ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL

ENHANCING VIRTUAL REALITY MUSICAL INSTRUMENT DESIGN : SOLVING SOFTWARE TOPOLOGY PROBLEMS WITH VRMI CREATION TOOLKIT

Ph.D. THESIS

Ozan SARIER

Department of Music

Music Programme

JANUARY 2021

ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL

ENHANCING VIRTUAL REALITY MUSICAL INSTRUMENT DESIGN : SOLVING SOFTWARE TOPOLOGY PROBLEMS WITH VRMI CREATION TOOLKIT

Ph.D. THESIS

Ozan SARIER
(409122004)

Department of Music

Music Programme

Thesis Advisor: Prof. Dr. Can Karadoğan

JANUARY 2021

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

SANAL GERÇEKLİK MÜZİK ENSTRUMANI TASARIMININ İYİLEŞTİRİLMESİ : YAZILIM MİMARİSİ SORUNLARININ VRMI CREATION TOOLKIT İLE ÇÖZÜMÜ

DOKTORA TEZİ

Ozan SARIER
(409122004)

Müzik Anabilim Dalı

Müzik Doktora Programı

Tez Danışmanı: Prof. Dr. Can KARADOĞAN

OCAK 2021

Ozan Sarıer, a Ph.D. student of ITU Graduate School student ID 409122004, successfully defended the thesis/dissertation entitled "ENHANCING VIRTUAL REALITY MUSICAL INSTRUMENT DESIGN : SOLVING SOFTWARE TOPOLOGY PROBLEMS WITH VRMI CREATION TOOLKIT", which he prepared after fulfilling the requirements specified in the associated legislations, before the jury whose signatures are below.

Thesis Advisor : **Prof. Dr. Can KARADOĞAN** ....
Istanbul Technical University

Jury Members : **Prof. Dr. Murat YAKIN** ....
Bahçeşehir University

Assoc. Prof. Yahya Burak TAMER ....
Bahçeşehir University

Assoc. Prof. Taylan ÖZDEMİR ....
Istanbul Technical University

Assist. Prof. Gökhan DENEÇ ....
Istanbul Technical University

Date of Submission : 24 December 2020
Date of Defense : 15 January 2021

In the memory of my beloved grandmother, Prof. Dr. A. Muhibbe Darga.

FOREWORD

I would like to thank my advisor Can KARADOĞAN for all his support and motivation through the years. I would also want to thank my committee, Yahya Burak TAMER, Taylan ÖZDEMİR, Murat YAKIN, and Gökhan DENEÇ. Their guidance and support made this thesis possible.

I also would like to take a step back in time and thank Dirk STROMBERG, Koray TAHİROĞLU, Pieter SNAPPER, and Reuben de LAUTOUR for their wonderful teaching and their role in my undergraduate and graduate education. I am also in debt to Amy SALSGIVER, who made this thesis possible by helping me become an early adopter of an HTC Vive VR set.

No amount of thanks can suffice for the support of my friends and family, who were able to bear with me in this long process. I would like to express that I am well aware of their value and thank them dearly.

İTÜ MIAM has been a home for me for the last decade and I would also like to thank everyone who kept it alive, including the directors Cihat AŞKIN, Şehvar BEŞİROĞLU, and Yelda Özgen ÖZTÜRK.

Pure Data is an important part of this work and without the brilliant work of Miller PUCKETTE, the libPD community and Niall MOODY VRMI Creation Toolkit wouldn’t exist. I would like to thank everyone in the PD community for their parts.

December 2020 | Ozan SARIER
Researcher

TABLE OF CONTENTS

Page

FOREWORD ..... **ix**
TABLE OF CONTENTS ..... **xi**
ABBREVIATIONS ..... **xv**
LIST OF FIGURES ..... **xvii**
SUMMARY ..... **xix**
ÖZET ..... **xxi**
1. INTRODUCTION **1**
1.1 The Motivation Behind Virtual Reality Instruments 3
1.1.1 From the viewpoint of economics 4
1.1.2 Creating a VRMI just for the sake of it 4
1.1.3 Creating economical versions of previous musical instruments 4
1.1.4 Non-serious music instruments 4
1.1.5 Novel VRMI paradigms 5
1.1.5.1 Obstruction of the performers' view 5
1.1.5.2 The designer not only creates the instrument but also the virtual environment around it 5
1.1.5.3 The problem of physical contact 5
1.1.5.4 Breakage of the linkage between the physical location of the performer and the virtual performing space 6
2. BACKGROUND **7**
2.1 Lutherie and Evaluation of VRMIs 9
2.1.1 Traditional lutherie versus DMI lutherie 9
2.1.2 Evaluation of DMIs and VRMIs 10
2.1.2.1 DMI/VRMI evaluation from the viewpoint of HCI 10
2.1.2.2 Learnability, learning curve and progressability 11
2.1.2.3 Efficiency 12
2.1.2.4 Audience interaction 13
2.2 History of Virtual Reality Technology 13
2.2.1 History of early virtual reality platforms 15
2.2.1.1 Morton Heilig: Sensorama, 1962 15
2.2.1.2 Charles Comeau and James Bryan: Headsight, 1961 16
2.2.1.3 Ivan Sutherland: Sword of Damocles, 1968 16
2.2.1.4 Thomas A. Furness III: Super-Cockpit, 1968 17
2.2.1.5 Jaron Lanier: Virtual Reality and VPL Research, 1985 17
2.2.1.6 Jonathan D. Waldern: Virtuality, 1991 17
2.2.1.7 Nintendo: Virtual Boy, 1995 18
2.2.2 Second wave of virtual reality 18
2.2.2.1 Oculus Rift Development Kit (DK1) 19
2.2.2.2 Oculus Rift DK2 19
2.2.2.3 Oculus Rift CV1 20
2.2.2.4 Steam VR and HTC Vive 21
2.2.2.5 Google Cardboard, Samsung Gear VR 22

2.2.2.6 Playstation VR 23
2.2.2.7 Standalone devices with limited functionality 24
2.3 Virtual Reality Music Instruments 24
2.3.1 Academic, research, experimental or artistic/installation VRMIs 24
2.3.1.1 ChromaChord 24
2.3.1.2 Wedge 25
2.3.1.3 Coretet 25
2.3.1.4 VIVR 26
2.3.2 Published VRMIs 26
2.3.2.1 EXA: The Infinite Instrument 27
2.3.2.2 Paradiddle 27
2.3.2.3 DrumMaster VR 28
2.3.2.4 Percussive VR 29
3. KEY CONCEPTS IN VIRTUAL REALITY 31
3.1 Definition of Virtual Reality 31
3.2 Augmented Reality and Differences with Virtual Reality 32
3.3 Technical Requirements of A Virtual Reality System 32
3.3.1 HMD and vision 33
3.3.1.1 Resolution 33
3.3.1.2 Refresh rate 34
3.3.1.3 IPD 34
3.3.1.4 Head-mounting 34
3.3.1.5 Optics 35
3.3.1.6 Generation of graphics 36
3.3.2 Tracking 36
3.3.2.1 IMU and internal tracking 37
3.3.2.2 Outside tracking 37
3.3.2.3 Inside-Out tracking 37
3.3.2.4 Room-scale tracking 38
3.3.2.5 Controller tracking 38
3.3.2.6 Hand tracking 38
3.3.3 Locomotion 39
3.3.3.1 No-locomotion 40
3.3.3.2 Directional movement with controller input 40
3.3.3.3 Room scale locomotion 41
3.3.3.4 Teleportation 41
3.3.3.5 Alternative means of locomotion 42
3.3.4 Interaction and input 43
3.3.4.1 Usage of the existing input devices 43
3.3.4.2 Gaze and stare 44
3.3.4.3 Head movements as confirmation 44
3.3.4.4 Single-button 44
3.3.4.5 Custom tracked controllers 45
3.3.4.6 Hand tracking 45
3.3.4.7 Haptic technologies 46
3.3.5 Sound 46
3.4 VR Specific Problems 47
3.4.1 VR sickness 47
3.4.2 Screen door effect 48
3.4.3 IPD and focusing problems 49

3.4.4 Effects of lag 49
3.4.5 Ergonomics 50
3.4.6 Fatigue 50
3.4.7 Visual-Haptic communication 51
3.4.8 Collision with real world 51
3.4.9 Locomotion 52
3.4.10 Self-representation 52
3.4.11 Going in and out of immersion 53
4. PROPOSED MODEL OF DESIGNING AND BUILDING VRMIs WITH A CHANCE OF SURVIVAL 55
4.1 Definition of A Good Instrument 56
4.2 Benefits of Survival 57
4.3 Problem of Survival 58
4.4 VRMI Design Strategies Towards Survival 60
4.4.1 Goal of publishability 60
4.4.2 Increasing the ease of development and lowering development costs 61
4.4.3 Platform compatibility 62
4.4.4 Accepting a variety of users 63
4.4.5 Altering the learning curve using performing aids and gamification 65
4.4.6 Improving user engagement with inspirations from video gaming 65
4.4.7 Marketing 67
4.4.8 User feedback 67
4.4.9 Data-collection and analytics 68
4.4.10 Updates 68
4.4.11 Fun 69
4.4.12 Consideration of VR specific issues 71
4.5 Proposed Model for VRMIs That Can Survive 74
4.5.1 Development platform 75
4.5.2 Development target 75
4.5.3 Audio engine 75
4.5.4 Starting point and helper libraries 76
5. VRMI CREATION TOOLKIT 77
5.1 Dependencies of the VRMI Creation Toolkit 79
5.2 Topology and the Elements of the VRMI Creation Toolkit 80
5.3 Object Hierarchy 82
5.3.1 Interactables 82
5.3.1.1 Presence detector 82
5.3.1.2 Pass-thru detector 83
5.3.1.3 Rotary potentiometer 83
5.3.1.4 Linear fader 83
5.3.1.5 Buttons 83
5.3.1.6 Distance detector 83
5.3.2 PD Audio source 83
5.3.3 Template PD patch 84
5.4 Capabilities and Benefits of the VRMI Creation Toolkit 85
5.5 Initial Setup 86
6. CONCLUSIONS 87
REFERENCES 90
CURRICULUM VITAE 94

ABBREVIATIONS

AR : Augmented Reality
DMI : Digital Music Instrument
IMU : Internal Measuring Unit
HMD : Head Mounted Display
VR : Virtual Reality
VRMI : Virtual Reality Music Instrument

LIST OF FIGURES

Page

Figure 2.1 : Transistor Count per Microprocessor by Year, 1971–2018. .... **7**
Figure 2.2 : Yamaha DX7. .... **8**
Figure 2.3 : Depiction of Plato's Allegory of the Cave **14**
Figure 2.4 : Morton Heiling's Sensorama. **15**
Figure 2.5 : Charles Comeau and James Bryan's: Headsight. **16**
Figure 2.6 : Ivan Sutherland's Sword of Damocles **16**
Figure 2.7 : Image from Super-Cockpit Project **17**
Figure 2.8 : VRS Virtuality Sit Down Unit **17**
Figure 2.9 : Nintendo Virtual Boy **18**
Figure 2.10 : Oculus Rift DK1 .... **19**
Figure 2.11 : Oculus Rift DK2 teardown **20**
Figure 2.12 : Oculus Rift CV1 with its peripherals. **20**
Figure 2.13 : HTC Vive headset, lighthouses, and controllers. **21**
Figure 2.14 : Depiction of HTC Vive play area **21**
Figure 2.15 : Cardboard VR. **22**
Figure 2.16 : Samsung's Gear VR. **22**
Figure 2.17 : PlayStation VR System with its camera and controllers. **23**
Figure 2.18 : Screenshot from Chroma Chord. **25**
Figure 2.19 : Screen capture from Wedge. **25**
Figure 2.20 : Coretet virtual instruments **26**
Figure 2.21 : Screenshot from Vivr performance. **26**
Figure 2.22 : EXA VR Promotional Screenshot **27**
Figure 2.23 : Screenshot from Paradiddle. **28**
Figure 2.24 : Drum Master VR promotional Screenshot. **29**
Figure 2.25 : Screenshot from Percussive VR **30**
Figure 2.26 : HTC Vive's adjustable strap system. **35**
Figure 2.27 : Close-up of HTC Vive's lenses. **35**
Figure 2.28 : Inside out tracking in action. **37**
Figure 2.29 : LeapMotion controller connected to a laptop **39**
Figure 2.30 : Teleportation control's in the game titled "The Lab". **42**
Figure 2.31 : Virtuix Omni. **42**
Figure 2.32 : Depiction of the screen door effect on various headsets. **48**
Figure 2.33 : Steam VR's chaperone system. **52**
Figure 2.34 : "Is the instrument ready for a performance?" **59**
Figure 2.35 : Sarıer's RubSynth. **70**
Figure 2.36 : Screenshot from VRMI Creation Toolkit. **81**
Figure 2.37 : Screenshot from VRMI Creation Toolkit, Unity Editor. **82**
Figure 2.38 : Unity Editor – PD Audio Source. **84**
Figure 2.39 : Template PD Patch **85**

ENHANCING VIRTUAL REALITY MUSICAL INSTRUMENT DESIGN: SOLVING SOFTWARE TOPOLOGY PROBLEMS WITH VRMI CREATION TOOLKIT

SUMMARY

This thesis aims to provide an insight into the current state of virtual reality music instruments and provides a framework for enhancing the design of such instruments. The framework consists of a set of design recommendations and an accompanying software library to streamline the creation process.

With recent developments in the virtual reality field, inexpensive consumer virtual reality equipment and related software development tools became available in the near past. Albeit being primarily designed for entertainment and video gaming, this new wave of affordable virtual reality devices also became an instrument for research, education, and artistic experimentation. Although the concept of virtual reality has been established for centuries and the technology to implement it has been present for decades, widespread adoption of the virtual reality technology happened on the so-called "second wave of virtual reality" due to the computing and technological requirements in fields of graphics rendering, displays, MEMS devices (micro-electrical mechanical systems), optics and many more. During this so-called "second wave of virtual reality", several manufacturers were able to produce and sell sub-1000 USD systems with the advent of modern GPU's, affordable MEMS IMU's, miniature high-resolution displays, and capable optics. Coupled with initial launch titles in the form of video games, the launch of the affordable consumer virtual reality headsets and system was well received and acquired a user base that was significant for the third-party developers to create content for the systems released. Currently, affordable virtual reality technology is actively developed and offers an exciting field for digital music instrument researchers and designers.

While virtual reality technology is currently being actively developed and deployed, the affordable hardware, software offerings, and software development tools are predominantly focused on gaming and related entertainment. This predominant focus on gaming and related entertainment reflect itself into virtual reality music instrument design with the lack of software development tools to embed real-time synthesis into distributable virtual reality music instrument software.

When the existing virtual reality music instrument creations are inspected, two dominant trends are clearly visible. The first trend is academic virtual reality music instrument creations, where advanced synthesis is often used to create serious music instruments, but the resulting software cannot be packaged into a distributable product and deemed to cease existing due to licensing and software topology issues. The latter trend is amongst the commercially available virtual reality music instruments. These instruments use existing development tools that are created for video gaming in mind to achieve distributability, but the lack of sound generation options limits these

instruments to simple sample playback, therefore severely limiting the sonic possibilities of a virtual reality music instrument.

Enhancing virtual reality instrument design first requires understanding the concepts of virtual reality, the technical aspects of currently available systems, virtual reality related paradigms, and virtual reality related problems. This thesis aims to enlighten the reader about the topics above with sections that are documenting them.

Secondly, in order to enhance the design, some questions such as “What is a good instrument?” and “What is a better instrument?” become important. When the existing digital music instrument research is inspected, there is not a single true answer to these questions, and in what way to enhance the virtual reality music instrument is unclear. However, amongst all the metrics to evaluate an instrument, the survivability aspect of an instrument is a common trait between all the instruments we care to remember. Therefore, this thesis argues that survivability is one of the most important qualities for a new instrument as it is a pre-requisite for any future advancements. The survivability of an instrument depends on many aspects and is virtually incalculable. While there is no formula for an instrument that will definitely survive, an instrument that will not survive can be formulated as there are several solid necessities for survival. In terms of software music instruments, the most obvious one of such necessities is publishability. If an instrument can’t be published and distributed, it will only serve the creator. Time has shown us that it is often the case that such unpublished instruments are even forgotten by their creators most of the time.

VRMI Creation Toolkit developed alongside this thesis aims to remedy the issues of survivability by alleviating the problems rooted in the software topology of contemporary virtual reality music instruments. As a helper library, VRMI Creation Toolkit enables developers to use their existing knowledge of music programming together with already established video game focused development tools; creating the possibility of real-time sound synthesis in a publishable and distributable end product.

VRMI Creation Toolkit also employs pre-made virtual reality related interaction objects and greatly reduces the development time. As survival is not always guaranteed, VRMI Creation Toolkit aims to increase the chances of survival by reducing the development cost and increase a developer’s chances to try again.

SANAL GERÇEKLİK MÜZİK ENSTRUMANI TASARIMININ İYİLEŞTİRİLMESİ : YAZILIM MİMARİSİ SORUNLARININ VRMI CREATION TOOLKIT İLE ÇÖZÜMÜ

ÖZET

Bu tez çalışması, sanal gerçeklik müzik enstrümanlarının mevcut durumlarını inceleyip, enstrüman tasarımını iyileştirmek için bir çerçeve sunmaktadır. Sunulan tasarım çerçevesi, yaratım sürecini kolaylaştıran bir takım dizayn tavsiyeleri ve tasarıma yardımcı olacak bir yazılım kütüphanesinden oluşmaktadır. Sunulan yardımcı yazılım kütüphanesi VRMI Creation Toolkit ile müzik teknolojisi alanında sık kullanılan Pure Data ses programlama ortamında yaratılan gerçek zamanlı ses sentezleyicilerin sanal gerçeklik uygulamaları içinde gömülü olarak çalışması kolaylaştırılmıştır.

Sanal gerçeklik alanındaki son gelişmelerle, geniş kitlelerce satın alınabilir, düşük maliyetli sanal gerçeklik donanımları ve ilgili yazılım geliştirme araçları yakın geçmişte hayatımıza girmiştir. Hayatımıza giren bu yeni nesil düşük maliyetli sanal gerçeklik donanımları, her ne kadar öncelikli olarak eğlence ve video oyunları için tasarlanmış olsa da, araştırma, eğitim ve sanatsal deneyler için uygun bir araç haline gelmiştir. Sanal gerçeklik konsepti asırlar, sanal gerçekliği mümkün kılacak donanım da on yıllarca evvele dayansa da, sanal gerçekliğin geniş kitlelere ulaşması, bilgisayar grafikleri, ekran teknolojisi, MEMS (mikro-elektrikli mekanik sistemler) aletler, optik teknolojiler ve benzeri teknolojik ve bilişimel gelişmelere ihtiyaç olduğundan dolayı ancak geçtiğimiz yıllarda, çoğu kişi tarafından sanal gerçekliğin ikinci dalgası olarak adlandırılan dönemde gerçekleşebilmiştir. Sanal gerçekliğin ikinci dalgası diye adlandırılan bu dönemde, düşük fiyatlı grafik işlemciler, MEMS IMU'lar, minyatür ekranlar ve yeni optik tasarımlar sayesinde 1000 Amerikan doları altında kalan son kullanıcı fiyatları ile sanal gerçeklik donanımları piyasaya sürülmüştür. Bilgisayar oyunları formunda, toplumsal merak ve heyecan oluşturan yazılım kütüphaneleri ile birlikte pazarlanınca, bu donanımlar toplum tarafından kabul görüp, azımsanamayacak kadar büyük bir kitleye ulaşmıştır. Ulaşılan kitlenin büyüklüğü de üçüncü parti yazılım geliştiricilerinin ilgisini çekecek kadar büyük olduğundan, birçok yazılım geliştiricisini alana çekmiştir. Günümüzde sanal gerçeklik teknolojisi aktif olarak geliştirilmektedir ve dijital müzik enstrümanı araştırmacıları ve tasarımcıları için de heyecan verici bir alan sunmaktadır.

Sanal gerçeklik teknolojisindeki yazılımsal ve donanımsal gelişmeler aktif olarak devam etmekte olsa da teknolojinin çıkış noktası ağırlıklı olarak bilgisayar oyunları üzerine kurulu olduğundan dolayı, yazılımların çoğunluğu, yazılım geliştirme araçları ve donanımların gelişimi de kişisel eğlence üzerine odaklanan bir şekilde ilerlemektedir. Bilgisayar oyunları ve benzer eğlence üzerine odaklanan bu trend, sanal gerçeklik müzik enstrümanı tasarımına gerçek zamanlı ses sentezlemeye imkan yaratmayan yazılım geliştirme ortamları olarak yansımaktadır.

Sanal gerçeklik müzik enstrümanları üzerine yapılan çalışmalar incelendiğinde, birbirinden kolayca ayrışan, iki baskın akım kolayca gözlemlenebilir. Bu akımlardan ilki olan akademik sanal gerçeklik enstrümanları, çoğu zaman ciddi enstrümanlar yaratabilmek için gerçek zamanlı ses sentezleme kullanmakta olup, bu gerçek zamanlı ses sentezlemenin uygulanış şekli sebebiyle tek bir yazılım olarak paketlenemeyen ve dağıtılamayan yazılımlardan oluşmaktadır. Yazılım mimarisi, lisanslama problemleri ve çoklu yazılım gereği gibi sebepler bu sorunun kaynağında yatmaktadır. Yayımlanamamaktan dolayı bu enstrümanların çoğu var olamadan kaybolmuştur. Baskın akımlardan ikincisi ise ağırlıklı olarak bilgisayar oyunlarının dağıtımlarının yapıldığı dijital platformlar aracılığıyla son kullanıcıya ulaşan sanal gerçeklik müzik enstrümanlarının içinde bulunduğu akımdır. Bu trende dahil sanal gerçeklik müzik enstrümanları, geliştirilme aşamasında bilgisayar oyunu geliştirmek amaçlı tasarlanmış yazılım geliştirme ortamları kullanıldığından dolayı paketlenebilir ve dağıtılabilir yazılımlar olsalar da yazılım mimarilerindeki eksikliklerden ve gerçek zamanlı ses sentezleme yerine kayıtlı seslerin geri çalınması prensibine dayalı olarak çalıştıklarından dolayı oldukça kısıtlı ses çıktısına sahiplerdir.

Sanal gerçeklik müzik enstrümanı tasarımını zenginleştirmenin ilk koşulu, mevcut teknolojiyi detaylı bir şekilde anlamak ve sanal gerçekliğe özgü durum, problem ve paradigmaları anlamaktan geçmektedir. Bu tez, bu hususları belgeleyip okuyucusunu bilgilendirmeyi amaçlamaktadır.

İkincil olarak, sanal gerçeklik müzik enstrümanı tasarımını iyileştirmeden evvel, "iyi bir enstrüman nedir?" ve "Bir Enstrümanı ne daha iyi yapar?" gibi sorular cevaplanmalıdır. Geçmiş literatür incelendiği zaman, bu soruların cevabı olabilecek tek bir doğru bulunmamaktadır. Müzik enstrümanı tasarımı çok yönlülüğü sebebiyle sabit kurallar arasında sınırlı kalamamaktadır. Buna rağmen, çoğu başarılı enstrüman arasında ortak bir özellik göze çarpmaktadır – varlığını sürdürebilme. Bu bilgi ışığında, bu tez dahilinde bir enstrümanın varlığını sürdürme kabiliyeti en önemli hedefler arasında sunulmuştur.

