Design Architecture in Virtual Reality

by

Anisha Sankar

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Architecture

Waterloo, Ontario, Canada, 2019

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Architectural representation has newly been introduced to Virtual Reality (VR) technology, which provides architects with a medium to showcase unbuilt designs as immersive experiences. Designers can use specialized VR headsets and equipment to provide a client or member of their design team with the illusion of being within the digital space they are presented on screen. This mode of representation is unprecedented to the architectural field, as VR is able to create the sensation of being encompassed in an environment at full scale, potentially eliciting a visceral response from users, similar to the response physical architecture produces. While this premise makes the technology highly applicable towards the architectural practice, it might not be the most practical medium to communicate design intent. Since VR's conception, the primary software to facilitate VR content creation has been geared towards programmers rather than architects. The practicality of integrating virtual reality within a traditional architectural design workflow is often overlooked in the discussion surrounding the use of VR to represent design projects.

This thesis aims to investigate the practicality of VR as part of a design methodology, through the assessment of efficacy and efficiency, while studying the integration of VR into the architectural workflow. This is done by examining the creation of stereoscopic renderings, walkthrough animations, interactive iterations and quick demonstrations as explorations of common architectural visualization techniques using VR. Experimentation with each visualization method is supplemented with a documentation of the VR scene creation process across an approximated time frame to measure efficiency, and a set of evaluation parameters to measure efficacy. Experiments either yielded the creation of a successful experience that exceeded the time constraints a common fast-paced architectural firm might allow for the task (low efficiency) or created a limiting experience where interaction and functionality were not executed to meet the required industry standards (low efficacy). This resultant impracticality based on time and effort, demonstrates that a successfully immersive VR simulation is not produced simplistically in VR; requiring a great deal of thought to be placed into design intent. Although impractical, documentation suggests that the user experience of creating VR content might be able to engage new ways of design thinking and impact the way architects conceptualize space, encouraging further research.

Acknowledgements

I would like to express my sincerest gratitude to my supervisor, David Correa. Thank you for your extensive support, patience, insight, and expertise. You generously helped me sort through ideas with countless discussions, and provided uplifting advice in moments of self-doubt. This thesis would not have been possible without your guidance.

Thank you to my committee member John McMinn, for encouraging me to objectively form a critique on the topics of this thesis, in a consideration of every architect's skillset and the design practice at-large.

Don McKay, thank you for helping me build both the foundations of this thesis and my knowledge of design. You introduced me to ideas I will continue to ponder on beyond this research.

Jane Hutton, thank you for engaging in the discussion of technology shaping the way architects approach design. Your valuable insight helped me hone in on the conlcusions of my work.

Miguel Guitart and Ivan Vasyliv, thank you for sharing in the conversation of virtual reality and architecture with me, and providing your design expertise.

To my talented friends, Matt, Amanda, Teresa, Victoria, Michelle, Sameerah, and Marc, thank you for your company, support and motivation. You've all taught me valuable life lessons and inspired me immeasurably.

To my best friend, Pavithra, thank you for your uwavering support, countless suggestions, insight and generosity, no matter the time of day.

To my partner Akshay, thank you for always being there for me, and helping me get through each deadline.

Last but not least, to my parents and my brother, thank you for your constant love, support and patience. I owe everything to you, and I am eternally grateful.

To the designers that put in the effort and time.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

VR	Virutal Reality
HMD	Head-Mounted Display
AR	Augmented Reality
MR	Mixed Reality
XR	Extended Reality
2D	Two-Dimensional
3D	Three-Dimensional
4D	Four-Dimesional
FOV	Field-of-View
IOD	Interocular Distance
PC	Personal Computer
FPS	Frames-Per-Second
DoF	Degrees-of-Freedom

> “*Architecture is like a mythical fantastic. It has to be experienced. It can’t be described. We can draw it up and we can make models of it, but it can only be experienced as a complete whole.*”
>
> *- Maya Lin*

Chapter 1: Introduction

Defining “Virtual Reality”

The invention of virtual reality (VR) can be traced back to the novel “Pygmalion’s Spectacles” by Stanley G. Weinbaum in 1935. This story depicts the invention of special goggles that play movies with which the wearer of the goggles can interact.[1] This idea was further developed in 1955 by cinematographer Morton Heilig, who invented a multi-sensory apparatus called the “Sensorama”.[2] This device included a three-dimensional stereoscopic display, speakers and haptics feedback through the vibration of the user’s seat. Morton Heilig is hence named the “Father of Virtual Reality” for his invention.[3] This arcade machine-style mechanical cabinet was essentially built to stimulate the senses and is representative of many features present in modern day head-mounted displays (HMDs). Post-“Sensorama,” the first ever HMD was created, which built on the same principles, providing a user with binocular 3D visuals and sound.

