ul style="list-style-type: none"> - Displaying with Hololens - Scanning with Hololens 3. Experiments on connectivity between Hololens and Kinect 4. Application logic design 	Delayed; part with Kinect has not yet begun.
Feb	<ol style="list-style-type: none"> 1. Implementation of communication between devices 2. Implementation of two drum modes 	To be completed
Feb - Mar	<p><u>Achieving the final goal</u></p> <p>Fine tuning</p> <ul style="list-style-type: none"> - Improving user interface - Adding customisation features - Sounds and graphics enhancement 	To be completed
Apr	Testing and optimisation	To be completed

*Please see Appendices 9.1 for the original Project Schedule in Project Plan

9.3 Development Machine Specifications

System Information report written at: 04/15/18 17:24:55

System Name: DESKTOP-4ND97VB

[System Summary]

Item Value

OS Name Microsoft Windows 10 Education

Version 10.0.16299 Build 16299

Other OS Description Not Available

OS Manufacturer Microsoft Corporation

System Name DESKTOP-4ND97VB

System Manufacturer System manufacturer

System Model System Product Name

System Type x64-based PC

System SKU SKU

Processor Intel(R) Core(TM) i5-7400 CPU @ 3.00GHz, 3001 Mhz, 4 Core(s), 4 Logical Processor(s)

BIOS Version/Date American Megatrends Inc. 2002, 9/20/2016

SMBIOS Version 3.0

Embedded Controller Version 255.255

BIOS Mode UEFI

BaseBoard Manufacturer ASUSTeK COMPUTER INC.

BaseBoard Model Not Available

BaseBoard Name Base Board

Platform Role Desktop

Secure Boot State On

PCR7 Configuration Binding Not Possible

Windows Directory C:\WINDOWS

System Directory C:\WINDOWS\system32

Boot Device \Device\HarddiskVolume4

Locale United States

Hardware Abstraction Layer Version = "10.0.16299.371"

User Name DESKTOP-4ND97VB\ckwong
Time Zone China Standard Time
Installed Physical Memory (RAM) 16.0 GB
Total Physical Memory 15.9 GB
Available Physical Memory 10.9 GB
Total Virtual Memory 18.3 GB
Available Virtual Memory 12.4 GB
Page File Space 2.38 GB
Page File C:\pagefile.sys
Virtualization-based security Running
Virtualization-based security Required Security Properties
Virtualization-based security Available Security Properties Base Virtualization Support,
Secure Boot
Virtualization-based security Services Configured
Virtualization-based security Services Running
Device Encryption Support Reasons for failed automatic device encryption: TPM is not
usable, PCR7 binding is not supported, Hardware Security Test Interface failed and device is
not InstantGo, Un-allowed DMA capable bus/device(s) detected, TPM is not usable
A hypervisor has been detected. Features required for Hyper-V will not be displayed.

9.4 Hi-hat Mechanism

This appendix is about the mechanics of a hi-hat and why it is omitted for the project.

The hi-hat is the basic component for keeping the beat. It is one of the most frequently used drum components. A hi-hat consists of 2 cymbal, and a foot pedal for adjusting the looseness of the 2 cymbals, as shown in Figure 22.



Figure 22: Hi-hat

Taken from <https://www.musicskanner.com/hi-hat-parts-hi-hat-work/> [25]

The hi-hat is usually hit with a stick with the pedal down i.e. the cymbals are closed tightly. A short, crisp, muted sound is produced. A variation of playing the hi-hat is to hit it with the pedal up i.e. the cymbals are fully apart. A shimmering, sustained sound is produced. Advanced skills involve hitting the hi-hat with different looseness, thus producing a huge spectrum of sounds.

Even if we confine only to detect the 2 sounds (i.e. open and closed), it depends only on a slight elevation of the left foot. As it is known that the Kinect is not sensitive enough for such little movement, we decided to omit the left foot pedal. The hi-hat in the project only produces the most basic sound i.e. the closed sound.