Bir müzik enstrümanının varlığını sürdürme kabiliyeti bir çok etkene dayanmakta olup, bu kabiliyetinin seviyesinin ölçülmesi veya hesaplanması nerdeyse imkansızdır. Varolma kabiliyetini geliştirecek veya garanti edecek bir formül olmasa da, bir müzik enstrümanının varolma kabiliyetini negatif etkileyeceği sabit olan bir sürü etken bulunmaktadır. Bir müzik enstrümanı yazılımı konu olduğunda, varolma yeteneğinin ilk gerekliliklerinden biri yayımlanabilirliktir. Bir yazılım müzik enstrümanı yayımlanmadığı sürece varoluşu sadece yaratıcısının kullanımı altında olacaktır. Geçmişe baktığımızda, bu tür müzik enstrümanların çoğunun yok olduğunu veya unutulduğunu görebiliriz.

Bu tez beraberinde geliştirilen VRMI Creation Toolkit (Sanal Gerçeklik Müzik Enstrümanları Geliştirme Kütüphanesi), güncel sanal gerçeklik müzik enstrümanlarının yazılım mimarisinden kaynaklı oluşan enstrümanın varoluşunu sürdürememesi problemini gidermeyi amaçlamaktadır. Bir yazılım geliştirme yardımcı kütüphanesi olan VRMI Creation Toolkit, yazılım geliştiricilerinin aslen bilgisayar oyunlarının geliştirlmesi için tasarlanmış olan Unity yazılım geliştirme ortamında,gerçek zamanlı ses sentezleme tekniklerini, çoğu müzik teknolojileri araştırmacılarının hakim olduğu Pure Data ses programlama dili ile uygulamasına imkan sağlamaktadır. Gerçek zamanlı ses sentezleme imkanının yanında, sanal gerçeklik dahilinde kullanılacak interaksyon modelleri VRMI Creation Toolkit ile hazır olarak geldiğinden dolayı, sanal gerçeklik müzik enstrümanı tasarımı bu yardımcı kütüphane ile büyük ölçüde hızlanmıştır. Bu yazılım mimarisi ile yapılacak yazılımlar, hem geniş bir ses yelpazesine sahip olma imkanına, hem de tek bir yazılım olarak paketlenip yayımlanabilir olma imkanına sahiptir. Yardımcı kütüphane sayesinde, gerçek zamanlı ses sentezleme yeteneğine sahip sanal gerçeklik müzik enstrümanı tasarımının muazzam bir şekilde hızlanmasış olup, üretim maliyetinin büyük kısmını oluşturan tasarım maliyeti de beraberinde düşürülmüştür. Bu sayede düşük maliyet ile içerik olarak zengin sanal gerçeklik müzik enstrümanları yaratmak mümkün kılınmıştır.

1. INTRODUCTION

With recent developments in the virtual reality field, inexpensive consumer virtual reality equipment and related software development tools became available in the near past. Albeit being primarily designed for entertainment and video gaming, this new wave of affordable virtual reality devices also became an instrument for research, education, and artistic experimentation.

Although the concept of virtual reality can be traced back to almost 360 BC to Plato's The Allegory of the Cave in his famous BookVII of The Republic (Steinicke, 2016:20), the first realized electronic hardware implementation, Ivan Sutherland's Sword of Damocles, can be traced back to 1968, (Steinicke, 2016:25), widespread adoption of the virtual reality technology happened on the so-called "second wave of virtual reality" due to the computing and technological requirements in fields of graphics rendering, displays, MEMS devices (micro-electrical mechanical systems), optics and many more (Anthes, 2016:1) (Dempsey, 2016:2). During this so-called "second wave of virtual reality", several manufacturers were able to produce and sell sub-1000 USD systems with the advent of modern GPU's, affordable MEMS IMU's, miniature high-resolution displays, and capable optics. Coupled with initial launch titles in the form of video games, the launch of the affordable consumer virtual reality headsets and system was well received and acquired a user base that was significant for the third-party developers to create content for the systems released. Wide-spread adoption of the consumer virtual reality headsets started and the adoption rate was predicted to be 19 million headsets in North America by the year 2019 (Rogers, 2017). Even with the successful launch and initial userbase, the initial user count was not high enough for big productions and a lack of AAA gaming titles endangered the future of the platforms (Lilly, 2017)(Rosenberg, 2018). The lack of AAA titles leads to a wave of indie developers developing for the platform.

While the general PC and Console gaming market was convoluted and extremely competitive, the lack of new titles created a space for indie developers. With a relatively small but content starved market, several virtual reality art experiences were also able to find an audience in the form of VR short movies and interactive experiences (Zimmerman, 2019). Another support for virtual reality adoption came from an unexpected source, the adult entertainment industry in the form of virtual reality adult content (Sutton, 2019) (Greene, 2018). At the time of writing, a great number of consumer virtual reality headsets have been sold (Rogers, 2019), and many different and capable devices are commercially available.

The availability of affordable hardware and the associated userbase also attracted developers and researchers from non-gaming, non-entertainment areas. The effect of the availability of affordable consumer virtual reality headsets on research can be seen from the growing number of published articles employing such consumer devices with diverse subjects like training medical professionals, use-of-force simulation, psychology (Serafin et al, 2019), Amblyopia treatment (Blaha and Gupta, 2014), driving simulation (Taheri et al, 2017), musical scenarios (Serafin et al, 2016) and many more subjects. (Cipresso et al, 2018)

Musicians, composers, and digital luthiers were also attracted by the affordable systems. When virtual reality musical research and available software instruments are examined, two software design approaches are observed (Serafin et al, 2017).

The first approach, mainly observed within published research, consists of virtual instruments, installations, or experiences developed using a combination of independent software, which are not packaged together. Although this is not problematic for prototyping, setting up installations, and a small volume user base, this approach creates a huge problem for developing a marketable product to the end-user.

The second approach is observed amongst the currently marketed software which falls under the category of virtual reality instruments. In this second approach, developers try to use the sound playback and generation capabilities of a game development platform such as Unity or Unreal Engine (Url-1, Url-2). Unfortunately, said capabilities are either too limited or too complex to develop anything beyond simple sample playback. This creates a barrier for VRMI developers, limiting their ability to use their existing skillsets for creating complex synthesis and sound generation (Url-3).

Although there are many aspects to DMI development, this dissertation focuses mainly on one of them, the survivability of an instrument, and assumes that it is the most important aspect of an instrument. A discussion on the reasoning behind this assumption will be presented in the following chapters of the dissertation. Along with the said discussion, a software framework for creating VRMI's – The VRMI Creation Toolkit is proposed.

VRMI Creation Toolkit is a framework that consists of a software design topology, design recommendations, and a software library to help VRMI development and research. The aim of the VRMI Creation Toolkit and this dissertation is to create a new approach in which VRMI developers and researchers can use their existing skillset to the full extent while maintaining the ability to create a shippable end product.

1.1 The Motivation Behind Virtual Reality Instruments

One can ask the following question "Why shall we create virtual reality instruments?" A quick answer to the said question is simply "Why not?". Humankind experimented with creating musical instruments with nearly every material, object, technology, and concept within its grasps. From prehistoric bone flutes (Sachs, 1941, p. 49) to engineering feats like Teleharmonium (Weidenaar, 1995), our history is filled with many musical instruments.

Virtual reality is just another technology with which we can create musical instruments. Virtual reality instruments are already being created, and relevant research is already becoming conducted.

The motivation behind these instruments and research varies just like the previous instruments in history. When we look at history, while some instruments are created for solving problems, enabling previously impossible things, sonic expressive enhancements, or creating an economic alternative to another instrument, some are created just for the sake of it, financial gain, or exploring possibilities.

All of the above can be a valid motivation towards creating a virtual reality music instrument. History has shown us that even shortcomings of an instrument can be a positive thing as it creates new paradigms for music-making in many aspects.

1.1.1 From the viewpoint of economics

As previously discussed, there is an established and growing userbase of virtual reality system owners. If one can interest even a small portion of the said userbase, worthwhile financial gain from the sale of a VRMI can be expected.

1.1.2 Creating a VRMI just for the sake of it

Although it is often discussed as a potential pitfall in DMI design, there is nothing inherently wrong with creating an instrument merely for using a certain new technology.

1.1.3 Creating economical versions of previous musical instruments

Virtual reality can be used to create instruments that mimic or fill the role of previously established instruments. As there is no material cost, VRMI's can be a cheap substitute. One example is a VR modular synth. While there is already software present that can simulate the functionality of a modular synth, they are constrained to a computer screen. VR can simulate the whole room as an interface, and the functionality of a modular synth for the fraction of a cost.

1.1.4 Non-serious music instruments

Not all the music instruments serve the purpose of just performing music. Many musical instruments are bought by hobbyists and serve the purpose of entertaining its user. There are many traditional and electronic non-serious, toy musical instruments. These non-serious music instruments or toys with limited capabilities can fit many roles in many areas such as entertainment, child development, education, and even serious music-making.

One imaginary example of such a scenario is a virtual recreation of a hard to reach instrument or an ancient instrument lost in time. Even with limited functionality, this can be an educational tool or a ludic experience satisfying public interest.

1.1.5 Novel VRMI paradigms

VRMI’s bring many novel paradigms. Compared to a traditional physical instrument, a VRMI is very different. While some of the differences are in the form of shortcomings and limitations, some of them are in the form of new possibilities that could never be imagined before.

The shortcomings and limitations of VRMI’s can have a positive effect by pushing the designer to find novel concepts. The new possibilities available through the use of VRMIs can lead to previously unthinkable music creation scenarios.

1.1.5.1 Obstruction of the performers' view

While using a VR Headset, the users' view is totally blocked and replaced with a virtual image, usually in the form of a virtual environment. This obstruction makes it impossible for the performer to see the audience or other musicians (if present) in traditional ways.

If needed, visual contact with the audience or other musicians must be supplied by other means. This limitation can force the designer to find novel scenarios where this contact can be established.

One imaginary example is supplying the audience with the point of view of the performer. A second imaginary example can be a virtual environment where the audience and other performers are represented with virtual avatars. A third example can be an online performance in which the audience or other performers are no longer in the same physical room with the performer.

1.1.5.2 The designer not only creates the instrument but also the virtual environment around it

While traditional instrument designers were only responsible for creating the instrument itself, VRMI designers must also take the environment around the instrument into consideration. This creates another responsibility for the designer and makes VRMI design even more multidisciplinary.

1.1.5.3 The problem of physical contact

As there is no physical instrument in a VRMI, there is no physical contact with the instruments. This can be remedied by using haptic feedback, custom input devices, or strategies that can make use of the existing physical objects around the performer; or it can be ignored altogether by the designer and force the designer to come up with contactless ways of performing.

This can lead to VRMIs that have novel control methods and suggest or impose new ways of thinking, organization, and interaction.

1.1.5.4 Breakage of the linkage between the physical location of the performer and the virtual performing space

With VRMIs, the linkage between the real performing space and the virtual one is broken. The performer is no longer constrained by the physical space around the instrument. Alternatively, the performing space can be even more constrained than the actual space. Creative and novel concepts can spawn from the control of the performing space.

One imaginary example is a VRMI built into an MMORPG (massively-multiplayer online role-playing game). The performing space becomes the game world, and other players become the audience. While this scenario can sound like a non-serious musical instrument at first, it doesn't necessarily have to be. Most of the MMORPGs have virtual currencies in them which can be exchanged with real-world currency. "Farming" of this virtual currency is being done by many people (Heeks,2009). Many families in hyper-inflation stricken Venezuela resort to "farming" in order to make a living Dubrowski et al, 2019) (Heeks, 2008). While being a somewhat dystopian one, this is an example of how the limitations and possibilities of VRMI's can lead to new paradigms of music performance.

2. BACKGROUND

Since the invention of computing, humankind experimented with the possibilities it brought and digital musical instruments were also a topic of experimentation. While at first the computing capabilities only allowed offline sound generation or the control of external analog sound generators, later developments resulted in computers that can generate real-time audio synthesis. In 1965, Gordon E, Moore suggested that the number of transistors packed in about every two years and will result in doubling the computing power. Moore's prediction, often known as "Moore's law", was true for many years and this resulted in personal computers capable of generating and processing real-time audio by the early 1990s (Schaller, 1997). As Moore's prediction carried on, at the time of writing even handheld devices such as smartphones have enormous computing capabilities for real-time audio processing. As seen in Figure 2.1, Moore's prediction still carries on.

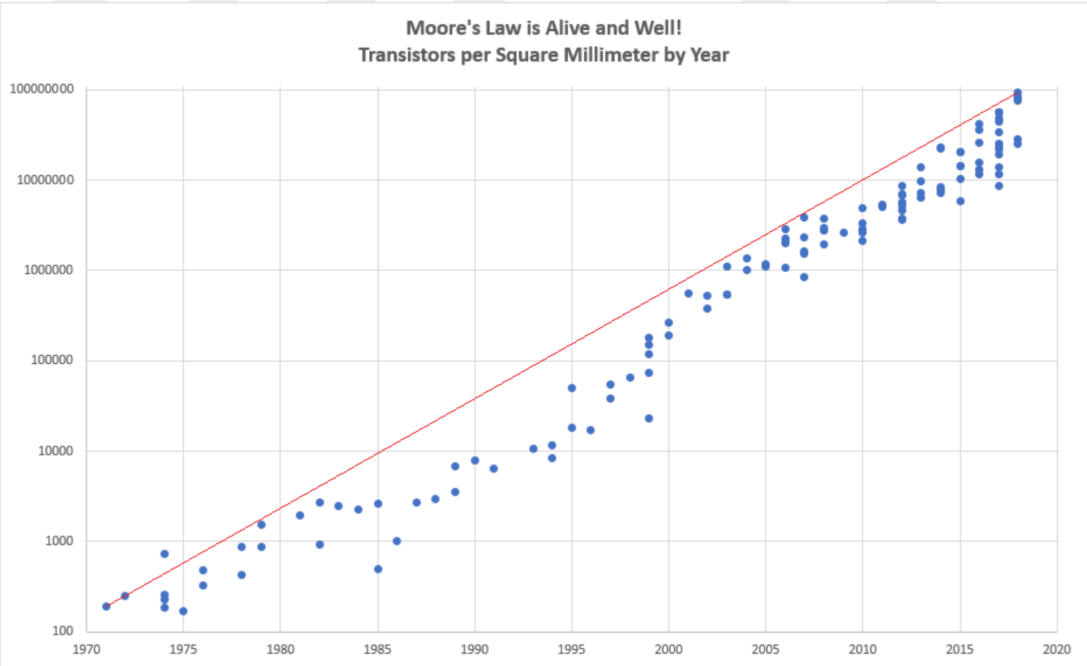

Figure 2.1 : Transistor Count per Microprocessor by Year, 1971–2018. Logarithmic scale. Data from Wikipedia. (Image credit: Eric MARTIN).

When we look at the history of innovation of the electronic and digital musical instruments in the 20th and 21st century, we can see that the sound generation was a driving factor up to the advent of capable consumer computing devices. Up to this time, while other factors were also of interest, a great emphasis on sound generation was present (Manning, 2013). We can extrapolate that, economic reasoning, such as making certain types of sounds – for example, the sounds of orchestral instruments – available to the public at a lower cost, was a strong motivation. One example is the famous Yamaha DX7 as seen in Figure 2.2. By employing FM Synthesis, the DX7 was able to create rich timbres on the cheap which was not previously possible (Chowning and Bristow, 1986). Up until the mid-2000s, a great number of hardware sound generators were on the market.

Figure 2.2 : Yamaha DX7 (Image credit: Vintagesynth Inc.).

Another focus on sound generators can be seen in the history of personal computing, gaming, and multimedia capabilities. While the personal computers lacked the computing and space requirements for high fidelity sound playback and generation, a great number of products in the form of soundcards with extended midi capabilities were produced, marketed, and sold. Again, this trend shifted when the personal computers gained the necessary storage speed and computing power to generate sound for the needs of the end-user.

After this technological shift, it was possible to generate, process, and control any complexity of sound with a common personal computer. This resulted in DMI inventors focusing more on controlling the sound rather than generating it (Jensenius et al, 2017).

Thanks to the virtually unlimited computing capabilities that arrived in the last decade, we can observe a fruitful collection of different music instruments, controllers, software, software development platforms, and so on.

2.1 Lutherie and Evaluation of VRMIs

In this section, we look at the lutherie and evaluation of VRMIs.

2.1.1 Traditional lutherie versus DMI lutherie

As discussed earlier, traditional lutherie and DMI lutherie had parallel interests and concerns on sound generation up until the availability of powerful and accessible computing. DMI luthiers are no longer under any limitations that arose from the media. Sound-wise, the only limitation for a DMI luthier is the methods and imagination whereas acoustic/traditional instrument luthiers are limited by physics and building material.

While innovation in terms of new instruments is still present in traditional lutherie, most of the efforts and focus are on enhancing or perfecting existing instruments sonically. On the other hand, because of the fact that a saturation in sound generation capabilities and diversity is reached, contemporary DMI lutherie delves into other areas of music performance and creation. This resulted in a wide variety of hardware and software musical instruments ranging from toy or game-like instruments to overly complex uncontrollable inventions that exceed the available human bandwidth.

VRMIs are a subset of DMIs. Just like DMIs, VRMIs have access to a virtually infinite amount of computing power, and sound generation can borrow any method or concept from DMIs.

As the sound generation possibilities are unhindered, VRMI lutherie should be more concerned with the control paradigms and the VR specific issues that will be discussed later.

2.1.2 Evaluation of DMIs and VRMIs

Existing research on DMI evaluation focuses mainly on the control methods as sound generation possibilities are endless and DMIs share the same concern with any new instrument – becoming playable. When the sound palette is infinitely customizable, what we are left with is the interaction with the instrument.

The interfacing and interaction with the instrument are detrimental to many aspects of an instrument such as the controllability, performability, learning curve, possibility of virtuosity, expression capabilities, difficulty, ludic activity, inspiration, audience perception, repeatability, producibility, adaptability, marketability, and so on.

With a VRMI, as the virtual performing space is also a part of the interface of the instrument, interface and interaction have even more effect on the performer and can be manipulated to change certain properties of the instrument and the users' experience with it.

2.1.2.1 DMI/VRMI evaluation from the viewpoint of HCI

One way to evaluate a VRMI is by looking at it from the viewpoint of classical human-computer interaction. In some cases, HCI evaluation methods such as Fitt’s law, Meyer’s law, and Steering Law can be employed while evaluating. Classical HCI can evaluate the control efficiency of a DMI/VRMI (Wanderley et al 2012). Especially with a VRMI, as possible controls include pointing, classical HCI metrics can be valuable.

On the other hand, easier task completion is not always preferable. Traditional instruments have varying levels of difficulty and the amount of difficulty, whether mental or physical can affect other aspects of an instrument. In some cases, even intentionally implemented difficulty can enhance an instrument (Sarier, 2012). The amount of difficulty has great impacts on ludic elements and kinesthetic feedback. Tanaka states that imperfections and limitations from the perspective of tool design often contribute to the character of an instrument (Tanaka, 2000). Therefore, evaluating DMIs/VRMIs from the point of view of classical HCI should be done with care.

2.1.2.2 Learnability, learning curve and progressability

Learnability and learning curve of an instrument is an important aspect and has a great role in a new instrument's chance in survival. The learning curve is a well-understood concept and can be systemically used to analyze any system that involves apprenticeship. Musical instruments are also systems where learning curves are applicable as apprenticeship is required (Jorda, 2004).

While having a relaxed learning curve can result in end users picking up the instrument faster, it can have the adverse effect of limiting the instrument's output diversity, reward, and capability. On the contrary, a steep learning curve can repel novice players and reward practice and apprenticeship with diverse output. Jorda illustrates this input/output complexity issue by giving the examples of a kazoo and piano and plotting the apprenticeship versus output diversity over time (Jorda, 2004).

> We intuitively grasp the learning curve concept; we know it can tell us a lot of information about the relation between a player and an instrument, on how the relation starts and evolves, but we do not know how to clearly define this concept, much less how to evaluate it. Could we, in order to compare different instruments, compare their learning curves? We can compare their learning curves shapes and tendencies, but we cannot evaluate their asymptotes or their inflexion points since we do not know the ordinates absolute values, not even what they represent. A serious attempt at defining the music instruments learning curve concept falls outside the pretension of this article, but that will not dismiss us from trying to manage the concept more intuitively. The piano has a steeper (i.e. better!) learning curve that the violin. The kalimba has an even steeper learning curve, although its asymptote may remain lower. The learning curve of the kazoo is almost a straight horizontal line, very low when compared to the other three asymptotes (since the abscissa of its ‘mastering point’ is very close to the time origin, and the ordinate is also close to cero given the reduced capabilities of the instrument. *(Jorda, 2004).*

Jorda also states that this trade-off also creates a trade-off between challenge and frustration and a balance is preferred.

Progressability of an instrument is the ceiling of apprenticeship. Commonly while steeper learning curves have a high ceiling of progressability, easier learning curves hit the ceiling early in the progression of practicing the instrument.

While the learning curve is an important aspect, when we look at the history of musical instruments, we can see that instruments with various levels of learning curve were able to survive the time and competition. This can show us that the learning curve is not the sole metric that determines an instrument's adoptability. An instrument can have a horrible learning curve and very basic input/output complexity, but it can still be a survivor due to its other rewarding properties. One example is an early electronic instrument, the Theremin.

However, this doesn’t mean that we can skip the learnability of an instrument as a design consideration In a VRMI design, the learning curve can be artificially modified towards less steepness by employing learning aids, performing aids, employing varying levels of selective difficulty, and gamification, while maintaining progressability. These will be later discussed in the dissertation.

2.1.2.3 Efficiency

As an extension to the learning curve, Jorda (Jorda, 2004) states that a musical instrument’s efficiency can be another metric for evaluation. He creates a model inspired by HCI which arrives at Human Efficiency by comparing Musical Output Complexity and Control Input Complexity. In this equation, when an instrument has little input complexity and high output complexity, it gathers a high efficiency. However, Jorda also gives the CD player as an example and tests it with the proposed model. CD player, having only one instruction, to start playing can be considered a low input complexity and high output complexity instrument. Jorda states that this creates a tricky situation and skews the proposed models' results and further expands the model to include performer freedom into the equation.

Although the efficiency can be a great model for traditional music instruments by comparing human effort and output in terms of volume and diversity, I find it hard to use the model, even as guidance while designing DMIs or VRMIs. History is again showing us that there are many instruments where the efficiency is low but said instruments survived for centuries up to this date and question the value of the metric.

2.1.2.4 Audience interaction

One of the biggest challenges of many DMIs is the diminishing perceived casual link between performance gestures and the system that produces the sound (A Framework for the Evaluation of Digital Musical Instruments). With performances of traditional acoustic instruments, there is a strong perceivable link between the actions of a performer and the sound generation mechanism of the instrument. Even with an unfamiliar acoustic instrument, the audience can figure out what is happening as acoustic instruments adhere to the same set of physics rules. However, with a DMI, the audience's ability to grasp the working and the logic of the instrument is not always great. Especially with "laptop" performers, where the interaction is mostly composed of staring at a screen and almost looks like daily computing, this loss of visual communication caused concerns amongst researchers as visual cues from the performer's gestures are an indicator of the performers' expressive and artistic intent. Several researchers found a positive link between performer's gestures and audience perception.

In VRMIs, there is room for easily perceptible bodily gestures. Even when the audience can't see the virtual instrument, depending on the scenario establishing a causal link is possible. One example is a VR drum set. In this scenario, the audience can easily deduct that the performer is performing on a virtual membrane instrument, and the interaction method is hitting on surfaces albeit the fact that surfaces are neither real nor present.

This point of view assumes that an audience is present. However, it is arguable whether an audience is a must for an instrument or not. In addition, there are already instances of music performances where the audience can't even see the performers (acousmatic music, orchestra pit, etc.).

2.2 History of Virtual Reality Technology

Virtual Reality is both fictionalized and materialized throughout history but the first commercially available technology that can create an acceptable immersion that VR promises happened on the so-called second wave of virtual reality.