The term “virtual reality” has multiple definitions. The Merriam-Webster dictionary defines virtual reality as: “an artificial world that consists of images and sounds created by a computer and that is affected by the actions of a person who is experiencing it.”[4] Colloquially, the phrase often represents an existential artificial world. The term “virtual reality” is an oxymoron, however, in the sense that it implies that such a reality is not real – that it is inherently “virtual”. Virtual Reality is, in fact, a very real thing; it is simply a reality that must be distinguished from being classified as not real, and instead, as non-physical. Hence, this thesis focuses on the technical definition of Virtual Reality, as an immersive artificial environment that is viewed through specialized equipment. This technical definition states that virtual reality is a simulation technology implemented through the use of headsets and physical props, such as controllers and sensors, to impose a realistic multi-projected environment on a user. VR stimulates and brings a user into a rendered scene by allowing them to look around, navigate and interact with objects in the artificial environment.

---

[1] Jeremy Norman, ““Pygmalion’s Spectacles,” Probably the First Comprehensive and Specific Fictional Model for Virtual Reality (1935): HistoryofInformation.com,” accessed October 23, 2018, http://www.historyofinformation.com/expanded.php?id=4543.

[2] Jon Turi, “The sights and scents of the Sensorama Simulator: Time Machines,” accessed October 23, 2018, https://www.engadget.com/2014/02/16/morton-heiligs-sensorama-simulator/.

[3] Ibid.

[4] Merriam-Webster, “Virtual Reality,” accessed October 10, 2018, https://www.merriam-webster.com/dictionary/virtual%20reality.

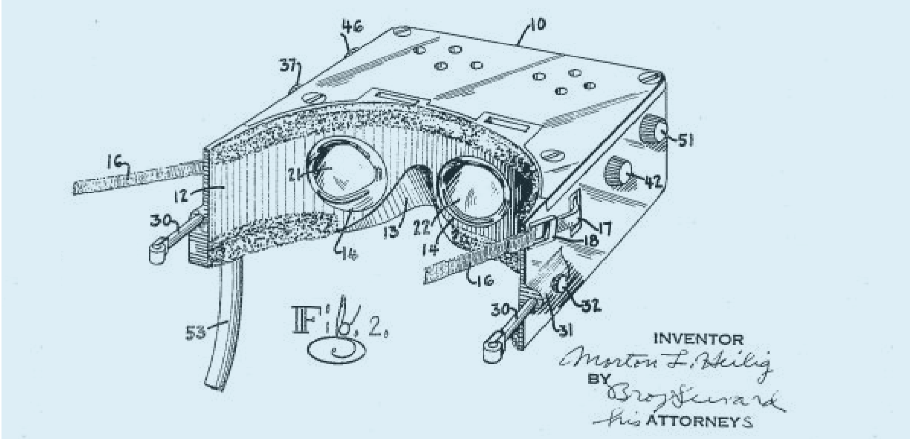


figure 1.01 Drawing of Morton Heilig’s Speciality “Telesphere Mask”