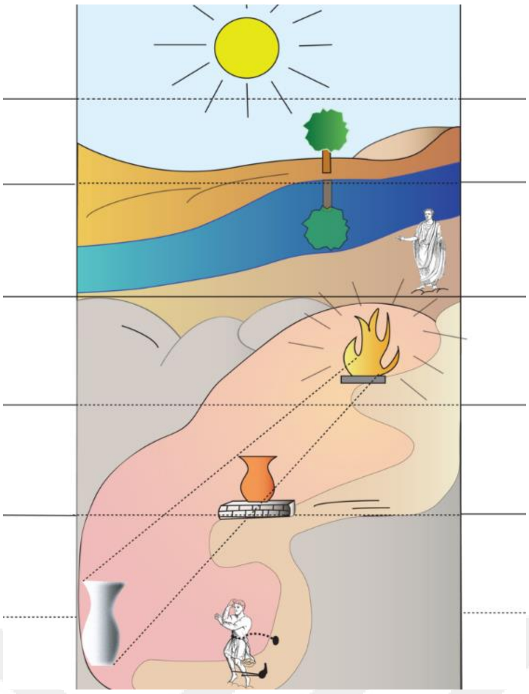

Figure 2.3 : Depiction of Plato’s Allegory of the Cave (Steinicke, 2018).

One of the first known fictional descriptions of virtual reality dates back to 360BC – to Plato’s Allegory of the Cave (Figure 2.3).

> In a dialogue between Plato’s mentor Socrates and his brother Glaucon, the former describes a dark scene in an underground cave, where a group of prisoners sit chained while they face the wall in front of them. Being in that position since birth, they can only see shadows that are projected on the wall. In addition to the chained prisoners, there are further people, which Plato called the “puppeteers.” These puppeteers are located behind the prisoners and cause the shadows by handling and interacting with objects in front of a fire. However, for the prisoners, who are unable to see puppeteers, these shadows would become reality. The allegory points out that the prisoners’ conception of reality is based on the limited view, which shows only an imperfect projection of the real world, defined by the puppeteers. (Steinicke, 2018, p. 20)

Daniel F. Galouye's Simulacron 3 (1964), Rainer W. Fassbinder's World on a Wire (1973), William Gibson's Neuromancer(1984), Brett Leonard's The Lawnmower Man (1992), Neal Stephenson's Snow Crash (1992), The Wachowski Brothers' The Matrix (1999), Vernor Vinge's Rainbows End (2006) and Ernest Cline's Ready Player One (2011) are notable examples to the VR science fiction.

Materializing of the virtual technology dates back to 1962 to Morton Heilig's Sensorama and any others tried their chances with creating devices that can create immersion by replacing the users view with a head-mounted display (Heilig, 1962).

I would like to list virtual reality technology in two different topics, early virtual reality history (before 2016) and the second wave of virtual reality.

2.2.1 History of early virtual reality platforms

The first half of the history of virtual reality platforms and devices is riddled with visionaries and attempts that were too ahead of their time. At their time, several technical requirements were impossible to be met and despite the excitement, none was able to acquire widespread influence.

2.2.1.1 Morton Heilig: Sensorama, 1962

Sensorama (as seen in Figure 2.4) is an invention of Morton Heilig, a true visionary that wanted to immerse all the senses of the viewer. Sensorama was an arcade type cabinet with a 3d wide display, sound, a vibrating seat, and a scent generator, capable of sensory immersion in more than just vision. Unfortunately, Heilig failed to commercialize the Sensorama and said that the sensorama may have been too revolutionary for its time. (Steinicke, 2018)

Figure 2.4 : Morton Heiling's Sensorama (Steinicke, 2018).

2.2.1.2 Charles Comeau and James Bryan: Headsight, 1961

The headsight (Figure 2.5) is an early HMD which is very close in principle to the HMDs of today. It used a CRT display element for each eye and employed a magnetic head tracking system. Its intended use was to create telepresence in dangerous situations and developed for military use. The head tracking was supposed to control a remote motorized camera, enabling users to look around in another environment (Steinicke, 2018).

Figure 2.5 : Charles Comeau and James Bryan's: Headsight (Steinicke, 2018).

2.2.1.3 Ivan Sutherland: Sword of Damocles, 1968

Sword of Damocles (Figure 2.6) was the first HMD to use computer-generated visuals and is considered to be the first HMD system of the like. It aimed to immerse the user in a virtual world composed of wireframe computer-generated graphics (Steinicke, 2018).

Figure 2.6 : Ivan Sutherland's Sword of Damocles (Steinicke, 2018).

2.2.1.4 Thomas A. Furness III: Super-Cockpit, 1968

Thomas Furness, the founder of the HIT Lab, developed display systems and VR based flight simulators for the U.S. Air Force (Figure 2.7).

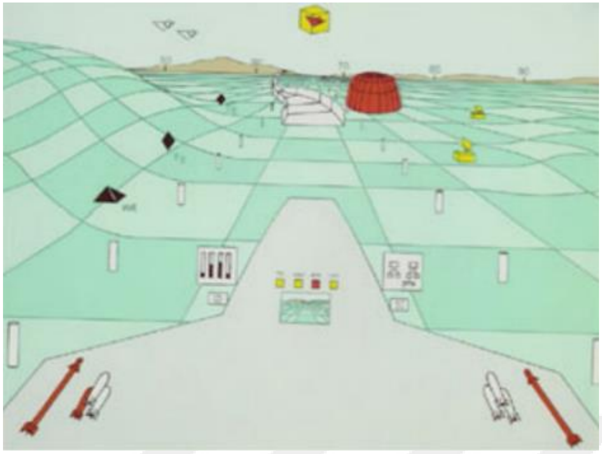

Figure 2.7 : Image from Super-Cockpit Project (Steinicke, 2018).

2.2.1.5 Jaron Lanier: Virtual Reality and VPL Research, 1985

Jaron Lanier, a co-founder of the Visual Programming Lab, coined the term Virtual Reality. VPL is a company responsible for building and selling various VR devices and it is the first company to sell VR goggles (Steinicke, 2018).

2.2.1.6 Jonathan D. Waldern: Virtuality, 1991

Virtuality (Figure 2.8) is a line of VR devices that was developed and sold by the Virtuality group. The said products were both aimed at professionals and gamers. Headset resolution was 276*372 pixels per eye and magnetic tracking technology was used. Software support and resolution were the main concerns of the end-user (Steinicke, 2018).

Figure 2.8 : VRS Virtuality Sit Down Unit (Image credit : VRS Corporation).

2.2.1.7 Nintendo: Virtual Boy, 1995

Virtual Boy (Figure 2.9) was Nintendo's attempt at commercializing VR technology. It was a tabletop console with a 3d stereoscopic display. Head tracking was not implemented over health concerns and liability. Virtual Boy is discussed as a huge financial failure (Steinicke, 2018).

Figure 2.9 : Nintendo Virtual Boy (Image credit: Nintendo Co., Ltd.).

2.2.2 Second wave of virtual reality

After the commercialization failures of VR in the 1990s, it took a lot of years and courage for initiatives to have a go at it again. After 2010, consumers had access to affordable GPUs that are able to run high-resolution displays at a high frame rate. Resolution and frame are key necessities for HMD based VR systems as will be discussed later thoroughly.

When these necessities were met, investors were still staying away from VR. The initial kick for the second wave of VR came from the public in the form of a kickstarter.com crowdfunding campaign. One of the first devices of this new era is the Oculus Rift DK1 (Development kit). The kit was presented on kickstarter.com and it was aimed at developers. The affordable price quickly gathered public interest, and the project was backed up not just by developers but end users. This interest of the end-users was extraordinary as there were no gaming titles or any other usage for the hardware at the time.

After the launch of the DK1, Oculus Rift DK2 was announced and end-user interest grew even higher as the hardware was still cheap, it was even more capable, and some of the developers with the DK1 were able to create content. Public curiosity was high and DK2 production couldn't meet. Some of the shipped DK2 started being resold at much higher places on online markets.

The enormous rise in the public interest was noticed by the investors which then funded the VR revolution, the second wave of VR. After the Rift, several companies started developing and selling VR systems.

2.2.2.1 Oculus Rift development kit (DK1)

Released in the March of 2013, Oculus Rift DK1 (Figure 2.10) featured a 7-inch LCD, capable of producing 640x480 pixels for an eye. The field of view was 90 degrees horizontal and 110 degrees vertical. The field of view was nearly twice of the previous offerings and one of the strongest selling points. Head tracking was done with a 6DoF IMU.

When we compare the resolution and the FOV of DK1, there is disproportionality. The resolution was not adequate for the wide FOV and it was criticized.

Figure 2.10 : Oculus Rift DK1 (Image credit : Facebook Technologies Ireland LTD.).

2.2.2.2 Oculus Rift DK2

After the initial DK1 kits were sold, Oculus announced the DK2. DK2 (Figure 2.11) featured a higher resolution (960x1080) with a low-persistence OLED display. The refresh rate was higher and supported positional tracking. Oculus Rift DK2 can be considered the first VR HMD headset that is adequate for a quality immersive experience from a technical point of view.

Figure 2.11 : Oculus Rift DK2 teardown (Image credit: iFixit Co., LTD.).

2.2.2.3 Oculus Rift CV1

Oculus Rift CV1 (Figure 2.12) is a descendant of the developer kits and it is the first consumer unit of the family. It features a greater resolution and many other refinements. However, this time CV1 was not alone in the marketplace as several other manufacturers were also able to develop and market other systems.

CV1 uses the Constellation system for tracking its headset and controllers. Externally placed IR sensors detect the light emitted from the tracked objects.

Figure 2.12 : Oculus Rift CV1 with its peripherals (Image credit : Facebook Technologies Ireland LTD.).

2.2.2.4 Steam VR and HTC Vive

HTC Vive (Figure 2.13) was a direct competitor to the Oculus Rift series. Unveiled in March 2015 and begin shipping in June 2016, HTC Vive had several improvements over the Oculus Rift Dk2.

The resolution per eye of the Vive was 1080x1200 and had a 90hz refresh rate with a 110-degree FOV. Apart from small technical improvements, a major difference between the Oculus DK2 and the rift was the tracking system employed.

Figure 2.13 : HTC Vive headset, lighthouses, and controllers (Image Credit : Valve Corporation).

Vive used the SteamVR Tracking, which uses infrared laser turrets that scan the room with invisible laser beams. Every trackable object has photoresistive surfaces on them and the laser beams shining on the photoresistive surfaces are used to calculate the orientation and the position of the object. When two laser turrets are used (the system shipped with two), the tracking system was able to track objects in a 5x5m space to a sub-millimeter accuracy with an unnoticeable delay. The tracking system and the overall system latency was so low that, when a controller was thrown at the user wearing the HMD, the wearer was able to catch the controller mid-air just by seeing the 3d representation of the controller through the HMD.

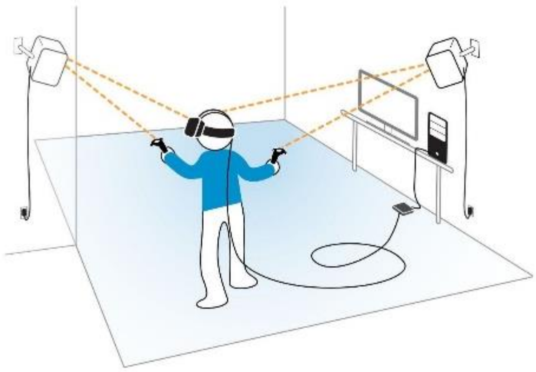

Figure 2.14 : Depiction of HTC Vive play area. (Image Credit : Valve Corporation).

Compared to rift's minimal tracking capabilities, this system allowed room-scale gaming. As the system could track the user in the "play area", users were not constrained to a single sitting position (Figure 2.14). This became a huge selling point albeit the higher price.

2.2.2.5 Google Cardboard, Samsung Gear VR

During the years, smartphone technology also came far and the said devices acquired high-resolution displays with high pixel density. It is found out that with lenses and an enclosure, smartphones could be worn as headsets and supply a somewhat satisfactory VR experience.

Figure 2.15 : Cardboard VR (Image credit: Google LLC).

Google cardboard (Figure 2.15) and many passive systems derived from the idea are generic headsets that accept a smartphone as the display and the computer. Samsung's Gear VR family (Figure 2.16) is also bred from the same idea, but they are designed for specific smartphones from the manufacturer and offer additional features like remote control and extra IMUs.

Figure 2.16 : Samsung's Gear VR (Image credit: Samsung Electronics Co., LTD.).

Overall, the smartphone experience is more limited than dedicated VR headsets. The first downside is the resolution and optics as they can't deliver the same performance as their dedicated counterparts. The other limiting factor is the system performance as these headsets rely on the mobile CPU and GPU, which are relatively weak than personal computers. This results in the inability to draw high resolution, real-time, rich graphics.

However, a great percentage of the VR users around the world is smartphone-based system users. While they can't provide the same level of tracking or GPU capability, they are adequate for media playback such as 2x180 VR footage of various materials. The video playback performance is also sub-par from the dedicated systems as these devices are resolution limited and available computation power currently can't decode HVEC video content. Nevertheless, the amazingly low entry cost gets ahead and put these systems ahead.

2.2.2.6 Playstation VR

Playstation VR (Figure 2.17) is a VR HMD system developed by Sony for their popular gaming console, Playstation 4. Compared to the Oculus Rift CV1 and HTC Vive, PSVR has a slightly worse resolution per eye. It is also hindered by the game console, as the console is designed to run games at 60 fps, whereas in VR, a higher frame rate is preferred due to motion sickness. Sony tries to avoid this by using interpolation to create 120 fps from the output of the 60-fps console.

Figure 2.17 : PlayStation VR System with its camera and controllers (Image credit: Sony Corporation).

PSVR also uses a less precise tracking system, employing a high frame rate camera. This system also fails to support room-scale tracking.

Despite its shortcomings, PSVR has a strong marketplace with 5 million units sold globally due to the low level of entry cost. While the Oculus Rift and HTC Vive require powerful personal computers for a satisfactory experience, PSVR can run with the relatively cheap Playstation 4 console.

2.2.2.7 Standalone devices with limited functionality

Recent years also showed a new breed of untethered headsets, mainly geared towards media playback. Oculus GO, Google Daydream, LG Mirage are some of the examples. These devices have built-in smartphone derived computing components and displays. They neither require a pc or a smartphone to run. With built-in high-resolution displays, they are initially geared towards newcomers to VR, or media playback. They also support VR gaming on a quality scale comparable to smartphone-based systems. Motivations behind the purchase of these devices are expected to be mostly viewing adult content (Roettgers, 2018).

2.3 Virtual Reality Music Instruments

VRMIs created to this date fall under two categories. The first category is academic, research, experimental or artistic/installation instruments that have no or little intention to reach the masses, and the second category is already marketed software or hardware that reached consumers.

2.3.1 Academic, research, experimental or artistic/installation VRMIs

This section contains examples of academic, research, experimental or artistic/installation VRMIs

2.3.1.1 ChromaChord

ChromaChord (Figure 2.18) is a virtual reality music instrument that runs on Oculus Rift and employs leap motion for hand tracking.

Users control the software through simple virtual graphics and the sound is generated by Max/MSP software that runs parallel to the VR software. It does real-time synthesis. The software is not publicly available for download (Url-4).

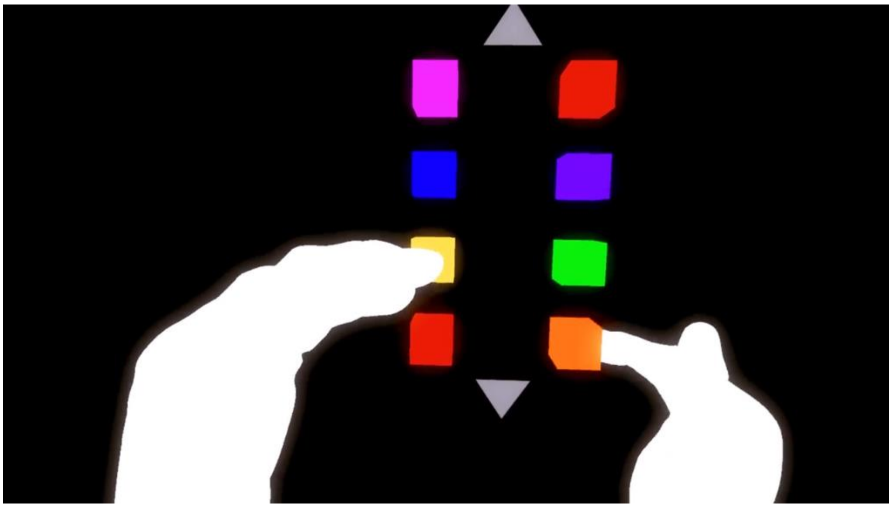

Figure 2.18 : Screenshot from Chroma Chord.

2.3.1.2 Wedge

Wedge (Figure 2.19) is a virtual reality music instrument that runs on Oculus Rift and employs leap motion for hand tracking. Wedge uses Unity for graphics and FluidSynth software for sound generation. The software is not publicly available for download (Moore et al, 2015).

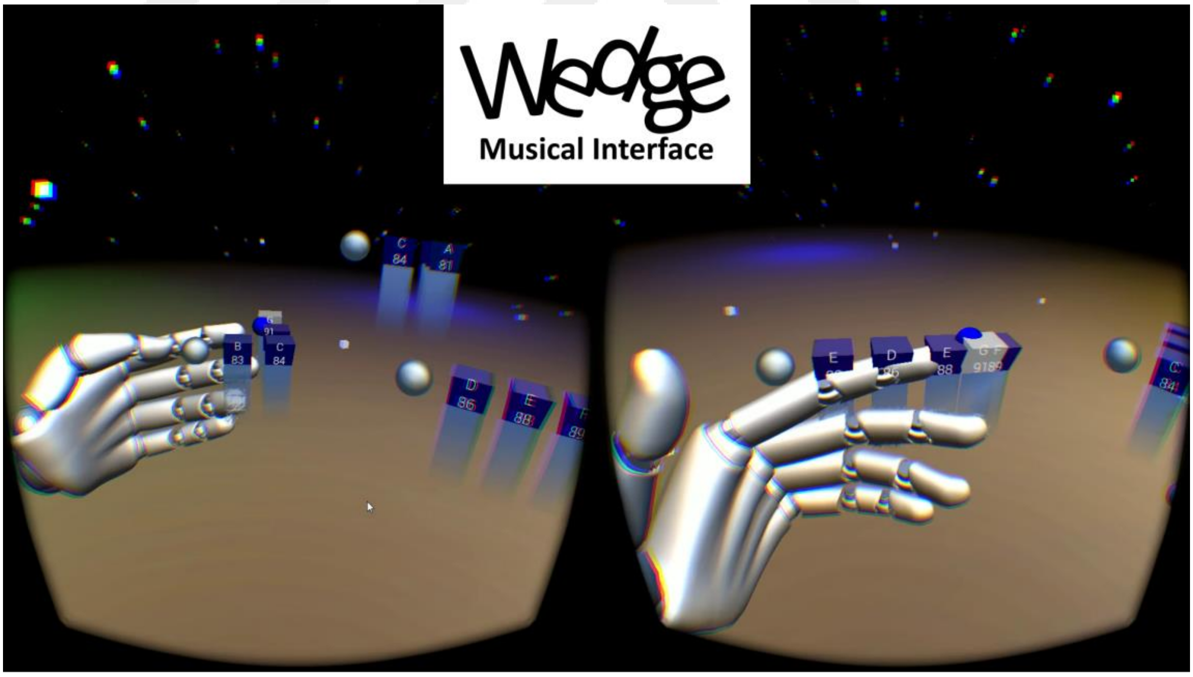

Figure 2.19 : Screen capture from Wedge.

2.3.1.3 Coretet

Coretet (Figure 2.20) is a collection of four virtual instruments somewhat resembling traditional acoustic instruments. While a Pure Data server is responsible for synthesis, Unreal Engine is responsible for the rest of the execution. The instrument is not publicly available for download (Url-5).

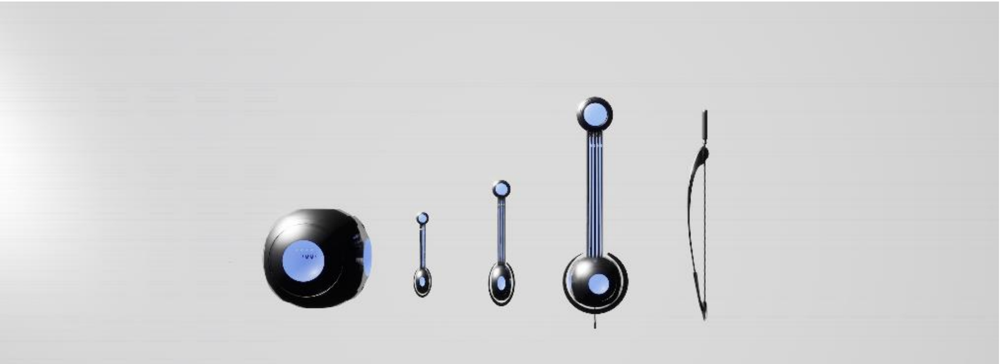

Figure 2.20 : Coretet virtual instruments (Url-5).

2.3.1.4 VIVR

VIVR (Figure 2.21) is a VRMI that uses HTC Vive as a headset, Unity as the main engine, and SuperCollider as the synthesis engine. While some of the code snippets are available for download, the whole instrument is not publicly available(Sánchez, 2018).

Figure 2.21 : Screenshot from Vivr performance.

2.3.2 Published VRMIs

Published VRMIs are VRMIs that are accessible to third parties. This section discusses a selection of them.

2.3.2.1 EXA: The Infinite Instrument

EXA: The Infinite Instrument (Figure 2.22) is a VRMI where the users can create their own instrument by drawing and placing various control surfaces of their own design. The interaction is done via hand controllers of a VR set and triggering sounds is mostly percussive in its nature. The instrument also has programmable functions like mini sequencers and triggers. The performing space is an abstract stage with fixed dimensions and futuristic light shows on the walls.

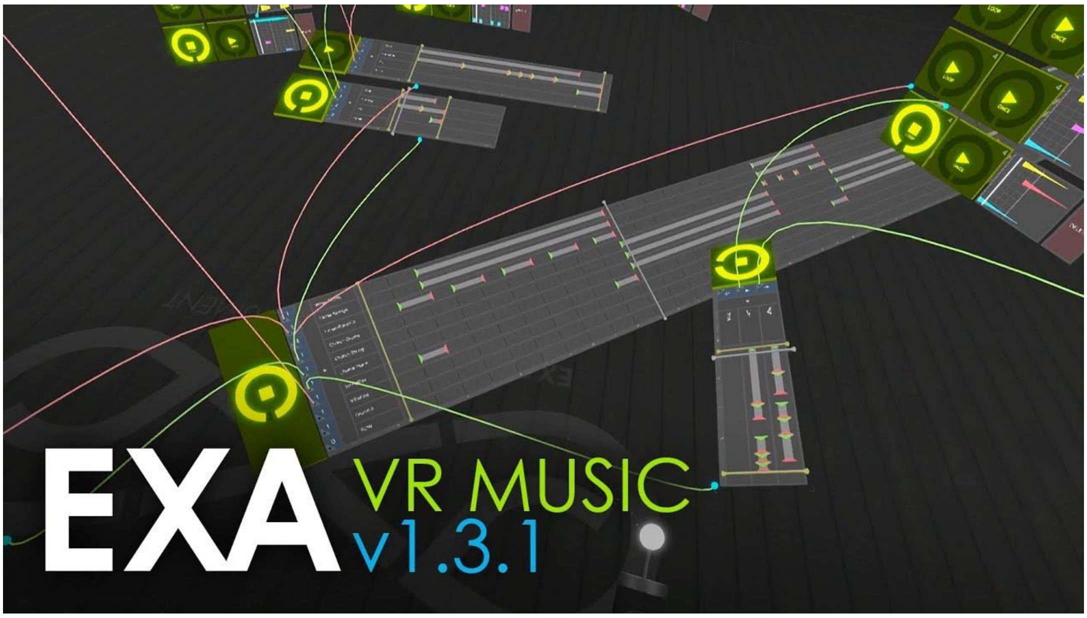

Figure 2.22 : EXA VR Promotional Screenshot (Url-7).

EXA produces sounds by playing back samples either from an SF2 soundfont format, or an audio container (WAV, OGG, MP3, etc). EXA doesn't have any synthesis.

The user feedback is mostly positive with only one negative feedback addressed to an early release of the software. The negative reviewer complains about the locomotion system and UI issues (Url-7).

2.3.2.2 Paradiddle

Paradiddle (Figure 2.23) is a virtual reality drumming simulator. Interaction is done with hand controllers. As a drumming simulator, triggering is percussive in its nature. Users are allowed to customize their drumset by picking and placing various elements in the playing space. The performing space is a space platform with stars and planets surrounding the performer with a few rings drawn on the floor to signify it.

Sound generation is done via sample playback and there is no synthesis.

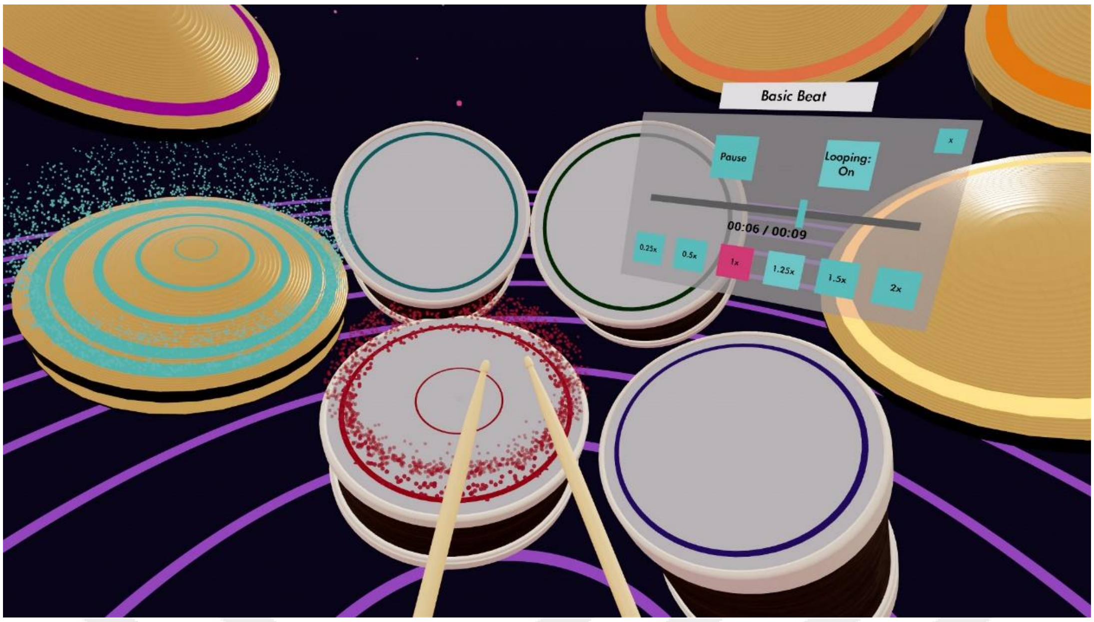

Figure 2.23 : Screenshot from Paradiddle (Url-8).

The user feedback is mostly positive but Paradiddle has 8 negative reviews. Most of the negative reviews focus on the interaction being poor and not fun. 2 users specifically address their issues with interaction as the virtual drumsticks clipping with the virtual drums. One user talks about his preference over a different VR drumming software, Drum Master VR, which is more responsive according to him (Url-8).

2.3.2.3 DrumMaster VR

DrumMaster VR (Figure 2.24) is another drumming simulator. Compared to the Paradiddle, DrumMaster VR's drawn elements are much more photorealistic. This photorealism is also accompanied by dynamic lightning that is user switchable. Users are able to dim the global lighting to their taste. The performing space is a stone platform on top of a mountain, surrounded by grass and other mountains in the distance. Cymbals of the VR drumkit adhere to physics and start moving in a realistic fashion as they are hit. All of the elements of the drumkit can be position and height adjusted. When the drum parts are hit with the virtual drum stick, a spark animation happens on the hitpoint giving the users visual feedback.

The DrumMaster VR uses audio playback to generate sound and doesn't employ any synthesis.

Figure 2.24 : Drum Master VR promotional Screenshot (Url-9).

DrumMaster has mixed reviews. While some users find the experience positive mostly because of the quality graphics, some find the experience negative. One negative reviewer complains about the lack of different sounds, one reviewer complains about very low fps and the other two complain about the virtual drumsticks not bouncing from the virtual drums, stating that the experience is sub-par to real drumming. From the reviewers' testimonies, it can be understood that the negative reviewers have real drumming experience while the positive reviewers are not real-world drummers (Url-9).

2.3.2.4 Percussive VR

Percussive VR (Figure 2.25) is a VRMI that imitates four different traditional percussion instruments, two marimbas, a glockenspiel, and a steel pan. Interaction is done via hand controllers and percussive in its nature. The performing space is a big hall with a wooden floor reminiscent of a practice space or a large depot. The instruments are modeled in a realistic way but the models are not highly detailed.

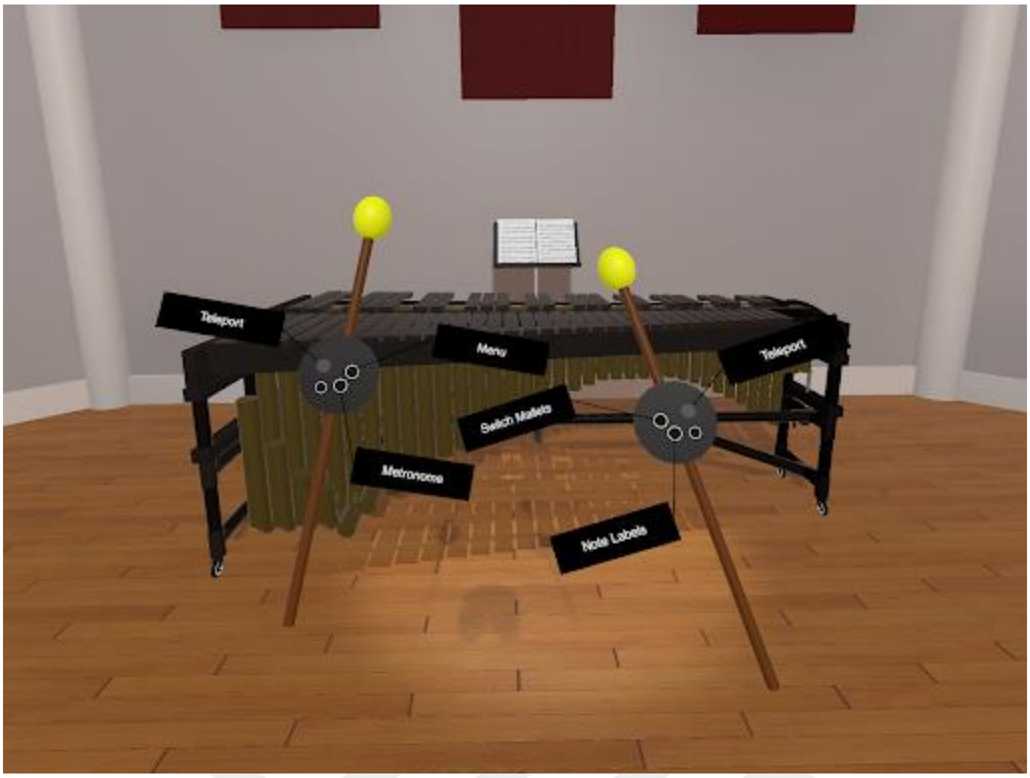

Figure 2.25 : Screenshot from Percussive VR (Url-10).

Percussive VR generates sounds via audio playback and doesn't have any synthesis engine.

Reviews of the software are mostly negative. Lag, hitbox detection, and general performance are of concern of the negative reviews. One reviewer also complains about the virtual mallets being offset to their virtual headsets controllers. Again, the missing haptic feedback, mallets bouncing from the instrument is noted as a negative by reviewers (Url-10).

3. KEY CONCEPTS IN VIRTUAL REALITY

Due to its immersive nature and its control over human senses, virtual reality brings new paradigms to computing and DMI lutherie. Therefore, it is of utmost importance to understand the key concepts of virtual reality before developing for it.

3.1 Definition of Virtual Reality

When looked at from a linguistic perspective, the term virtual reality constitutes an oxymoron as the word virtual contradicts reality. In the background section, we covered that Jaron Lanier from the VPL invented the term virtual reality. In his context, virtual reality was used to describe the immersion experience created via. HMD VR system.

Before moving on to the key concepts of VR, it is important to define what virtual reality means in this context as the term is used interchangeably for other types of experiences commonly.

In the scope and context of this thesis, the term Virtual Reality refers to a system with three features. Steinicke's definition of virtual reality perfectly describes these three features:

> Virtual Reality (VR) requires three real features: (i) real-time rendering with viewpoint changes as head moves, (ii) real space, i.e., either concrete or abstract 3D virtual environments, and (iii) real interaction, i.e., possible direct manipulation of virtual objects (Steinicke, 2018).

In addition to the three features, an outcome of an adequately working VR system, the feeling of presence is also should be present with Virtual Reality for the purposes of this dissertation. VR pioneer Abrash describes the feeling of presence with his own words as follows:

> Presence is when, even though you know you're in a demo room and there's nothing really there, you can't help reaching out to try to touch a cube; when you automatically duck your head to avoid a pipe dangling from the ceiling; when you feel uneasy because there's a huge block hanging over you; when you're unwilling to step off a ledge. It's taking off the head-mounted display and being disoriented to find the real world there. It's more than just looking at someplace interesting; it's flipping the switch that makes you believe, deep in your lizard brain, that you are someplace interesting. Presence is one of the most powerful experiences you can have outside reality, precisely because it operates by engaging you along many of the same channels as reality. For many people, presence is simply magic (Abrash, 2014).

To summarize, the definition of a VR System for this dissertation is that VR Systems are systems with which a user can be decoupled from the real world at least by sight by projecting a different virtual one, with the possibility of 1:1 interaction of head and body movements in the said virtual world in order to create a strong feeling of presence.

3.2 Augmented Reality and Differences with Virtual Reality

While augmented reality uses similar technology and tools, it differs from virtual reality as it doesn't seek presence. Instead of immersion and presence, AR uses the world around its user and doesn't attempt at a decoupling.

In augmented reality, virtual reality is projected on the real world around the user for various reasons. Augmented reality doesn't require a headset, even a small portal such as a smartphone screen can be used as a portal to the overlaid elements as immersion and presence is not a concern.

3.3 Technical Requirements of a Virtual Reality System

In order to achieve the level of immersion sufficient for the feeling of immersion, certain technical requirements need to be fulfilled.

Thanks to the years of research behind virtual reality technology, we now have an understanding of these technical requirements. During the Steam Dev Days in 2014, Michael Abrash tells us that a convincible immersion is the result of convincing human perception that operates on a low level:

> At Valve, we've spent a lot of time investigating the factors that affect how people experience virtual reality. Most critically, we've learned that the key to making the experience unique and compelling is convincing perceptual systems that operate at a low level - well below conscious awareness - that they're perceiving reality (Abrash, 2014).

Due to this, the feeling of presence is fragile and can easily be broken by anything unfamiliar or unnatural to the low-level perception. With a VR system, steady conditions like low-resolution, narrow FOV, long pixel-persistence, or low frame rate blocks the feeling of presence from occurring at all. On the other hand, momentary glitches like frame drops, mismatch of real and virtual movements, loss or jump of tracking, and freezes destroy an established presence. When an established presence is regularly disturbed, the user's confidence drops, and establishing a presence again becomes harder.

For an interruption-free VR experience with a strong feeling of presence, the system must have adequate specifications and solid performance.

3.3.1 HMD and vision

A head-mounted display is the first gateway to the virtual environment. It is the part that decouples the real world from the virtual one. It should be light and comfortable, enough to be ubiquitous to the user. It should also block all the light coming from the outside.

The main function of the HMD is to display the virtual environment to the user. The display system must have adequate specifications.

3.3.1.1 Resolution

Resolution is very important in VR. A 1080p computer monitor might be adequate for many people because of the fact that a computer monitor only fills a fraction of our sight. In VR, the available resolution is stretched over the whole FOV. The display panel is placed centimeters apart from the users' eyes. Because of this, even individual pixels can be identified if the resolution is low enough.

With a low resolution, the VR experience suffers and the feeling of presence becomes impossible for most users. When we look at the history of the VR system, the feeling of presence only became possible after HMD's achieved more than 1000x1000 pixels of resolution per eye.

Even after achieving presence, complaints on resolution are still being received. One common complaint on current HMD resolution is the screen door effect that the users are reporting. We can deduce that there is still room for better displays from the reports.

3.3.1.2 Refresh rate

The refresh rate is another aspect of a VR system. Because of the fact that the users' head movements must be reflected in the projection of the virtual world lag-free as possible, delays between the frames is important. When we analyze VR systems present, 70 fps looks like the minimum to achieve presence, while additional fps strengthens the illusion.

Apart from the refresh rate of the display, the hardware and the software that it runs must be able to run above the said fps. Software developers must not exceed contemporary GPU capabilities in order to do so and take them into consideration.

3.3.1.3 IPD

IPD is the abbreviation for inter pupil distance. It is the distance between the pupils of the eyes of the user. This distance differs from person to person and HMDs offer physically changing the IPD of the headset. Usually, by a screw mechanism, the lenses and the screens behind are moved close together or apart.

The IPD of the content projected also needs to match the user's IPD. Depending on the mismatch, objects can look wider or narrower than their intended dimensions. When the IPD of a headset is adjusted, ideally the virtual cameras in a scene should replicate the adjustment.

3.3.1.4 Head-mounting

HMD should firmly fit the user's head and stay fixed. As it is possible for the activity of the user to have big motions, the fitment must endure such forces. While being fitted, it should be comfortable and light enough not to hinder the user and be ubiquitous as possible. Figure 2.26 shows a contraption for fixing a headset to the user's head.

Another problem for the HMD is the cord that connects it to a computer (if present). In room-scale scenarios where the user freely moves in an allowed space, this cable gets in the way of the user and breaks the experience.

Figure 2.26 : HTC Vive’s adjustable strap system (Image Credit: Valve Corporation).

Recently, wireless communication systems became available for HMDs such as HTC Vive, they free the user from the tether at the expense of cost and weight.

3.3.1.5 Optics

Optics are the connection between the eyes and the display. The quality of the optic determines the fov and the clearness of peripheral vision. Current offerings are clear in the center and not so on the outside. It is human tendency to move our eyes to look at something in our sight instead of moving our head and staring dead forward. Figure 2.27 shows a close-up photo of HTC Vive’s lenses.

Figure 2.27 : Close-up of HTC Vive’s lenses (Image credit: Valve Corporation).

Because of the state of the lenses, users report problems with eye movement. In my own experience, this phenomenon changed my habits and I learned not to use eye movements to look around.

3.3.1.6 Generation of graphics

For a satisfactory VR experience, software developers must find a balance between good performance and good graphics. VR needs greater resolution than software that runs on a screen because of the reasons explained above. This puts a strain on the GPU. After the introduction of VR systems in the second wave of VR, several new generations of GPUs become available.

This resulted in some developers pushing the graphics a little further. However, to ensure compatibility with more users, the common current trend in VR graphics is to use a low-poly language. This is especially true for mobile VR applications as the mobile platform is much more limited in terms of graphical computing power.

3.3.2 Tracking

As discussed earlier, tracking employed in a VR System is crucial. The polling rate must be high, the system must be linear and precise, and it must allow freedom to the user for the best experience.

The polling/update rate affects the controller lag. In a VR case, HMD is also a controller and directly to the camera. Even the slightest lag creates a mismatch between the users' movements and the virtual camera position and results in motion sickness/nausea. It also affects the users' ability to perform timely functions.

Any non-linearity in the system distorts the users' perception of the space in a virtual environment. Ideally, a real movement must translate to the virtual world on a 1:1 scale. Non-linearity distorts this relationship and has direct effects on the immersion and presence effect.

Finally, the system should allow the user 360-degree freedom. If the tracking works when the user is facing one way in the real room and doesn't work well when facing the other side, the usability of the system is seriously hindered.

3.3.2.1 IMU and internal tracking

Some HMD systems rely on internal measuring units. IMUs employ accelerometers and gyroscopes to detect motion and orientation. IMUs enable tracking head motions to a degree, but they can't be used to track the absolute position of an object in space. Smartphone systems use IMU-based internal tracking systems.

3.3.2.2 Outside tracking

Outside tracking systems use external devices to track the position of trackable objects position such as the HMD and hand controllers on a typical VR System.

HTC Vive's SteamVR system, where base stations that scan the room with IR rays are employed, Oculus Rift's Constellation system where IR sensors that capture the light emitted from the trackable objects are employed, or Playstation VR system where a camera is employed are examples to outside tracking.

3.3.2.3 Inside-Out tracking

As shown in figure 2.28, inside out tracking is where the HMD employs technologies to track its own position from the cues around its environment. Google Daydream or Lenovo Mirage are two examples of HMDs that use inside-out tracking. The employed system is called Worldsense, and it uses outside-facing fish-eye cameras to recognize the environment around with computer vision and deducts the device's position in space from this deduction.

Figure 2.28 : Inside out tracking in action (Machkovech, 2018).

Worldsenses reported accuracy is sub-centimeter while SteamVR reports sub-millimeter accuracy. Inside-out tracking is clearly lacking in performance but it eliminates the necessity to set up external equipment to function.

3.3.2.4 Room-scale tracking

Room-scale tracking is a term coined by the Valve corporation within their SteamVR tracking system. This system allows the user to set a safe space, and freely move in the said space during usage. This allows natural locomotion and freedom of movement.

Several other bodily gestures like crouching, jumping, leaning, and moving the controllers behind the user are also better tracked in room-scale systems as these systems employ more than one outside tracking station to cover the whole space and directionality.

While room-scale tracking allows natural locomotion, it is limited by the clear space and systems maximum support and it is not often alone sufficient. Because of this, usually other modes of locomotion are bundled too with room-scale tracking.

3.3.2.5 Controller tracking

The IMU-based, inside-out, and outside tracking strategies are also present for handheld controllers. The quality of the tracking system directly affects interaction quality. As will be discussed later, this has a direct influence on VRMI design.

3.3.2.6 Hand tracking

Currently, commercial VR headset offerings do not allow the tracking of the users' hands. While the built-in technology of hand tracking is currently unavailable, Oculus announced that their next-generation system will feature hand tracking, freeing the user from using bulky controllers (Url-12).

Figure 2.29 : LeapMotion controller connected to a laptop (Url-11).

Although there are no commercial VR systems with built-in hand tracking features, headset developers allowed and encouraged the use of add-on systems. The most notable of these add-ons is the LeapMotion hand tracking system.

As seen in figure 2.29, LeapMotion is a combination of a stereo infrared camera setup and accompanying software for decoding the captured stereographic images into hand positions. LeapMotion device was developed to be a desktop device, but in VR usage, users have to mount the device on their headset, cameras facing where the user gazes at. Because of the narrow field of view of the camera system employed, the tracking space is very limited. When the user’s hands disappear from the sight of the cameras, tracking is lost (Url-11).

We can expect hand tracking from next generations of VR headsets, but most probably they will use videographic tracking methods which will be sub-par compared to hand controllers in terms of update rate, accuracy, and tracking space.

3.3.3 Locomotion

Locomotion in VR is a very important and special topic. In VR, users are met with an alternate presence in a virtual world and this feeling of presence urges the users to move and explore the world. However, enabling locomotion while maintaining the immersion and the feeling of presence in a VR scenario is encumbered with limitations and requires special care. The most common modes of locomotion in VR are listed below.

3.3.3.1 No-locomotion

No-locomotion is the scenario where the user's avatar stands still in the virtual world. Another terminology for no-locomotion is seated play. Users are expected to sit down or stay in one place while using the software.

In the most rigid scenario, users are able to look around but they are not able to move even the head of the virtual avatar. The only available locomotion interaction is the ability to look around with a fixed anchor. This system behavior is mostly associated with fixed media viewing such as 360-degree videos or double 180-degree stereo videos. As the position of the camera is predetermined while recording, users must observe the recorded material from the same point as the camera as there is no way to render the footage from another point of view. This rigid scenario is also present with a system that doesn't employ proper head tracking.

A less restrictive version of this no-locomotion, seated play scenario is in systems and software that allow proper head-tracking, even with limited space. Albeit being limited, this allows establishing a high degree of immersion where the software scenario doesn't necessitate locomotion, or the user's avatar is transported with a vessel. A shooter game, where the player stands still and shoots at incoming enemies, or a dj'ing simulation where the users are position right behind virtual decks can be examples of seated play where locomotion is not necessitated. A racing or flight simulation can be examples of where the locomotion is supplied with a virtual vessel.

3.3.3.2 Directional movement with controller input

In scenarios in which it is necessary for the user to move in the virtual world, one method to enable locomotion is to enable directional movement. This can be done with any kind of input such as directional buttons, joysticks, tracked controllers, etc. While this is a common and unproblematic method for locomotion in non-VR, screen-based 3d applications and games, this type of locomotion is very problematic for VR.

The very first problem comes from how humans locomote in the real world. While humans walk, our method of walking moves our heads and therefore our eyes also in the vertical axis, creating an up and down movement in every step. When this natural up and down movement is not employed, direct locomotion in VR feels unnatural, almost like the movement of a floating ghost. When this movement is artificially implemented, it creates an almost instant movement sickness in most of the users as the vertical movement is observed, but not registered with the inner ear, as there is no physical movement.

The second problem comes from the low-level perceptual systems of humankind. When we locomote, our inner ear registers coherent data, just like an accelerometer and a gyroscope combined. When moving with controller input, these coherent data is missing and it creates an uneasy feeling which results in cybersickness (which will be discussed later) in most users. The chance of cybersickness occurring is increased much more when walking animation that mimics the subtle head movements of human walking is employed.

3.3.3.3 Room scale locomotion

Room-scale locomotion is where the player is allowed and tracked in a space where physical modes of locomotion are possible. The advantage of room-scale locomotion is that the users are able to employ natural locomotion, such as walking. Contrary to locomotion with controller input, cybersickness is greatly reduced due to real movement and their registration via the inner ear. The user's movement is 1:1 reflected in the virtual world.

The downside of room-scale locomotion is that the amount of natural locomotion possible is limited to the amount of real free and trackable space around the user. For example, the base HTC Vive VR headset supports a 5m x5m trackable space, but it is also hindered by the amount of clear space the end-user can provide. Wired headsets also create another hindrance as the available locomotion gets limited also by the length of the cabling to the headset from the computer it is attached to.

Due to the above downside and limitation, most software and video games that support room-scale locomotion also employ other modes of locomotion if the users are expected to locomote greater than the common play area.

3.3.3.4 Teleportation

Teleportation is another way of locomotion in VR. With locomotion that employs teleportation, the users are given the choice to highlight an area in the virtual world where their avatar will be teleported (Figure 2.30). This allows players to navigate a virtual world, greater than the room-scale tracking available (if any) and free of the cybersickness that is caused by directional movement via controller input.

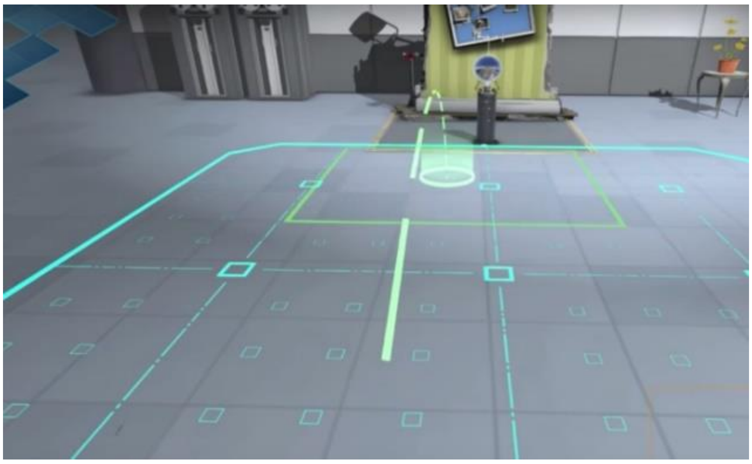

Figure 2.30 : Teleportation control's in the game titled "The Lab" (Image Credit: Valve Corporation).

While it doesn't cause cybersickness, locomotion via teleportation is also perceived as unnatural and results in confusion, and hinders the feeling of presence. Therefore, it is employed in scenarios where it is the only choice, or in scenarios in which it is combined with other means of locomotion such as room-scale locomotion.

3.3.3.5 Alternative means of locomotion

Apart from the listed modes of locomotion, alternative means of locomotion are also present and new models are being researched as all the listed modes of locomotion have shortcomings.

Infinite walking platforms, such as the Omni by Virtuix function like omnidirectional treadmills, allowing the user to employ natural locomotion, unlimited by the available space around (Figure 2.31). While these platforms solve most of the problems of locomotion, they are costly and not targeted at the average end-user.

Figure 2.31 : Virtuix Omni (Url-14).

Another alternative method of locomotion in VR is distance compression. While natural room-scale translates the real worlds in a 1:1 scale to the virtual world, distance compression changes the scale of the translation and returns bigger movement in the virtual world, therefore enabling the user to locomote beyond the available real-world space. As the movements are no longer translated exactly, this also becomes a source of cybersickness.

Cheap locomotion systems that can be marketed to home users are also being researched. WalkOVR is an example where several sensors are employed to capture stepping motion, and the captured data is then used to move the avatar of the user in the virtual world directionally. As the user is mimicking the walking motion in the real world, the vertical head movements are naturally occurring.

3.3.4 Interaction and input

Interaction and input in VR are accomplished with a variety of methods where it is necessary. While entirely passive VR experiences, such as viewing a 360-degree video are possible, it is almost guaranteed that even the most basic modes of interaction enhance the user experience. For example, in a passive video playback scenario, a method of input and interaction allows the player to start the playback, pause, or control various parameters of the sad playback.

3.3.4.1 Usage of the existing input devices

This method of interaction employs the usage of existing input devices such as a computer keyboard, mouse, gamepad, or similar device. In this method, these devices can be used as both pointing devices and direct inputs.

The main problem with this method results from the fact that existing controllers are not tracked and their 3d representation can’t be drawn onto the user’s vision. As the VR headset blocks the user's view from the real world, the user no longer has the ability to locate or navigate such devices with vision. Another problem arises from the fact that some of the existing input devices are either wired or they necessitate a surface such as a table to work on (keyboard and mouse for example), limiting the VR experience to a seated one.

Depending on the type of usage, the hindrance this creates ranges from huge to virtually none. For example, asking users to type on a keyboard that they can’t see creates a big hindrance. On the other hand, using a steering wheel for control input on a driving simulation creates little to no hindrance as the nature of the experience requires seated play, the steering wheel is in a fixed position, it's 3d representation can be calculated from the input and the controls are easy to find without vision due to their size.

3.3.4.2 Gaze and stare

Gaze and stare is a simple input method, mostly used when there is no alternative. Smartphone VR usage with passive VR Headsets like the Google Cardboard is an example. In this kind of usage, the user's smartphone is enclosed in the headset and both the buttons and the touchscreen can not be used for input. Gaze and stare become a viable option when there is no option for physical input.

In gaze and stare interaction, users interact with a GUI by directly gazing at menu items or a 3d objects in order to highlight them and staring at the selection until a timer runs out for confirmation of the decision.

As discussed, this is a very crude interaction scheme but it can be employed without any hardware other than a head tracking headset. This scheme of interaction can also be seen in use as an accompaniment to various other input methods as head tracking is a viable method of pointing in a 3d space.

3.3.4.3 Head movements as confirmation

This method of interaction employs gesture recognition of the head movements to acquire binary yes or no input. In this method of interaction, users are asked to nod their heads in specific ways for a binary choice of yes or no. This is method is also a crude one and mostly used when there is no alternative.

3.3.4.4 Single-button

Single-button interaction is often caused again by hardware constraints and it is often used in combination with other elementary modes such as stare pointing. In a gaze and stare interaction, even a single binary button enhances the usability of the interaction as it eliminates the necessity of a stare timer.

3.3.4.5 Custom tracked controllers

In order to combat the downsides of traditional controllers, VR HMD manufacturers have come up with a new breed of input devices. This new breed of controllers is tracking enabled and can be tracked in a 3d space. This allows the VR software to create a 3d representation of the controller (or a suitable avatar for it) enabling the users to find the controller in the real physical space even when wearing an HMD.

The tracking furthermore allows novel ways of interaction. When the position of the controllers is 1:1 mapped to the virtual scene, users become able to "touch" and "grab" virtual objects simply by creating a positional collision. Apart from the usage of absolute position, most controllers are also equipped with IMU's and the data from the IMU's can be decoded to gestural input.

One problem with custom tracked controllers is market fragmentation. Contemporary commercially available VR HMD systems use different strategies for tracking the controllers as described before and show different performance metrics on the accuracy, trackable space, latency, and repeatability. They also have different button configurations. This creates the necessity for a VR software developer to carefully evaluate the offerings and design around limitations.

Another feature of the custom tracked controllers is the built-in vibration motor. With this built-in motor, limited but useful haptic feedback is possible.

Innovation still continues with custom tracked controllers and exciting new features are expected to arrive at the consumers such as finger position tracking.

3.3.4.6 Hand tracking

While hand tracking is currently unavailable with commercial VR HMD offerings, it is expected from the next generation of devices. Some developers announced plans to employ hand tracking as discussed. It is also possible to employ hand tracking with additional commercially available devices such as the Leap Motion controller.

Hand tracking frees the user from using any controllers and uses the positions of the hands, fingers, and gestures as input. It allows the natural mechanics such as grabbing and touching to be used as a control input.

However, hand tracking has many downsides. The first downside is caused by the lack of physical objects. As there are no physical objects that are actually being touched or grabbed, the natural mechanics of the hand turn into mimicry.

3.3.4.7 Haptic technologies

Haptic technologies try to emulate the human sense of touch by creating cutaneous, kinaesthetic, or haptic cues. While specialized hardware that can simulate various cues, such as temperature, texture, and extensive force feedback, commercial VR controllers lack the depth of haptic feedback due to the cost and complexity.

Common commercially available and widespread VR controllers rely on simple kinaesthetics and simple tactile feedback from vibration motors at the moment.

Simple kinaesthetics are supplied by 3d controller tracking. As the user is supposed to move the controllers in a 3d space, limb locations and positions give back kinaesthetic feedback. On the other hand, as there is no force feedback present, kinaesthetic feedback in the form of muscle strain, weight, and meeting boundaries are not present.

For tactile feedback, most hand controllers carry a vibration motor to create an entry-level of feedback. For example, simple vibrations can be employed to alert the user of a collision between the controller and a set 3d space, such as a game object.

Even simple haptic feedback, such as a vibration-confirmation model, greatly enhances the user confidence and ability in controller positioning, especially in VRMIs. (Berdahl et al, 2012, Ryge et al, 2017).

3.3.5 Sound

Sound has the second most importance in creating immersion in VR. Heiling estimates that while sight's role in immersion is %70, hearing’s role comes in second with %20, consecutively followed by smelling (%5), touch (%4), and taste (%1) (Heilig 1992). Although this rough estimate is true for the fact that sound has secondary importance and vision is the main driving force, the sound has the ability to break the immersion solely on its own.

The sound is another sensory barrier in VR and that’s where its importance comes from. To detach the user from the real-world and create the feeling in presence, the sound becomes the second line of defense and is therefore included in commercial widespread VR offering with emphasis. Most HMD's are marketed with headphone support or built-in headphones. Compared to playback from loudspeakers, headphones are able to create a barrier from the real-world sounds. It is also easier to implement spatialization thru headphones.

In a scenario where users are allowed to move their heads, interactive spatialization of the sounds greatly enhances immersion and presence. When the latency in the interaction is low, this can further help position solving as small head movements are also of human nature for determining the location of incoming sounds (Howard and Angus, 2017). Modern software development platforms also ship with powerful sound spatialization tools (Url-3).

When VRMIs are considered, the sound itself gains another importance as the output of a VRMI is audio. This creates further latency considerations as the output of the instrument must closely match user input in the time domain.

3.4 VR Specific Problems

As virtual reality communicates with low-level systems of human perception in artificial means, various problems specific to VR exist.

3.4.1 VR sickness

VR sickness, often remembered by similar names such as simulator sickness, HMD sickness or motion sickness is the name for all the different feeling of discomfort that comes with the use of VR. VR sickness can happen in the form of eye strain, dizziness, headache, sweating, nausea, and stomach awareness (Mazuryk, T. and Gervautz, M., 1996).

The second wave of the VR reduced the rate of these said symptoms simply by employing better hardware as the main source of these symptoms are caused by optics, displays, latency, and lack of synchronization between the vision and the vestibular system. Ebenholz reports that most of the said symptoms are not present with subjects with LD conditions (Labyrinth-defective, dysfunctional vestibular systems) whereas they are present with healthy individuals (Ebenholtz 92).

With recent technological developments, manufacturers were able to produce HMD systems with very low tracking and display latency, high resolution, adequate optics, and good ergonomics. Due to the reduced latency and increased accuracy of the system, mismatch between the vision and the vestibular system is greatly reduced, therefore resulting in an acceptable level of VR sickness.

3.4.2 Screen door effect

The screen door acquires its name from the user reviews. It is a situation where the user can see grids between individual pixels, just like viewing the world from behind a screen door. Even with a high pixel density, screen elements are susceptible to the screen door effect as they are placed just centimeters away from the user and they are magnified with the optics of an HMD. The screen door effect is most noticeable when the users focus on the display and move their heads around. As the screen grid is static, it gets overlaid on the projected dynamic image and become noticeable even more. The screen door effect is experienced in different levels by different subjects.

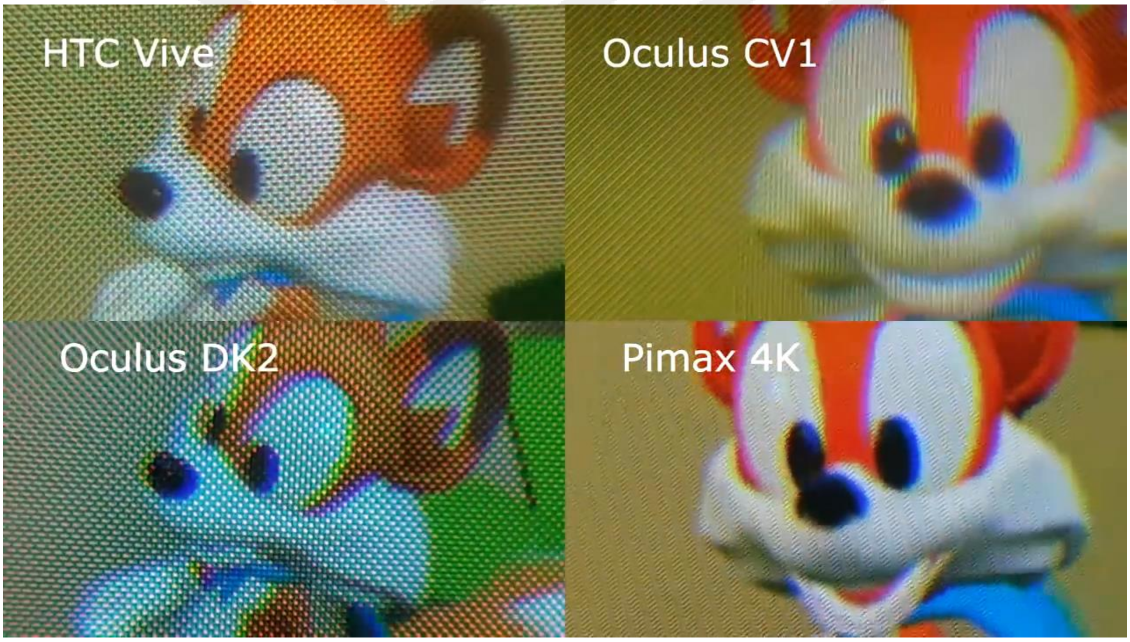

Figure 2.32 : Depiction of the screen door effect on various headsets (Image credit: VR Man Inc.).

In order to combat the screen door effect, HMDs must have high-resolution displays to make the grids even smaller and unnoticeable. As seen in Figure 2.32, the screen door effect highly depends on the resolution of the headsets.

3.4.3 IPD and focusing problems

Inter-pupilar distance (IPD) is another consideration of VR. IPD in humans ranges from 40mm to 80mm in extremes and with the vast majority of the adults, it ranges between 50mm to 75mm (Variation and extrema of human interpupillary distance).

The mismatch of the users' IPD with the VR system can happen in different ways and has various outcomes depending on the point of mismatch.

The first mismatch can happen in the headset itself. To combat weight and size requirements, HMDs employ smaller screens with matched optics to create an adequate FOV instead of giant screens. Optics enlarge the image from the smaller screens to a bigger one. Due to refraction, various distortions occur in the lenses and the least distortion prone are of the lenses are the center. Therefore, the headset must accommodate users with varying IPDs. When the IPD is not matched with the headset, users are met with blurred vision. Many commercial HMD offerings, therefore, support adjustable IPD. Mismatch on this stage is also a big contributor to eye strain.

Another point of mismatch is between the IPD of the stereoscopic image and the users. With computer-generated graphics, this can be solved by adjusting the position of the virtual cameras to the IPD measurement reported by the headset. However, when fixed media is viewed, such as a video shot by a 3d camera, the ipd is fixed and it can be corrected only by adjusting stereo images' positions. Mismatch on this stage results in horizontal image distortion and results in virtual objects being displayed with unrealistic sizes.

3.4.4 Effects of lag

Lag can happen in VR systems in different forms. All forms of lag are reported to be a serious contender in VR sickness.

The first form of lag happens in the tracking phase. If the headsets' and controllers' position and accuracy are not tracked with adequate latency, a great mismatch happens between the vision, the vestibular system, and kinaesthetic sensory systems resulting in VR sickness.

Another form of lag happens in the rendering phase. When the images that will be projected on the HMD are not rendered in time, dropouts occur and these dropouts are also a contributor to the VR sickness. The display is also a source of lag as there is significant latency between frames with low refresh rates. The persistence rating of a display is also important as images must disappear before the next frame comes. High persistence results in a syndrome known as ghosting and limits the effective frame rate.

Lastly, lag can also occur in the audio due to various reasons and greatly reduce both the immersion of a VR system and the performance of an instrument, especially in a VRMI scenario.

3.4.5 Ergonomics

Ergonomics is another important consideration in VR. The virtual world's ergonomy must match the physics of the user submersed in. Humans have different kinds of heights, and the virtual objects in interactive VR software must accommodate different users.

For example, imagine a virtual table with a static height. This table can be comfortable for average height users, but short users must lift themselves up on their toes, and tall users must bend over to something on the said table. Depending on the time spent with the said object, this can cause enormous discomfort. In scenarios like VRMIs, this poses increased importance as the users can have prolonged interaction with objects.

Some videogames tackle this ergonomy issue by adjusting the height of the virtual avatar but depending on the adjustment, the users can float above or dive under the virtual if they rest the HMD on the floor or come very near it.

Another solution is to make the virtual objects user adjustable. This can solve the issue without side effects but it is not applicable to all scenarios.

3.4.6 Fatigue

Fatigue is another special case of VR. Using tracked controllers and room-scale interaction methods creates more fatigue compared to seated keyboard and mouse control. Most of the fatigue comes from the requirement of lifting the arms up with weights in the form of the controllers but different modes of interaction can create additional sources of fatigue, such as bodily poses, walking, jumping, crouching, etc.

3.4.7 Visual-Haptic communication

As the objects in the virtual world are virtual, the haptic connection is broken. The collision between the users' hand controller and a virtual object is not a physical barrier and users can move their hands thru the virtual objects. As the collision is not physical, haptic feedback from touching, hitting, or colliding is also not present.

In order to remedy this, clear visual cues, audio cues or haptic feedback in the form of vibrations must be present and the VR software must be designed with the said limitations in mind.

3.4.8 Collision with real world

Another issue in VR is the possibility of a collision between the user and a real-world object. As the users' vision is detached from the real world, they are not able to see the real-world objects around them. In order to safely experience VR, users must create an empty, object free play area around them. Even when such an area is created, users can move out of the safe area as they can't see the real world and they must be kept in the center of the said safe area.

Various methods can be used to keep users in a safe designated area. One method is to draw a marker on the virtual world floor to indicate the center of the safe area. Users can look at the marker and return to the center when they drift too apart. As the immersion and the feeling of presence are very strong with VR, this method fails easily especially during heated gaming as the users are preoccupied with interacting with the VR world and not necessarily looking at the floor all the time.

Another method is to draw warnings or virtual walls on to the users' vision to indicate boundaries or drift from the center. As permanent walls drawn on the vision can be bothersome, systems like the SteamVRs Chaperone system (Figure 2.33) draw these virtual boundary indicators only when the user is near physical boundaries.

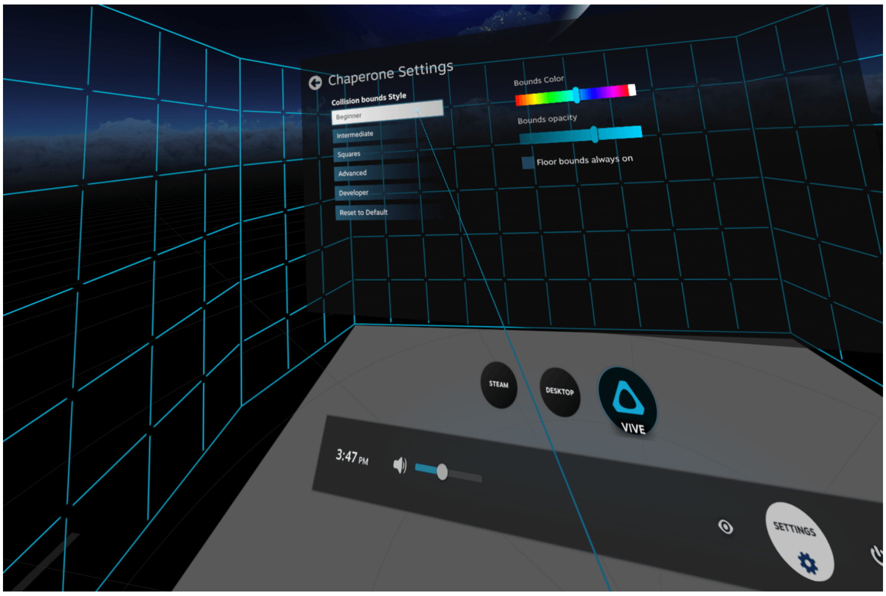

Figure 2.33 : Steam VR’s chaperone system (Image credit: Valve Corp.).

3.4.9 Locomotion

As discussed before, locomotion is a source of problems for VR and must be carefully implemented. There is no solution that is perfect for all scenarios.

VR sickness results when the user's view is moved automatically, therefore, making the room-scale walking the safest mode of locomotion. On the other hand, room-scale locomotion is limited by the bounds of the room, is a source of fatigue, and disables seated usage options, therefore, requiring additional locomotion modes such as directional keys or teleportation, each creates another set of problems on their own.

Due to the complex issues that result from locomotion, it is of high importance when designing VR Software.

3.4.10 Self-representation

As current commercial VR HMD systems only track the users’ heads and hand controllers, self-representation becomes a problem as the rest of the user’s body is unknown to the software. The unknown positions can be guessed from the position of the head and the hand controllers, but the mismatch results in the reduction of immersion. Because of the dangers, most software developers omit self-representation, meaning users can’t see their bodies when they tilt their heads to look at them.

Another problem with self-representation or avatar representation is with a second viewer. Imagine a multiplayer game where users can see each other's avatars. In order for a second user to see the other one, an avatar must be drawn. A rough estimation for the user's avatar can be drawn, but the system won't be able to differentiate between crouching or bending over for example.

This is also problematic for VRMIs where the performance is to be observed by a second party.

3.4.11 Going in and out of immersion

Going in and out of immersion too frequently can be discomforting and must also be carefully thought. When the fatigue is high, or the frustration is high, the users might need breaks from the experience. In a VRMI scenario, in which the user uses hand controllers and bodily interacts with a VRMI, and practices the said VRMI both the fatigue and the frustration is high. These can lead to the need for frequent breaks.

However, alternate, low fatigue, low frustration secondary objects, such as mini-games, can be employed to keep the user in the virtual world to avoid going in and out of immersion.

4. PROPOSED MODEL OF DESIGNING AND BUILDING VRMIs WITH A CHANCE OF SURVIVAL

As discussed earlier, this thesis assumes that survivability is the most important aspect of an instrument and aims to better the chances of an instrument's ability to survive.

When we look at the previous research on the evaluation of digital musical instruments, we can arrive at the fact that there is no perfect formula towards making a great instrument. Considerations in terms of efficiency, musical output, diversity, expression capabilities, the possibility of virtuosity, and mapping cannot be globalized and these considerations can be true only for specific cases.

If we look at the history and look at adopted musical instruments, we come across a wide variety of acoustic and digital instruments, which succeeded and failed in various performance metrics of previous thoughts of instrument evaluation.

If we think of musical instruments as tools, it is easier to see why a single instrument can't be the answer for all scenarios as specialized tools will be better for specialized scenarios.

One common trait amongst the instruments, whether they are acoustic, digital, or software, is the fact that they were produced in some form. Acoustic and digital hardware instrument designs that survived are the ones that can be realized, built, and sold. On the other hand, software musical instrument designs' ability to survive necessitates a similar but different set of properties. While the same set of principles like being realized (programmed), published, and sold, are also correct for software instruments, the survival of the instrument also depends on the survival of the associated hardware platform it is designed for, and the software's ability to be transformed (ported) on to successor platforms if the former platform ceases to exist.

Digital technologies evolve rapidly and it is common for a platform to cease existing. On occasions like these, the successor platform doesn't necessarily provide backward compatibility, and software instruments need to be re-written to work on the successor platform. Another concern of a software instrument is to be able to support multiple platforms from the start to reach a greater userbase as the userbase of the instrument will be a fraction of the targeted platforms' userbase.

Whether acoustic or digital hardware, the cost of producing an instrument involves design costs plus the cost of materials, assembly, packaging, shipping, and marketing. However, the greater percentage of the cost of producing a software instrument lies on the design stage as there is no physical production, packaging and the shipping can be done via electronic distribution.

While the sound production capabilities of the acoustic and digital hardware instruments are highly correlated with the physical cost of the instrument, any capability of a software music instrument cost in development time only as we have nearly infinite processing power in regards to sound generation on even the simplest of the possible target platform.

VRMIs are a subset of software instruments and VRMI design can greatly benefit from enhancements from a software topology that can ease the development process, lower the development cost, harvest the existing skillsets of developers, and have portability and reusability of the code.

4.1 Definition of A Good Instrument

The quest for defining "the good instrument" has been an ongoing battle amongst the DME researchers since it's the first step towards enhancing design (Tanaka, 2000, Jorda 2005, Barbosa et al 2015). It is easy to conclude from the previous research that it is impossible to generalize and define what a good instrument is as it is highly dependent on the context and there are an enormous set of auxiliary factors involved.

While it is hard to define a good instrument, it is pretty easy to point out the weaknesses in existing instruments and enhance them in a specific way. However, this doesn't necessarily make the enhanced instrument a better one. Take electric guitar as an example. One can say that an electric guitar with humbucker pickups is an enhanced version over an electric guitar with single pickups. While the one with the hum-canceling pickups is technically superior on paper in terms of output power and signal to noise ratio, it is not better than the one with the single pickups. The humbucker electric guitar opens the doors to a new repertoire but it is not exactly compatible with the established repertoire of the single-coil electric guitar. The humbucker guitar is now better for the newly created scenarios, not universally better.

Take the violin as another example. One can say that the modern violin with steel strings is an improvement over the baroque violin with gut strings as steel strings bring better control, the stability of intonation, and louder output. While certain aspects of the modern violin are better than the baroque violin, they are now two different instruments and it is impossible to say that one is better than the other.

Another perspective for the "good" comes from asking the question "good for who?". From the performer's perspective, performance metrics weights significantly more than anything, but the perspective of the performer isn't the only one. How about the manufacturer's perspective? From a manufacturer's perspective, the good metric can be the ease of manufacturing, projected sales, or total profit. How about the perspective of the environment? Now the "good" becomes the environmental cost. While ivory can be the best material for piano keys, is a piano with ivory keys a better piano? There are many perspectives to look at an instrument and as the perspectives grow, defining what is a good instrument becomes harder.

When we can't define what a good instrument is with confidence, the next best thing is to define is "good enough". Jorda gives traditional instruments as an example and states that they must be good enough that they resisted the acid test of time (Jorda, 2005). In order for an instrument to survive, it must be good at at least something, survival.

4.2 Benefits of Survival

The first and main benefit of survival is existing. If an instrument doesn't exist, all the other metrics of the instrument, like expression capabilities, sound palette, learnability, etc, also cease existing. Any qualities of the instrument don't matter if it doesn't exist.

Another important benefit of survival is the extension of the time the instrument is accessible. While an instrument that ceased existing can still have value, as interaction with it can create repertoire, ideas, and inspiration that doesn't necessarily cease existing with the cease of the existence of the instrument that led their creation, an instrument that isn't accessible at all doesn't even create values like these. The longer an instrument survives, the longer it stays accessible. Factors and necessities that make an instrument survivable also make it accessible.

A feedback loop between the performer and the luthier can only be established if an instrument survives. This feedback loop is highly beneficial as it leads to evolutionary improvements over the instrument's life.

Survival is not only beneficial to the instrument itself, benefits extend to every imaginable perspective. For example, practical benefits such as profit from commercializing the instrument can be observed from the perspective of the instrument maker/luthier/developer/manufacturer.

As existence is a prerequisite for any other quality of an instrument, and survival is a prerequisite for existence, it can be argued that the survivability of an instrument is the most important aspect of them.

4.3 Problem of Survival

As important as survival is, it is not always guaranteed and indeed, the chances of survival of a new instrument are pretty low. These low chances are caused by both internal and external factors. Internal factors are design considerations, that affect the survival of an instrument. External factors are the factors that are out of the control of the designer.

The major external factor that lowers the chances of survival is the competition with the already existing instruments. Instruments that already exist have the advantage of having performers, virtuosos, repertoire, and media exposure. Without performers, it is not possible to have virtuosos and repertoire. Without repertoire, it is hard to acquire performers, virtuosos, and natural media exposure. This conflict makes it even harder for new instruments to get picked up and survive.

Another external factor is the competition from the outstanding number of new instruments that are being created every day. The attention of the small percentage of people interested in new instruments is divided between many new instruments. Even if a new instrument can create a userbase, this can be short-lived.

While external factors are uncontrollable and will mostly always exist, internal factors, the design of the instrument, is also problematic as there is no perfect formula for a future instrument that will survive as the factors and considerations involved are inherently too complex to calculate the outcome. However, while there are no design considerations that can guarantee success in terms of survivability, there are big ones that can guarantee failure.

The main design factor that affects the survivability of an instrument is the intention to survive. In terms of physical instruments, this is the maker's intention for finishing the development, producing copies, and delivering to the public. In terms of software and virtual instruments, it is the intention of publishing the instrument. When there is no intent to finish development and publish (or deliver), survivability and adoption can't be expected.

This design factor can be observed mostly amongst hobbyist and academic instruments. According to Morale's research, an alarming number of new instruments presented in the leading international conference series, NIME, New Instruments in Musical Expression, failed in longevity.

The research points out that 47.1% of the instruments are not ready for a performance and 15.5% of these 47.1% don't exist anymore (figure 2.34). Another finding of Morale's longevity research shows that 68.7% of the instruments are not available to buy or hire, and out of the 31.3% that is commercially available, only one-fifth of them have sold at least 1 copy. These alarming numbers show that there is a survivability problem amongst academic new instrument research (Morreale, 2017).

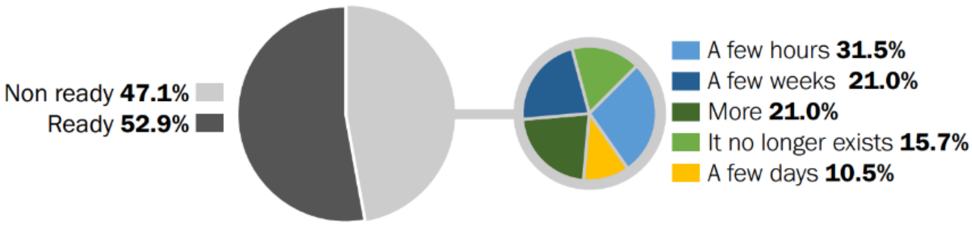

Figure 2.34 : Answers to the question " Is the instrument ready for a performance?" (Morreale, 2017).

The problem of survivability isn't solely caused by intention. Design considerations such as software topology also play an important role. The tools and libraries that the researchers use while creating the software don't allow for a packaged end product. As seen in examples of academic VRMIs in section 2.3.1, most of the instruments are comprised of several different software working in harmony. This method of building instruments creates both packaging and licensing problems, destroying the possibility of an end product. This decision making can be understood by figuring out the limitations of the software development platform for general software. While being great at the UI, graphical and interface control, popular software development environments such as Unity3D and Unreal Engine lack the tools for creating complex sound generation. To remedy this, academic instruments tend to run a second software, such as Max/Msp, PureData, and SuperCollider, amongst the main program for the sound engine. Handling the audio synthesis in these software are not only because of the capabilities of these software but because of the fact that most researchers are fluent in programming with these software as teaching audio programming in these environments are in the curriculum of many music technology programs across the globe for the past 2 decades.

On the other hand, when we look at the commercial VRMI offerings listed in section 2.3.2, the survivability and adoption rate is poor, not due to programming considerations of survivability but the content of the software. Firstly, many of these software rely on sample playback as complex synthesis is not possible by default in Unity3d or Unreal Engine. This limits both the type of instruments that can be created and the expressive options. Secondly, as can be understood from the user comments, these instruments lack the features, such as gamification, that a regular video game employs towards user engagement. The result becomes, toy-like VRMIs that lack the ludic elements of modern gaming.

4.4 VRMI Design Strategies Towards Survival

As discussed earlier, survival is a very important aspect of a new instrument and while we can't guarantee success, we can adopt practices that can eliminate the pitfalls that guarantee failure of survival and can potentially increase the chances of survival.

4.4.1 Goal of publishability

The quest for survival starts with the first pre-requisite. The first goal of VRMI design should be a functioning instrument. In order to be publishable, the software must be free of licensing problems that can result from its various sub-parts. Also, the software should be one piece, and shouldn't necessitate helper software, such as an external audio engine, to function. Compatibility with common software distribution platforms such as Steam(steam link) is not a necessity but being listed in such a platform can dramatically increase public attention and reach. Installation of the software should be trouble-free and shouldn't require extra effort from the end-user.

The job of the developer doesn't end with the creation of the VRMI software. The developer should also create marketing and infomercial material for the software. This can range from a simple website to a detailed product information page on a software sales platform such as Steam.

4.4.2 Increasing the ease of development and lowering development costs

Due to the elements involved, VRMI lutherie is an utmost multi-disciplinary field. A VRMI is comprised of various assets such as 2D/3D graphic assets, virtual architecture, user interface, logic programming, audio engine, control mapping, and instrument design.

As there is no guarantee of the survival of a new instrument, increasing the ease of development and lowering the development cost can have a positive effect as it will create more incentive to start a VRMI project as well as developing another one if the former fails in terms of survivability. Ease of development and low costs also allows the developer to implement more features in a set time, increasing the potential of the VRMI in question.

The first step towards the easement of the development and lowering the costs is using a suitable development environment. Modern development platforms such as Unity3d and Unreal engine allow rapid prototyping, modeling, and programming. These platforms also have free editions as well as paid versions that favors small volume developers. Built-in graphics engines of these platforms are capable of competing visuals both in quality and performance and eliminate the need to develop a graphic engine. These platforms also ease the interaction with various computational modules, such as the operating system, file systems, input devices, VR-specific hardware, and system sound. Having pre-made features for such interaction greatly reduces development time and effort, therefore, reducing cost. Both of the platforms allow team collaboration and a project can have multiple developers working on it. Another very useful feature is the asset stores that come with these platforms. Users of the platforms have access to various building assets, such as 3d models, lighting systems, UI plugins, sounds, materials, and so on. This greatly reduces the multi-disciplinary burden of VRMI creation, as even a single developer can create high-quality VRMI with little knowledge on various other disciplines (such as 3d design) by outsourcing several components with ease.

However, although both of the said platforms have extensive audio capabilities, such as positional audio, DSP processing, granular synthesis, external plug-in support, and more, none features modules that can be used in a software instrument as synthesis is problematic and limited in these platforms. It is still possible to do complex synthesis via custom modules or external plug-in support but it is cumbersome and doesn't efficiently extract the skillset of a common music technology graduate.

As previously discussed, audio programming platforms such as Max/MSP, PureData, and SuperCollider are frequently taught in music technology curriculums and many DME luthiers are already fluent with these languages. Amongst the listed three, PureData is both GNU license and portable. It is possible to use PureData as the sound engine, and port it inside an Unreal/Unity3D project, still maintaining the possibility of a single package end product while harvesting the already existing audio programming skillset of the designer. The integration of PD into Unity3D will be discussed in detail in later sections.

Harvesting the existing audio programming can greatly reduce the burden on the audio programming side while increasing what features can be programmed in simultaneously.

4.4.3 Platform compatibility

Platform compatibility is another important concern in VRMI design in terms of survivability as it directly affects the number of potential users. The greater the platform compatibility is, the greater the number of potential users are.

As mentioned earlier, there are various commercial VR offerings with varying already established user bases. These commercial offerings are divided between devices that connect to personal computers, devices that rely on the use of smartphones, standalone devices, and VR systems for game consoles. Although it is possible to create a VRMI that can target all breeds of the current VR systems, this approach will limit the functions of the software with the limits of the lowest-performing system. As VR systems that work with smartphones don't have the precision tracking and the computing power the other systems have, this becomes a severe limitation as of current technology.

Compromises in platform compatibility, therefore, results in more resources but diminishes the targeted userbase. In theory, with enough funding, a new VR system with custom controllers that are specific to the VRMI in question can be manufactured but this will be a huge financial risk for the developer as it reduces the already established userbase to zero. Also, the costs of manufacturing and development will need to be addressed to the end-user and will create a big cost barrier. Designing a custom controller for existing systems is another idea that will still introduce a significant cost to the end-user, but as it is just a controller, the costs will be much smaller.

On the other hand, the desktop breed of current commercial VR Systems already has near-infinite processing power, controllers with precise and fast tracking, and an adequately established userbase. Targeting these systems only will not need any additional hardware to be designed, built, sold, shipped, and added to the cost of the VRMI.

These tradeoffs make the platform choice a very delicate matter, and it is an essential part of the VRMI design.

4.4.4 Accepting a variety of users

Existing DME research shows us that there is a tradeoff between the complexity and the learnability of an instrument. An instrument with great learnability attracts novice performers but seasoned performers are repelled as there is no room to grow. An instrument with a difficult learning curve can be rewarding for serious performers but will repel novice users as it will be too time-consuming to establish an adequate level for performance.

This tradeoff can be observed in various traditional instruments, DMEs, and novel interfaces. The complexity of an instrument had always been a big concern of the instrument designer in terms of survivability.

The video gaming industry also has this tradeoff and they were able to tackle this problem with various methods. One common method is to introduce various levels of difficulty, by either artificially reducing or increasing the difficulty by creating bonuses and handicaps. For example in a first-person shooter game, a lower difficulty setting lowers the enemy healths, and a greater one increases them. Another method of accepting users with various skill levels is by keeping the difficulty constant but helping the user with assists. An example of this would be a driving simulator where the users are shown the best line on the track and input complexity is reduced with driving aids such as anti-lock brakes and traction control. With enough mastery all these aids can be gradually reduced and turned off, allowing the game to serve a large spectrum of players, from novice to expert.

In multiplayer contexts, the gaming industry tackles this issue by grouping players with similar performance levels together. One example can be Blizzard's game Overwatch. Players are asked to several online placement matches and are given a rank depending on their performances. After the placement matches are completed, players are matched with others closer to their ranks on their ranked gameplay. Although not always perfect, this levels the playing field for many players.

While these methods are hard to implement with fixed instruments such as traditional instruments and fixed hardware controllers, the beauty of software instruments such VRMIs is the fact that they are not inherently fixed and can be designed flexibly. For example, imagine a VRMI implementation of a theremin. The theremin is an instrument with a simple but hard to control input scheme and extremely unforgiving to novice users. It has two antennas and the interaction is done by varying the distance of hands between the antennas. One antenna controls the pitch, the other controls the amplitude of the signal generated. As the instrument doesn't have fixed pitch intervals, precisely arriving and holding a certain pitch requires great mastery. In a VRMI of a theremin, a difficulty slider that adds pitch stops can be implemented. On easier settings, pitch stops, like virtual frets, can be introduced to reduce the input difficulty greatly.

Another thing that we can learn from the gaming industry is the fact that inherent difficulty brings psychological rewards and ludic motivations which attracts a special audience. For example, the game series Dark Souls is notorious for its difficulty, punishing novice players nearly every minute through the gameplay, and still, the series was a success in terms of sales and criticism. Another example is the game Surgeon Simulator where the players are asked to control the hands and digits of a surgeon with a very inefficient control scheme, forcing players to mastery. Failure in this game is not frowned upon and players can be observed enjoying failure scenarios.

There might be a market for an inherently difficult music instrument that punishes its users. The lack of efficiency can be remedied by the ludic elements and psychological rewards of accomplishing things.

4.4.5 Altering the learning curve using performing aids and gamification

As mentioned earlier, a VRMI in its software nature is not fixed but flexible, and therefore optional features for different performer skills can be programmed in.

These optional features can be made to alter the learning curve and turn the instrument into a novice-friendly one. As the features are optimal, they won't be limiting or burdens to expert performers as they can be simply turned off.

Performing aids that simplify the control scheme can be implemented to allow novice users to produce meaningful output with little mastery. Visual, audible, and haptics aids can be employed to help users navigate controls easier.

A teaching method can be programmed right into the instrument acting as an interactive tutor. The learning progress can be turned into a more enjoyable one with the help of psychological rewards such as player levels, unlockable in software objects, or a storyline.

Compared to a well-established instrument, a VRMI will start with zero expert performers. Every new performer will be a novice at the start. While some performers might carry their experience over other instruments and will be generally more skillful, all will start from the start. Helping performers get past the novice level can dramatically alter the survivability of the instrument.

Skill level is also dependent on the intention of the user. While some users can aim to practice and master the instruments, there will also be users who don't look forward to going into mastery as seriously. Implementing performing aids and gamification can also make the VRMI functioning for these kinds of users.

4.4.6 Improving user engagement with inspirations from video gaming

Throughout the history of video game history, game designers developed an extensive library of design elements that increase user engagement in gaming software. This library is not only constructed with research but also with trials and failures, end-user – developer feedback, and sometimes with luck. With more than 50 years of history, video gaming has a lot to get inspiration from. Throughout its history, the video gamings path also crossed with music-making many times in many ways.

Currently, creating a practical VRMI necessitates borrowing hardware and development tools from the video gaming industry. A finished VRMI can be sold amongst video games and even the targeted userbase will be largely comprised of video gaming industries userbase as current commercial VR headsets are designed and marketed to mainly gamers. Since the paths of VRMI design and video game design is this close, it is inevitable to look at video gaming for inspiration.

User engagement is a concern for the video gaming industry as the competition is fierce with many new titles being release everyday. This competition introduces the same survivability concerns as new musical instruments with slight differences.

For an example inspiration source, exploration in a video game is a concept what we can find analogs in music instruments. Exploration is one of the stronger driving forces of video games and it’s analogous to exploring a musical instrument. While there is some analogy in exploration, the exploration timeline can be completely different in video games. Depending on the genre and the video game itself, introducing explorable elements can be costly and limited. Some video games, therefore, introduce methods, often referred to as grinding, where the player will have to acquire in-game points to unlock the next piece of explorable material. The act of unlocking the next bit becomes a psychological reward, and slowly but progressively reaching the goals becomes a ludic element. This way, even with a limited set of explorable content, video games are able to extend the user engagement time.

This method of user engagement enhancement can be implemented for the practice of a VRMI, rewarding performers through their training, making the dull process more enjoyable. This method can also turn the VRMI into a video game in which even users with no intention of music performance can be reached.

4.4.7 Marketing

Like any modern software, a VRMI will need marketing in order to survive. In today's attention economy, it is hard to acquire an audience without spreading the word.

Marketing is a field on its own and out of the scope of this thesis but nevertheless, it's worth mentioning that marketing should be thought along with the development of the VRMI.

4.4.8 User feedback

User feedback had always been an important aspect of instrument design as it leads to evolutionary enhancements in an instrument or instrument family's lifetime. As previously mentioned, it leads to a healthy performer – luthier feedback and creates the possibility to come up with improvements that can't be thought of during the design stage.

With hardware and traditional acoustic instruments, the development stops at the time of shipping the product and any user feedback gathered can be put to use on the next iteration of the instrument/instrument family. On the other hand, software instruments, and therefore a VRMI, can be updated anytime in their lifetime, as frequently as the developer is ready.

This ability to update on demand is game-changing as it has the power to both enhance a good instrument and to save a doomed one. Even if an instrument fails to satisfy a customer base at the initial launch, user feedback can be put to use to fix what is wrong.

When all of the above is considered, gathering user feedback is a must and a platform for gathering user feedback should be a consideration of the developer. While there are various ways to collect user feedback which range from a simple advertised e-mail account dedicated to support and feedback to online forums, some publishing platforms (Steam for example) already create a sufficient platform to receive user feedback. As seen in section 2.3.2, user reviews on the Steam product page can give us valuable feedback from the perspectives of many people.

4.4.9 Data-collection and analytics

Albeit being controversial, anonymous data collection and telemetry are powerful tools that can give insight into how the users are spending their time in the software in question.

A data collection mechanism can keep track of anything happening on the software and report it back to the developer. Reports like the host systems capabilities, such as CPU, GPU, disk space, VR headset information, average fps can give the developer insights about the running performance of the VRMI versus the user bases computing capabilities. With this information, VRMI can be tailored to run better amongst the common computing capabilities.

Users' actions in the VRMI can also be logged. Reports on information like the user time spent in software, user progress on the content, user progress on levels (if any present), most used features and the causes that make the users quit the software can give valuable insights to the developer which was only possible with close testing and monitoring. For example, if reports can be provided on the most used content, the developer can choose to improve the popular content or fix what is keeping the less used content down.

While anonymous data collection and analytics is beneficial both to the developer and the end-user, there is still controversy around it as it can be seen as an invasion of privacy. If such a system is chosen to implement, the developer should be clear about what data is being collected for what reasons and the users should be given the chance to opt-in or out.

4.4.10 Updates

As mentioned earlier, updates are a great way to continue development after the initial release. Unforeseen aspects of the software can be fixed, user feedback can be put to good use and more content can be added over time.

Another benefit of updates is the perceived image of the software. Users are known to favor software that is being updated regularly. Just like users, software markets such as Steam, Apple’s App Store, or Google’s PlayStore favor regularly updated software in listings.

4.4.11 Fun

Fun and ludic measures are an often overlooked factor in music instrument design. While fun and ludic measures are not directly related to the performance, it is an important factor in an instruments learnability and survivability. This can be especially important for a VRMI as there is a lot of common grounds with video gaming, where the expectation of fun is high.

When we look at the user reviews of sample commercial VRMIs listed in section 2.3.2, the lack of ludic elements and overall user experience being poor are big concerns amongst the reviewers and this is often caused by the fact that the instrument and its capabilities are the only explorable element in the software.

Lack of fun especially turns off novice performers and people who buy the software expecting a video game. Since a brand new instrument doesn’t have any master performers, capturing the attention of the novice performers is very important for the survivability of a VRMI.

The fun factor of an instrument is often linked to its learnability and therefore its input-output complexity and finding a right balance between boredom, frustration and challenge become an issue. A simple instrument with easy learnability can be picked up very easily, but it will lack in challenge and explorability for performers past the novice stage. On the contrary, an instrument that is hard to master can be too big of a frustration for novice performers(p59 Jorda).

As discussed earlier, a VRMI doesn’t have to be a fixed instrument, and the learning curve can be altered with varying difficulty and performing aids, eliminating to find a right balance of boredom, frustration, and challenge (if there is any right balance there, to begin with).

However, the alteration of the learning curve is not the only solution to the ludic elements. The instrument itself doesn’t have to be the only fun factor in a VRMI.

Again, the video gaming industry is a great inspiration for a VRMI and ludic elements of video gaming can be adopted to VRMI lutherie. Gamification of a VRMI can greatly improve its appeal to novice performers and non-serious performers. Introducing levels, psychological rewards, storylines, task, and auxiliary activities can be sources of ludic elements apart from performing.

Another source of ludic elements can be sourced from the task of performing. Successful video games task the user with a certain number of actions per minute and keep the user almost always busy. Sometimes intentional difficulty is introduced to these tasks, even with aerobic difficulty to create a challenge-reward mechanism for mundane tasks. For example, while a door can be programmed to be opened by a simple button press, some video games require a prolonged button mashing to do the simple action. As the user has to put in the effort, accomplishing this mundane task has more perceived reward. This method is an example of introducing frustration when the challenge gets too low. With inspirations from video gaming, RubKeys (figure 2.35) uses a similar mechanism to introduce intentional aerobic difficulty into a 2-octave touch screen keyboard. Excitation of the keys becomes harder in the last octave progressively. Reported feedback is positive in terms of ludic elements (Sarıer 2014).

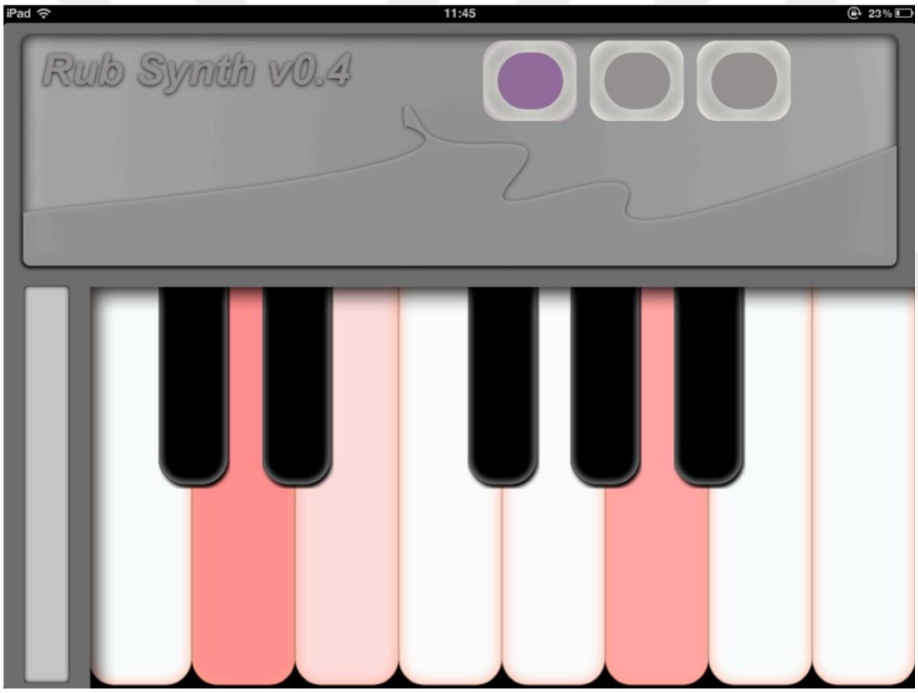

Figure 2.35 : Sarıer's RubSynth (Sarıer, 2014).

One might think that with enough ludic mechanism borrowed from the video game industry, A VRMI can stop being an instrument and become a video game itself and there is apparent danger in implementing these mechanisms. However, turning into a video game doesn't necessarily stop a software from being an instrument. In fact, several video games, with no intention of being a musical instrument, have been used as musical instruments by artists. For example, the popular creative sandbox/survival game Minecraft introduced a single musical element in-game – the noteblocks.

Noteblocks are an in-game placable object that produces a single upon user interaction. In nearly no time, some of the users started to build sequencers in the game and creating artworks with it. Because of the fact that Minecraft has millions of active players, even a small percentage was enough to start a wave and begin using Minecraft as a musical instrument.

Another behavior that can be observed from video gaming is the player-created difficulty levels. DME research suggests that an instrument that is easy and limited in its output potential is limiting and boring to master performers. In video gaming, there are many instances where some players become "virtuosos" of a game, beating the hardest difficulty easily and then proceed to implement their own rules to increase the difficulty, such as completing a game without taking damage, without saving, without using a certain function, etc.

With enough users, the functions of a VRMI can grow bigger than the developers' imagination and intent. An easy, limited output but popular VRMI with a million users might have a greater variety of musical output than a complex, infinitely open output but unpopular VRMI with only a few users. From this thinking exercise, we can see that users are what feed the potential of an instrument and survivability again shows how important it is for an instrument.

4.4.12 Consideration of VR specific issues

VR specific issues must be handled well in a VRMI. Motion sickness is still present among VR users and must be eliminated to a minimum. This necessitates having an adequate resolution, ipd conformity, and frame rate.

The resolution is dependent on the choice of VR system platform and the frame rate is dependent on the choice of targeted hardware and the graphical workload of the software. Because of the fact that the users whole field of view is covered with the projected image, and each eye needs a separate image, the final rendering resolution requirement is much higher compared to a 3d software that is meant to be projected on a 2d screen (such as a monitor). Rendering at such resolution is currently taxing for the available GPU power in common users, and if a good rendering engine is not used, the drawn 3d content will be severely limited. Fortunately, modern development platforms such as Unity3d and Unreal Engine has 3d engines that are well optimized. Optimizations dramatically reduce the workload on the GPU and allow more polygons to be rendered in real-time. For example, occlusion culling is a method in 3d rendering where objects that can't be seen by the user due to being blocked by other objects are not rendered. As many objects are usually behind others, this method greatly reduces the polygon count to be rendered each frame. Another example is baked lighting maps. Instead of real-time lighting calculations, the effect of static lighting is pre-calculated and baked on to the textures of an object in the scene.

Using a modern IDE and its graphic engine greatly reduces the programming work for the developer and increases the frame rate. Thanks to these IDEs, even a single developer is now able to create a VRMI, 3D game, etc.

Locomotion is another important factor in creating VR Sickness and therefore must be implemented in a way that doesn't cause sickness. When there is a mismatch between the players' actions and the movement of the virtual avatars, VR Sickness is introduced. Currently, the best practice for locomotion is using room-scale tracking in combination with teleportation as discussed in section 3.3.3.

Ergonomy is another VR specific issue, especially with VRMI. Music instruments often require prolonged hours of use and users should be comfortable with their body position and gestures. People with varying heights should be taken into considerations as virtual objects placed in the virtual world can be too low or high depending on the height of the user. Users should be able to adjust their virtual avatars' height or the placements of virtual control objects if there are any. Compared to non-VR computer software where the user is sitting in front of a monitor, VR is already taxing on the user due to the greater bodily gestures of moving in a virtual world. Bad ergonomic decisions can rapidly tire the user, resulting in shorter user engagement.

Another VR specific issue is the fact that the performing area now has to be crafted by the developer as well as the instrument. Performing location has a strong impact on the performing experience. As can be seen in the reviews of the commercially available VRMIs in section 2.3.2, some complaints arose from the virtual space alone. Creating a pleasing virtual environment is a complex issue and a multi-disciplinary field on its own. It involves 3d modeling, texturing, architecture, scripting, animation, and many more subtle details. One approach to cut down on the time and resources used in creating a virtual environment is to use assets and premade objects. Assets and premades are objects that are a composite of various smaller objects such as lighting, 3d models, textures, cameras, sounds, etc. Environmental objects, such as decoration objects can be acquired as assets. Even whole environments, complete with their 3d models, textures, various decoration objects, and lighting are available as assets.

Assets can greatly increase the speed of production and decrease the number of specialists needed (or knowledge in the special field, such as 3d modeling).

Another approach to a user satisfactory virtual environment is to outsource it to users themselves. Many videogames have support for user-generated content where more content is created by the community of the software. For example, the racing simulator Assetto Corsa has vast amounts of tracks, cars, helper software, and car skins that are made by the community. There are even community-run websites for sharing the content. The online game sales platform Steam has a Workshop functionality to streamline this process of user-generated content. It provides a hosting space for such content and provides easy integration with the software.

Virtual Reality is an immersive and shocking experience. Going into a virtual world is an out of this world experience and likewise, going back to the real world has a negative effect. Music practice often requires breaks and can result in frequent going in and out of immersion in a VRMI. If secondary tasks, such as mini-games, are programmed into a VRMI, users can be kept in the virtual space longer and can result in a more pleasant overall experience. Therefore, the break times and going in and out of immersion can also be an important concern for the VRMI luthier.

Lag is an important issue for both music instruments and virtual reality software. For VR software, significant lag breaks the immersion/presence and introduces VR sickness. For musical instruments, significant lag makes the inputs uncontrollable. It is vital to keep lag to a minimum.

Lag can happen in many stages of a VRMI. Visual lag can happen with heavy graphic workloads where frames can’t be drawn in realtime. This also creates an input lag where the controller information is delayed. Ensuring stable and high framerates is therefore important. The developer should make sure that the software can run without hiccups on the lowest advertised system. Sound lag can happen when both input lag happens, or there is latency in the audio engine. Both input lag and audio engine latency should be kept as low as possible.

4.5 Proposed Model for VRMIs that Can Survive

As discussed earlier, the underlying cause of the state of current VRMIs can be traced back to the software topologies used. VRMIs that are made with game development environments such as Unity3d and Unreal Engine lack depth and variety due to the limitations of the built-in audio engine capabilities. On the other side, VRMIs that are rich in features are often a combination of software that is running in parallel, and can't be packaged into publishable products. In order to combat this dilemma, this thesis proposes a new hybrid software topology that can harvest both the strength of a popular modern game development environment and established fluency and audio programming capabilities in a well known open-source audio programming library.

The proposed model consists of using Unity3d for the main structure of the VRMI and using libPD to create a PureData instance in a Unity project to use as the audio synthesis and signal processing engine. However, running PureData in this configuration is not straightforward and has issues on its own. A helper library named VR Creation Toolkit, which was developed along with this thesis is also a part of this model to combat the integration issues as well as serving template functions to greatly speed up the development process. VR Creation Toolkit will be presented in Chapter 5 in detail.

With this proposed framework, VRMI developers can handle the 3d rendering, VR headset integration, control mapping, physics, UI, and packaging in Unity3D, while they can use a familiar environment, Pure Data to create and shape the sound engine in any way they want. VR Creation Toolkit ensures easy integration and speeds up the development process by supplying the developer with shortcuts to bindings various virtual UI elements to the PD audio engine. VR Creation Toolkit also comes with sample projects and tutorials to make figuring out the framework a lot easier. The developed project can be packaged into a single application through Unity3d and a shippable end product can be created.

With this software topology and framework, it is possible to create feature-rich VRMIs with endless synthesis options that can be packaged into a single, publishable software. This topology eliminates the survivability's first obstacle, creating a publishable software. It also tackles the problems of limited audio synthesis when working in a game development environment.

4.5.1 Development platform

Unity 3D is chosen as the development platform for its capabilities, user-friendliness, and the author's familiarity with it. Unity 3d has a free version and there are a vast amount of free educational resources on how to use it. Unity 3D's built-in asset store allows outsourcing several aspects of a VRMI.

Unreal Engine is also very close to Unity3D in terms of capabilities, educational material, licensing, and so on. The final deciding factor for the platform was hugely affected by the author's experience with Unity 3D.

VR Creation Toolkit can be ported to be used in Unreal Engine or another capable game development environment in the future.

4.5.2 Development target

The current development target is kept to Steam VR and the HTC Vive headset. At the start of writing this thesis, HTC Vive was the only commercially available VR system that allowed accurate, fast and room-scale tracking. Although the VR Creation Toolkit and the proposed framework currently targets the HTC Vive headset, it is possible to target other desktop systems, as well as mobile platforms. While a huge portion of the VR Creation Toolkit is UI elements that are meant to be controlled with the controllers of a desktop VR system, with a slight alteration, they can be used also in a smartphone-based VR system. The fundamental of the VR Creation Toolkit that makes libPD run behind a Unity made application will still function in a different target.

4.5.3 Audio engine

Pure Data is chosen as the audio engine due to its capabilities and portable nature. In terms of portability, there is no competitor. Through the use of libPD, it was possible to use PureData programming language in Unity 3D after ironing some issues in the latest release at the time of writing.

A bug in libPD stopped it from working in applications built-in Unity versions greater than 5. Since new Unity3d versions came with a lot of useful new features, the problem that kept it from compiling to 64 bits was fixed, and a freshly compiled DLL is a part of the VR Creation Toolkit.

Another problem with using PureData was the fact that it replaced Unity 3Ds built-in sound engine altogether. While the Unity 3Ds engine lacks the advanced programming, synthesis, and ease of use of Pure Data, it also has useful features such as positional audio. When PureData is the sole audio engine, the 3d positioning of sounds must be done in Pure Data itself and it is a cumbersome task. With a simple but clever trick, PureData patch instances can be placed onto Unity 3D sound objects and Unity3Ds positional audio system can be used alongside PD. This trick of allowing PD instances is also a part of the VR Creation Toolkit.

4.5.4 Starting point and helper libraries

Without a starting point and helper libraries, implementing the proposed software topology comes with a lot of hassles that needs to be solved by every VRMI luthier that chooses the topology. As discussed earlier, ease of development is beneficial to the survival of a VRMI as the developers will be able to add more features or they will be more likely to try again if a project fails to survive.

VRMI Creation Toolkit is a helper library with starting templates. It was created alongside this thesis to assist future VRMI developers and help them save significant time in their future projects.VRMI Creation toolkit is described in detail in the following chapter.

5. VRMI CREATION TOOLKIT

VRMI Creation Toolkit is a software library developed alongside this thesis. This library is composed of elements that greatly speed up VRMI creation with Unity as the development environment and Pure Data as the sound engine.

As discussed in great depth in the previous chapter, software topology is the root cause of the current state of commercially available VRMIs. Unity, Unreal Engine, and the likes are great software development platforms for creating VR content. When coupled with already established VR related libraries such as Steam VR, Open XR, and the likes, these platforms allow the creation of virtual reality applications by supplying many of the needed elements such as hardware interfacing, graphical engine, world-building, sound spatialization, scripting, UI building, networking and so on.

However, while these platforms supply nearly all the necessary elements, one key element for VRMI creation is missing; sound synthesis. While these platforms allow low-level access to audio and one can go through the hurdles of low-level audio programming to accomplish synthesis (using Unity's AudioPlugin API for example), when compared to the ease these platforms create in all other areas, it is really cumbersome and creates a big obstacle for VRMI luthiers as low-level audio programming is not a part of the common skillset of many people.

When we look at the commercial VRMI offerings analyzed in chapter 2.3.2, many of the commercial VRMI offerings resort to the built-in capabilities of these development platforms, which is mostly limited to sample playback. Being limited to sample playback greatly handicaps the VRMI luthier and narrows down the types of VRMIs that can be built.

When we look at the academic examples of VRMIs, we can see that real-time synthesis is accomplished by using common audio programming environments such as Max/MSP, Pure Data, and SuperCollider, and the resulting instruments (or interactive experiences) are way boarder with their interaction methods and sound outputs however they suffer the same demise of many DMIs, due to the fact that most of the time they are composed of a bunch of software working together, far away from being able to be packaged together and published both due to technical and licensing difficulties).

As discussed earlier, this thesis assumes that survivability of an instrument is the most important virtue and it is dependent on both controllable aspects and aspects that are virtually uncalculable and dependent on chance.

One big controllable aspect that can hinder a software instrument's survivability its publishability. If the software instrument can't be published in a meaningful way, one cannot expect it to be picked up by people.

Also, from the discussion in the previous chapters, we can deduce that the sound palette of an instrument is still an important factor and the ability of programming real-time synthesis can greatly enhance the possibilities in terms of sound palette.

Lastly, we can also deduce from the previous discussions that chance is hard to eliminate factor in instrument design as there are many uncalculable aspects to the success of an instrument in terms of survival. An instrument designer can only hope for success even after carefully planning the controllable aspects of design as it is impossible to model and predict the public reaction. Change is also prevalent in nature and evolution tackles this issue with brute force. For example, human sperm has a very low chance of reaching and fertilizing an egg. While the human body might have produced a mega-sperm that had better chances of reaching and fertilizing an egg, this would mean the cost of the sperm to the host would be much greater. Instead, evolution tackled this issue with a strategy, identical to a brute force attack, where the cost is kept low and sperm count is increased. Increasing the tries increases the chances of success.

While we can't eliminate the chance factor in instrument survival, we can increase our chances by more tries. If the cost of development can be kept low, a luthier can have more tries towards a successful instrument.

The VRMI Creation Toolkit is aimed towards enhancing VRMI lutherie towards these three key points. VRMI Toolkit allows the use of a popular cross-platform software development environment, Unity, together with a common audio programming environment Pure Data. The toolkit allows the usage of both platforms together without hindering any of these platforms' abilities. The resulting software can be packaged and distributed easily.

Apart from the bridging of these two platforms, VRMI Toolkit pre-solves many of the technical issues present and allows the luthier to skip the hurdles and focus on the instrument design. The toolkit also has many pre-built objects that can help a luthier to easily create interaction to audio engine communication, virtual instrument elements, bindings, mappings, and messaging without the necessity to code.

Using Pure Data as the sound engine can harvest the already established skillsets of many people in the field and can greatly reduce the cost of software development. Likewise, the Unity software development platform also offers many features to ease and lower the cost of development. The VRMI Toolkit also allows Unity to retain its audio capabilities such as spatialization, reverb zones, and so forth, which are way too easier to implement natively rather than handling these tasks in Pure Data.

5.1 Dependencies of the VRMI Creation Toolkit

VRMI Creation Toolkit is a helper library built for the Unity software development platform. In order to achieve its functionality, VRMI Creation Toolkit brings together several software libraries together.

For handling the communications with the VR headset and controllers, it uses the Steam VR 2.6.1 library designed by the Valve Corporation. All of the pre-made interactions that come with the library support the Steam VR 2.0+ platform, and compliant with the codebase.

In order to run Pure Data in a Unity project, the toolkit uses the LibPDIntegration library developed by Niall Moody, which is also dependent on the libpd library maintained by Peter Brinkmann and the libPD team. Moody’s LibPDIntegration allows running more than one instance of libPD, therefore more than one Pure Data patches at a time.

5.2 Topology and the Elements of the VRMI Creation Toolkit

The VRMI Creation Toolkit consists of several Unity prefabs that can be dragged and dropped in a Unity scene and sample scenes and projects to demonstrate the usage of the elements.

The player pre-fab is responsible for the SteamVR interaction system. It consists of objects that handle the virtual camera for the HMD, objects that handle the controllers, objects for the audio listeners, and other objects related to the Steam VR interaction system.

The Teleportation pre-fab is responsible for the teleportation locomotion system and allows the designation of teleportable areas as well as the visuals for the teleportation system.

The PDInstance pre-fab is a pre-fab that can run a PD patch. It is pre-built in a way to only require the binding of the target PD patch from the developer which can be accomplished via the editor window with a drag and drop action. Multiple PD instances can be present on a scene and can be initialized with a simple drag and drop.

The audio output of the PdInstance prefab is a point source AudioSource element that uses the built-in Unity spatialization. When attached to a GameObject, spatialization can be natively carried out. The PDInstance prefabs can also be affected by the native reverb zones and can be routed to the internal audio mixer if wanted and processed by built-in effects.

The PDMessageServer prefab is an object for handling the communication between PDInstances and GameObjects. It maintains the drag and drop functionality by employing editor areas where source PDInstances can be mapped, and target areas to designate GameObjects to communicate to.

The PDInteraction_ class of pre-fabs are VR interactables that are designed for a variety of modes of interaction. While these interactables come with premade 3d models, the developer can quickly change the associated 3d models with a simple drag and drop interaction. Also, the GameObject to PDInstance communication can be simply mapped via the built-in editor options. All of these interactions conform to the Steam VR 2.0+ interaction system which allows controller mapping with a built-in graphical user interface.

PhysicalObjects_ class of prefabs are GameObjects that are adopted from the SteamVR library which conform to various physics rules and are designed in a way to allow interaction with the PDInteractables_ class of premades.

The Plugins folder contains pre-compiled libraries of libPD targeted for the Windows x64 platform.

PDPatches folder contains templates and sample Pure Data patches that can be used for testing and understanding the Unity to PD and PD to Unity communication. The sample patches contain various blocks towards programming real-time synthesis and Pure Data Vanilla compatible signal processing examples.

The scenes folder contains two Unity scenes, the BareBones scene, and the VRMI Creation Toolkit Demo scene. The BareBones scene contains all the necessary elements that must be set in a VRMI along with a sample interactable that creates a 440hz sinus beep in a Pure Data instance. This scene allows a developer to quickly set up a new scene by copying the listed gameObjects from it.

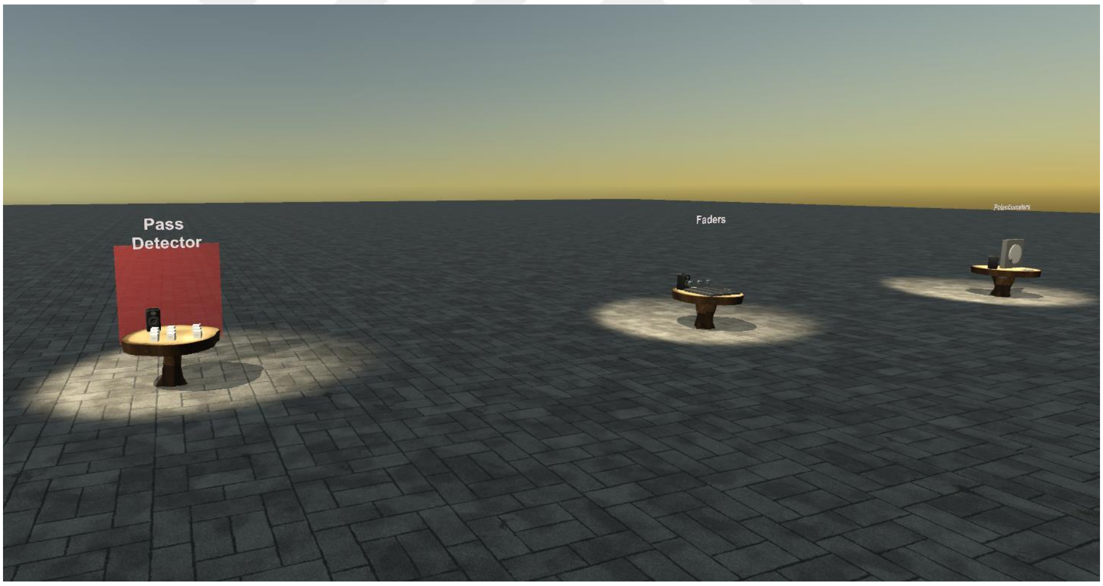

Figure 2.36 : Screenshot from VRMI Creation Toolkit.

The VRMI Creation Toolkit Demo Scene (Figure 2.36) showcases the functionality of the toolkit by the usage of the prefabs. All of the prefabs listed above are present in the demoscene, set up in a plain way to demonstrate functionality without clutter. This scene is created as a tutorial for the developers. Within this scene, developers can test the functionality of the pre-fabs and experience how the audio engine behaves. All of the gameObjects in this scene can also be copied over to a new scene easily.

All of the functionality and elements of the VRMI Creation Toolkit is described in detail in the VRMI Creation Toolkit Manual, which is bundled with the toolkit.

5.3 Object Hierarchy

There are three kinds of premades in VRMI Creation Toolkit. Interactables, PD Audio Sources, and PD Message server.

5.3.1 Interactables

Interactables are pre-made objects for various VR interaction methods. Their usages are demonstrated in the demo scene in detail. As seen in Figure 2.37, linkage to other elements is handled by drag and drop action to PDAudioSources. This particular example sends a bang on collision to the target PD Instance and patch with the send name r5 for activation and r5_0 on deactivation.

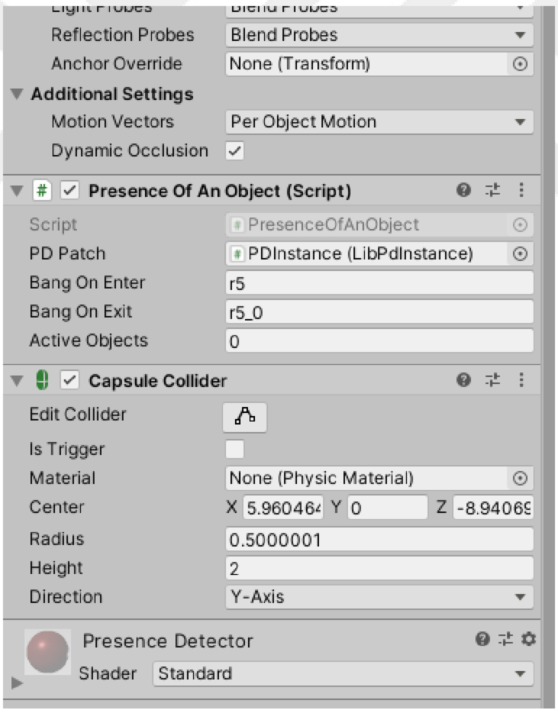

Figure 2.37 : Screenshot from VRMI Creation Toolkit, Unity Editor.

Several pre-made VR interactables are available. Examples can be seen in the following sub-chapters.

5.3.1.1 Presence detector

Presence detector detects the presence of an object inside an area, reports either a bang or float coordinates of the intersecting object.

5.3.1.2 Pass-thru detector

Pass-thru detector detects an object passing through a 3d area, reports either a bang or float coordinates of the intersecting object.

5.3.1.3 Rotary potentiometer

The rotary potentiometer is a VR grabbable potentiometer that reports back either float or int value of the rotation. Limits and ranges can be set by the user from the editor.

5.3.1.4 Linear fader

Linear fader is a VR grabbable fader that reports back either float or int value of the fader. Limits, movement, and range can be set by the user from the editor.

5.3.1.5 Buttons

VR Interactable buttons can be set to both latching and non-latching types. Buttons report back either bool value in int format or simple bang.

5.3.1.6 Distance detector

The distance detector is a two-part object. Composed of two prefabs, the distance detector calculates the distance between prefab 1 and prefab 2 and reports it as a float value.

5.3.2 PD Audio Source

PD Audio source carries the script to initialize libPD and launch the target PD patch. Although unused, an audio source element must be active for the script to function. As seen in Figure 2.3, the target PD patch is again selected from the editor.

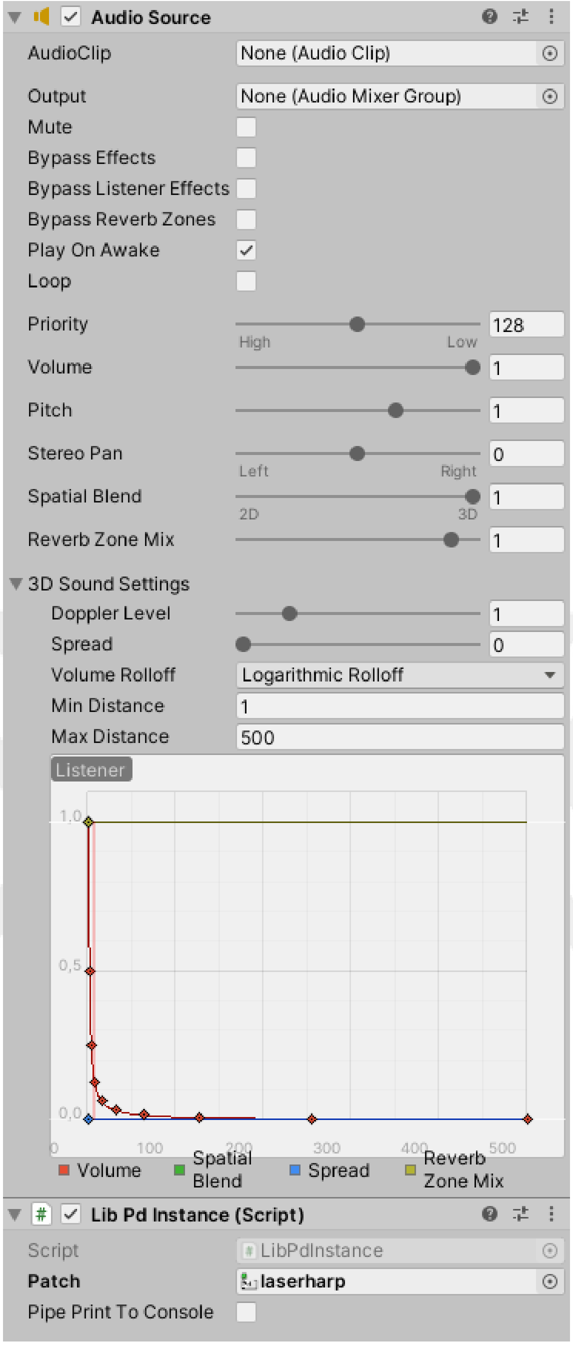

Figure 2.38 : Screenshot from VRMI Creation Toolkit, Unity Editor – PD Audio Source.

5.3.3 Template PD patch

Template PD patch, as seen in Figure 2.39, consists of an output and receives. Receive names that correspond to the entered values in the Unity editor. Type checking must be handled on the Unity side.

Figure 2.39 : Screenshot from VRMI Creation Toolkit, Template PD Patch

5.4 Capabilities and Benefits of the VRMI Creation Toolkit

VRMI Creation Toolkit can accomplish the following:

- A VRMI can be created within minutes.

- A VRMI can be created without writing a single line of code

- The created application can be compiled into a single application, therefore the result is easily publishable.

- The VRMI Creation Toolkit doesn't limit any of the Unity software development functionality.

- Pure Data patches can be run inside the application. The patches live on gameObjects instances and can output sound as point sources in virtual 3d space.

- Multiple instances of Pure Data patches can be run simultaneously.

- Unity's audio capabilities such as spatialization, reverb zones, and effects and still be used with the Pure Data instances. Handling these in Unity greatly reduces the workload as otherwise, all of these spatialization, reverb, and DSP must be run in every Pure Data instance, and gameObject position must be frequently sent to the instance.

- Running Pure Data instances doesn't replace the Unity sound engine. Simple audio interactions such as sample playback for interface events, etc. can be offloaded to the Unity Audio Engine for simplicity. Therefore, audio programming is needed only for the VRMI elements that require it.

- Common VR interaction is supplied as prefabs and greatly reduces the workload of the developer.

- All of the dependencies have licenses that allow them to be used freely even in a commercial setting, therefore the resulting application is free from licensing issues.

5.5 Initial Setup

The initial setup is simple but the procedure must be followed strictly. VRMI Creation Toolkit depends on the Steam VR plugin for Unity. The user must first download and import SteamVR (v2.60+) into the Unity project. After that default Steam VR bindings must be populated from the SteamVR input window. After importing the Steam VR plug-in, VRMI Creation Toolkit can be imported to the project and simple drag and dropping of the pre-mades are sufficient to continue.

6. CONCLUSIONS

This study aimed to enhance virtual reality music instrument design by proposing a set of design guidelines and a software framework that allows real-time synthesis in a publishable form. When we look at the various examples of previous acoustic instruments, digital music instruments, virtual reality music instruments, and the evaluation and design of these instruments, we can see that there is a big ambiguity on what is a better instrument both from the consumer and academic point of view. With examples from the past, we can see that most of the traditional metrics like expressivity, difficulty, learning curve, tonal palette, and input/output complexity are metrics that have outcomes that can have both positive and negative impacts on the instrument. For example, an instrument with a very high difficulty can have a positive impact as being able to play such an instrument can be rewarding – it also has a negative impact as it can frustrate many players. If we compare the musical instrument to a tool that accomplishes a certain task, metrics like these could have worked – who wouldn't want a tool that can accomplish anything, in any way, as the user desires, with the push of a single button. A musical instrument on the contrary doesn't exist necessarily to accomplish certain tasks. From history, we can gather that musical instruments are picked up both as tools and objects of entertainment. The delicate balance of fun factor also affects the outcome of the usage of the instrument when it is used as a tool. In many examples, we can see that aerobic difficulty can be observed by the audience and it is rewarding for not just the performer. Another way to discuss what a good instrument is by looking at it from the viewpoints of different people. Is the instrument good for the performer, audience, designer, builder, marketer, or the investor? There are many ways to look at an instrument and derive what a good instrument is.

Above all metrics, survivability is a concrete metric that is very hard to argue against. In order for an instrument to exist, it has to first be built, then exist. Existing and being picked up is healthy for an instrument as it creates a feedback loop between the user and the creator and creates ways for evolutionary improvements as well as repertoire for the instrument.

When we look at history, survival is a common trait amongst all the instruments that we can call successful, and survivability had always been a big concern amongst new instruments. With these findings, we established that in order to enhance VRMI design, survivability is a valid concern and must be addressed.

The evolution of digital music instruments also gave us hints on the motivation behind new instruments. It can be observed that a great portion of digital musical instrument design tackled the issue of the sound palette due to reducing costs and technical limitations. Currently, a digital luthier has all the processing power and tools to create an instrument with virtually any palette desired. Sound generation is no longer an issue, but a preference. After tackling the issue of sound generation, DMI design evolved into finding a good formula of fun, expression, difficulty, and playability. Again, when we look at offerings that were picked up by the public, there is a big ambiguity of the importance amongst these metrics and there is no perfect formula for the good instrument. In this era, we can again depend on the survivability metric as a valid concern as most of the instruments that are considered successful are the ones that are picked up.

The survivability of an instrument depends on many aspects and is virtually incalculable. While there is no formula for an instrument that will definitely survive, an instrument that will not survive can be formulated as there are several solid necessities for survival. In terms of software music instruments, the most obvious one of such necessities is publishability. If an instrument can’t be published and distributed, it will only serve the creator. Time has shown us that it is often the case that such unpublished instruments are even forgotten by their creators most of the time.

When we look at VRMIs, the concern of survivability, due to the publishability, is again a valid concern. Here we can also observe a very important trend in VRMIs when we split them into two groups. Academic instruments, and published instruments. Academic VRMIs are instruments that are either not available to the public, or available with very obscure means. Often the instrument is a combination of the usage of several software and can not be directly run by an end-user even if a download is possible. While these instruments only exist in publications, we can observe that their sound palette and expressive capabilities are greater than the published instruments we can compare them to.

On the other hand, when we look at the published instruments, that are available to the public on popular platforms such as steam, we can see that the offerings are limited on what they can accomplish sonically due to the absence of real-time synthesis.

VR technology in its current form is dominantly geared towards gaming. When we look at the popular software development platforms for gaming, we can observe that while these platforms are great for developing almost anything, they offer close to nothing for music technology researchers in terms of audio programming. While the option to do low-level audio programming is there, established high-level languages such as Max Msp, Pure Data, or SuperCollider are not easily integrated. This creates a distinct gap between academic instruments and current commercial offerings as academic instruments often employ such high level, familiar languages as their sound engines, and commercial offerings skip synthesis all together and rely on simple sample playback.

The current commercially available VR technology is deeply rooted in gaming. The available headsets are designed towards gaming, software libraries are geared towards gaming, libraries are geared towards gaming and the controllers are geared towards gaming. While we can fight this predominantly gaming focused state of VR, we can also embrace it and play amongst it.

If we have the survivability as our most dominant concern towards a VRMI, fighting is a guaranteed loss. There is an already established user base of current commercially available VR headsets, which is growing at a steady pace. Embracing gaming is my first strategy towards VRMIs that can survive. Instead of using custom controllers or hardware, the already established VR hardware is a viable choice for VRMIs, albeit the limitations. Requiring custom hardware can create a cost barrier that can outweigh any lifted limitations in terms of survivability.

Because of this concern, this study focuses on creating VRMIs on the already established Steam VR platform which offers several advantages in terms of tracking compared to the competition as discussed earlier.

Another benefit of embracing gaming comes from the non-conflicting concerns of VR issues. The gaming industry tackles the same technical issues, like VR sickness, latency, field of view, tracking, etc; the previous research argues positively for VRMIs (Serafin, 2016, Abrash, 2014).

While embracing gaming, we should not stop at just borrowing hardware and software and look at video game design principles too. There are many ways video game design principles can enhance DMIs and VRMIs. Ludic experience is also a part of music instruments and gamification can greatly improve the fun factor. As discussed earlier, strategies from gaming can be used to artificially alter classic metrics such as learnability without altering any other metric negatively.

Likewise, many of the design considerations I proposed have roots in gaming. While sharing the technology and platform with gaming, many of these similarities are hard to ignore, and evaluating VRMIs as video games as well as musical instruments can be a valid choice. As can be observed from the past, the line between a toy and a serious musical instrument is only drawn by the intent of the performer. For many, serious musical instruments are expensive toys, and for some video games that were never designed to be a musical instrument are instruments (Url-13). Looking at music as a ludic experience is a valid point of view and should not be skipped. (Moseley, 2016). Maybe the world's next most picked up instrument will be a VRMI that is only part of an MMORPG game. Future can surprise us.

Apart from my framework of design considerations, I identified that software topology is a root issue in current VRMI offerings and created the VRMI Creation Toolkit. VRMI Creation Toolkit aims to harvest already established audio programming skills amongst music technology researchers by employing Pure Data as the synthesis engine.

As success in terms of survivability can’t be guaranteed with a formula, I took nature’s approach to limited chances and employed brute forcing as a strategy. If you can’t increase chances, increase tries. VRMI Creation Toolkit establishes this by reducing the development time tremendously, while maintaining publishability, maintaining platform conformity, and harvesting existing skillsets of possible developers. By reducing the cost of development, it gives incentive to the developer to try more.

I hope that VRMI Creation Toolkit will be picked up by others and we will see an increase of variety in future VRMI offerings. VRMI Creation Toolkit is open source and I hope that it will also grow with community support.

REFERENCES

Abrash, M (2014). *Talk from Steam Dev Days*, Retrieved on August 4, 2019 from http://media.steampowered.com/apps/abrashblog/Abrash%20Dev%20Days%202014.pdf

Anthes, C., García-Hernández, R. J., Wiedemann, M., & Kranzlmüller, D. (2016, March). State of the art of virtual reality technology. *2016 IEEE Aerospace Conference* (pp. 1-19). IEEE.

Barbosa, J., Malloch, J., Wanderley, M. M., & Huot, S. (2015). What does" Evaluation" mean for the NIME community?.

Berdahl, E., Smith, J. O., Weinzierl, S., & Niemeyer, G. (2012). Force-sensitive detents improve user performance for linear selection tasks. *IEEE Transactions on Haptics*, *6*(2), 206-216.

Blaha, J., & Gupta, M. (2014). Diplopia: A virtual reality game designed to help amblyopics. *2014 IEEE Virtual Reality* (VR) 63-164.

Chowning, J., & Bristow, D. (1986). FM theory & applications. Tokyo: Yamaha Music Foundation.

Cipresso, P., Giglioli, I. A. C., Raya, M. A., & Riva, G. (2018). The past, present, and future of virtual and augmented reality research: a network and cluster analysis of the literature. *Frontiers in psychology*, 9, 2086.

Dempsey, P. (2016). The teardown: HTC Vive VR headset. *Engineering & Technology*, *11*(7-8), 80-81.

Dubrowski, D, Peterson, S, Gilmore H. (27 Mar 2019)"Playing for Their Lives: How a 2001 Video Game Is Feeding Venezuela." *The National Interest*. Retrieved on June 22, 2020, from https://nationalinterest.org/feature/playing-their-lives-how-2001-video-game-feeding-venezuela-49187

Ebenholtz, S. M. (1992). Motion sickness and oculomotor systems in virtual environments. *Presence: Teleoperators & Virtual Environments*, *1*(3), 302-305.

Greene, K. (27 March 2018) *How VR Porn Is Secretly Driving The Industry*. Retrieved January 7, 2019, from https://www.vrfocus.com/2018/04/how-vr-porn-is-secretly-driving-the-industry/

Heeks, R. (2008). Current Analysis and Future Research Agenda on'Gold Farming': Real-World Production in Developing Countries for the Virtual Economies of Online Games. *Development informatics working paper*, (32).

Heeks, R. (2009). Understanding" gold farming" and real-money trading as the intersection of real and virtual economies. *Journal For Virtual Worlds Research*, *2*(4).

Heilig, M. L. (1962). *U.S. Patent No. 3,050,870.* Washington, DC: U.S. Patent and Trademark Office.

Heilig, M. L. (1992). El cine del futuro: The cinema of the future. Presence: Teleoperators & Virtual Environments, *1*(3), 279-294.

Howard, D. M., & Angus, J. (2017). *Acoustics and psychoacoustics.* Taylor & Francis.

Jensenius, A. R., & Lyons, M. J. (Eds.). (2017*). A NIME Reader: Fifteen Years of New Interfaces for Musical Expression* (Vol. 3). Springer.

Jordà, S. (2005). *Digital Lutherie Crafting musical computers for new musics' performance and improvisation*. Department of Information and Communication Technologies.

Jordà, S. (2004, June). Digital instruments and players: part I---efficiency and apprenticeship. *Proceedings of the 2004 conference on New interfaces for musical expression* (pp. 59-63).

Lilly, P. (25 July 2017) *HTC talks Vive and lack of AAA games for VR.* Retrieved January 5, 2019, from https://www.pcgamer.com/htc-talks-vive-and-lack-of-aaa-games-for-vr/

Manning, P. (2013). *Electronic and computer music*. [electronic resource] (Fourth edition.). Oxford University Press.

Machkovech, S (2018) *Oculus Quest, a fully wireless VR headset, shipping spring 2019 for $399* Retrieved on December 10, 2019, from: https://arstechnica.com/gaming/2018/09/oculus-quest-a-fully-wireless-vr-headset-shipping-spring-2019-for-399/

Mazuryk, T., & Gervautz, M. (1996). Virtual reality-history, applications, technology and future.

Morreale, F. (2017). Design for longevity: Ongoing use of instruments from NIME 2010-14.

Moseley, R. (2016). *Keys to Play*. University of California Press.

Moore, A. G., Howell, M. J., Stiles, A. W., Herrera, N. S., & McMahan, R. P. (2015, March). Wedge: A musical interface for building and playing composition-appropriate immersive environments. *In 2015 IEEE Symposium on 3D User Interfaces (3DUI)* (pp. 205-206). IEEE.

Rogers, J. (22 May 2017). *Study Predicts Fewer Than 10 Million Monthly U.S. VR Headset Users This Year, 17 Million by 2019*. Retrieved December 25, 2017, from http://variety.com/2017/digital/news/vr-headset-data-mau-2017-2019-1202440211/

Rogers, S. (21 June 2019) *2019: The Year Virtual Reality Gets Real*. Retrieved on December 10, 2019, from https://www.forbes.com/sites/ solrogers/2019/06/21/2019-the-year-virtual-reality-gets-real/

Roettgers, J. (14 May 2018) *Oculus Go Porn: Adult Industry Has High Hopes for Facebook's New VR Headset* on June 25, 2019, from https://www.nasdaq.com/articles/oculus-go-porn-adult-industry-has-high-hopes-facebooks-new-vr-headset-2018-05-24

Rosenberg, A. (28 January 2018). *Virtual reality's moment looks to be over in gaming, at least for now*. Retrieved January 5, 2019, from https://mashable.com/2018/01/24/virtual-reality-gaming-loser-gdc-2018-survey/

Ryge, A., Thomsen, L., Berthelsen, T., Hvass, J. S., Koreska, L., Vollmers, C., ... & Serafin, S. (2017, March). Effect on high versus low fidelity haptic feedback in a virtual reality baseball simulation. *In 2017 IEEE Virtual Reality (VR)* (pp. 365-366). IEEE.

Sachs, C. (2012). *The History of Musical Instruments* Retrieved from https://books.google.com.tr/books?id=W615TIDz97UC (Original work published 1941)

Sánchez Carranco, J. (2018). Vibrating instruments in virtual reality: A cohesive approach to the design of virtual reality musical instruments. *Phd. Dissertation.* Retrieved on 10 June 2020, from: https://aaltodoc.aalto.fi/bitstream/handle/123456789/35438/master_S%C3%83%C2%A1nchez_Carranco_Juan_2018.pdf

Sarier, O. (2014). Rub Synth: A Study of Implementing Intentional Physical Difficulty Into Touch Screen Music Controllers. In NIME (pp. 179-182).

Schaller, R. R. (1997). Moore's law: past, present and future. *IEEE spectrum*, 34(6), 52-59.

Serafin, S., Erkut, C., Kojs, J., Nilsson, N. C., & Nordahl, R. (2016). Virtual reality musical instruments: State of the art, design principles, and future directions. *Computer Music Journal, 40*(3), 22-40.

Serafin, S., Adjorlu, A., Nilsson, N., Thomsen, L., & Nordahl, R. (2017, March). Considerations on the use of virtual and augmented reality technologies in music education. In 2017 *IEEE Virtual Reality Workshop on K-12 Embodied Learning through Virtual & Augmented Reality (KELVAR)* (pp. 1-4).

Serafin, S., Adjorlu, A., Andersen, L., & Andersen, N. (2019). Singing in Virtual Reality with the Danish National children's choir. *14th International Symposium on Computer Music Multidisciplinary Research* (p. 241).

Steinicke, F. (2018). *Being Really Virtual Immersive Natives and the Future of Virtual Reality*. Cham: Springer International Publishing.

Sutton, K. (19 April 2019) *Can Porn Save VR?.* Retrieved on May 25, 2019, from https://www.adweek.com/digital/can-porn-save-vr/

Taheri, S. M., Matsushita, K., & Sasaki, M. (2017). Virtual reality driving simulation for measuring driver behavior and characteristics. *Journal of transportation technologies*, 7(02), 123.

Tanaka, A. (2000). Musical performance practice on sensor-based instruments. *Trends in Gestural Control of Music*, 13(389-405), 284.

Url-1 <*http://www.unity.com*>, last accessed 10 December 2020.

Url-2 < *https://www.unrealengine.com/*>, last accessed 10 December 2020.

Url-3 < https://docs.unity3d.com/Manual/AudioOverview.html/>, Audio Overview, *Unity User Manual* last accessed 10 December 2020.

Url-4 <https://idialab.org/chromachord-a-virtual-musical-instrument/> last accessed 10 December 2020.

Url-5 <http://www.coretet.io/> last accessed 10 December 2020.

Url-6 < http://sopi.aalto.fi/research/vivr/> last accessed 10 December 2020.

Url-7 <https://store.steampowered.com/app/60620/EXA_The_Infinite_Instrument/> last accessed 10 December 2020.

Url-8 <https://store.steampowered.com/app/685240/Paradiddle/> last accessed 10 December 2020.

Url-9 <https://store.steampowered.com/app/1100920/DrumMasterVR/> last accessed 10 December 2020.

Url- 10 <https://store.steampowered.com/app/536370/Percussive_VR/> last accessed 10 December 2020.

Url-11 <https://www.ultraleap.com/product/leap-motion-controller/> last accessed 10 December 2020.

Url-12 <https://www.oculus.com/blog/introducing-hand-tracking-on-oculus-quest-bringing-your-real-hands-into-vr/?locale=en_US> last accessed 10 December 2020.

Url-13 <https://www.youtube.com/watch?app=desktop&v=mjLDM1AY1-E> last accessed 10 December 2020.

Url-14 <https://www.virtuix.com/> last accessed 10 December 2020.

Wanderley, M. M., & Orio, N. (2002). Evaluation of input devices for musical expression: Borrowing tools from hci. *Computer Music Journal*, *26*(3), 62-76.

Weidenaar, Reynold (1995). *Magic music from the telharmonium.* LONDON : The Scarecrow Press,

Zimmerman, M. (2019) *Conscious Existence - A Journey Within.* Available from: https://store.steampowered.com/app/1093330/Conscious_Existence__A_Journey_Within/

CURRICULUM VITAE

Name Surname : Ozan SARIER

Place and Date of Birth :

E-Mail :

EDUCATION :

- **B.Sc.** : 2008, İstanbul Bilgi University, Faculty of Science and Literature, Music Department
- **M.Sc.** : 2012, İstanbul Technical University, Faculty of Social Sciences, Music Department

PROFESSIONAL EXPERIENCE AND REWARDS:

- (2004- Present) Sound Engineer and Designer for various projects. Music albums, feature films, tv shows and commercials, field recording, sound design, classical music production and live sound.
- (2012- Present) Researcher at Istanbul Technical University - Center For Advanced Studies in Music
- (2019-Present) Instructor at Bahçeşehir University
- (2014) Andante Magazine – 2014's best recording award with Izmir Barok's - Music in European and Ottoman Palaces from the 16th Century until the first half of the 18th Century album
- (2016) Andante Magazine – 2016's best recording with Aydın Karlıbel's Doğu Masalları Album

PUBLICATIONS, PRESENTATIONS AND PATENTS ON THE THESIS:

- **Sarıer O.**, and Karadoğan C**.** 2018: Borrowing Tools From Video Game Industry To Create A Virtual Reality Music Instrument: VRMine – Porte Akademik Müzik ve Dans Araştırmaları Dergisi – İstanbul.