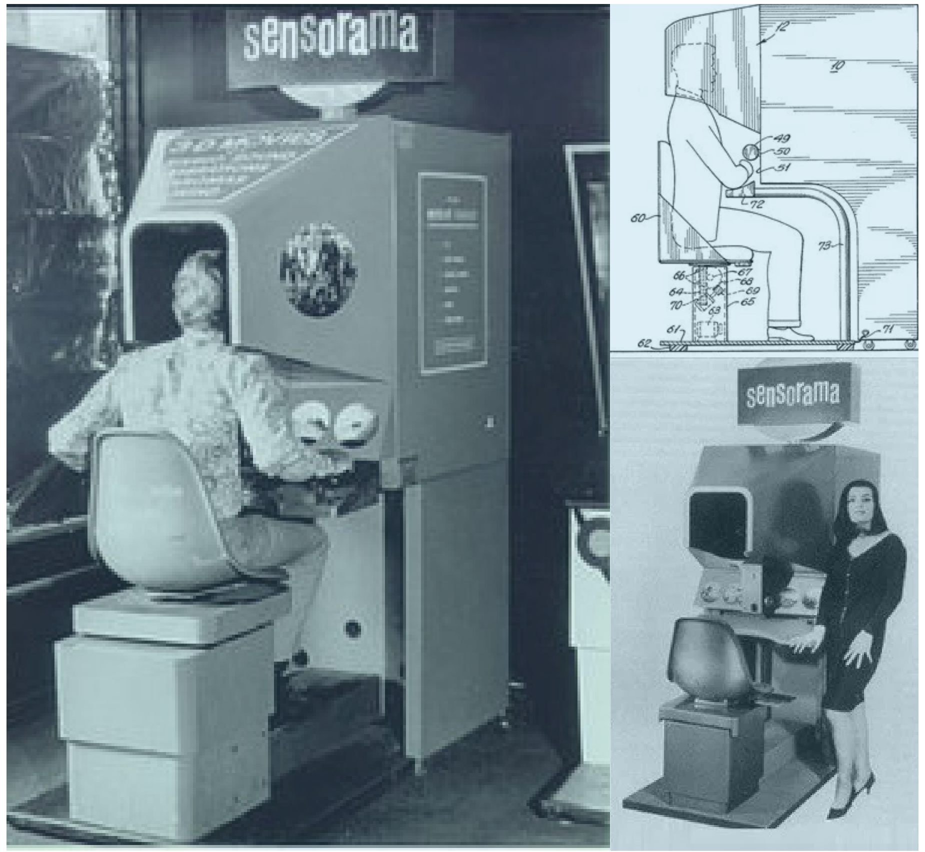
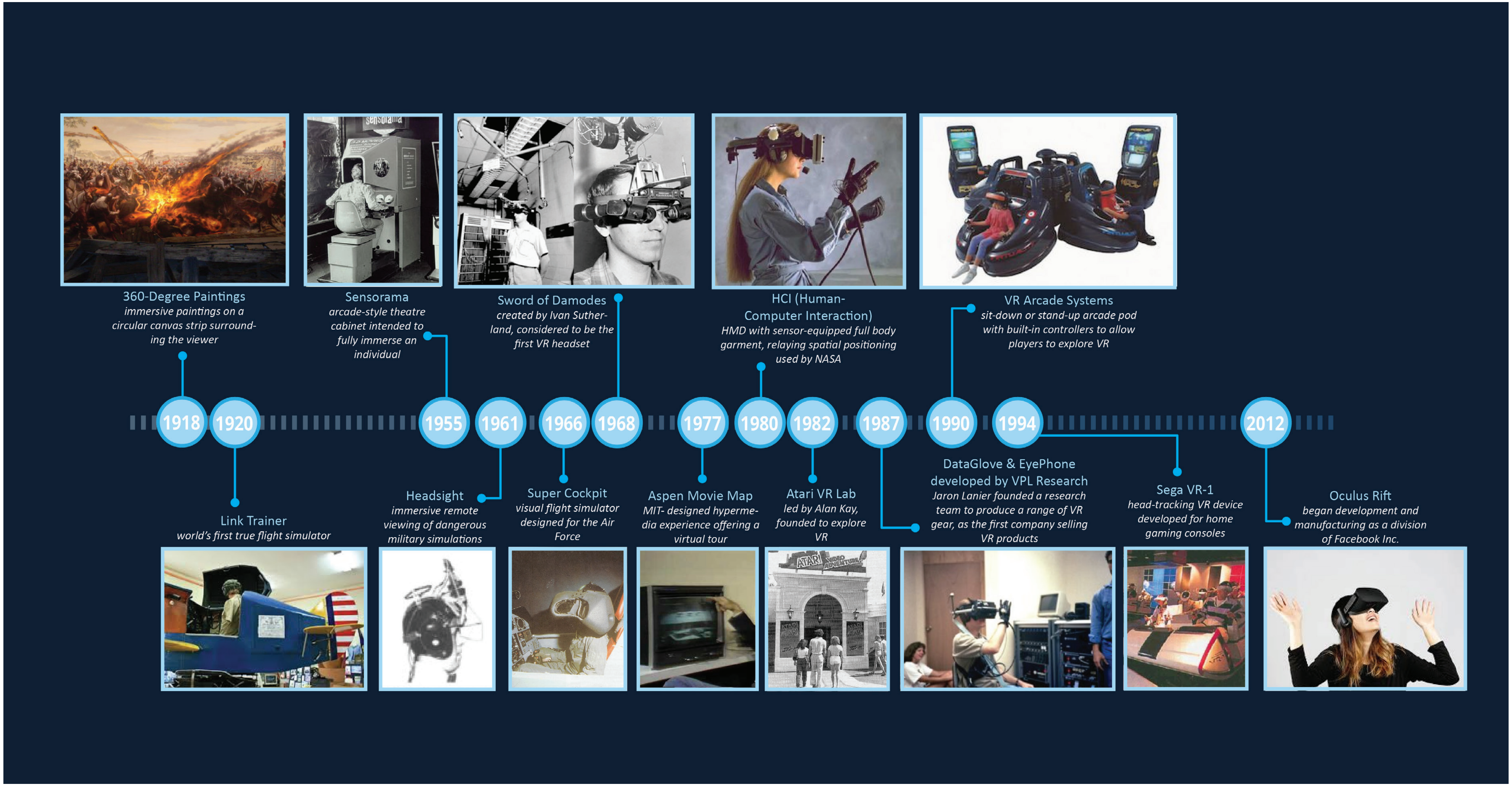


figure 1.02 Sketch (top right) and pictures (bottom) of the Sensorama Simulator



1918

360-Degree Paintings
immersive paintings on a circular canvas strip surrounding the viewer

1920

Link Trainer
world's first true flight simulator

1955

Sensorama
arcade-style theatre cabinet intended to fully immerse an individual

1961

Headsight
immersive remote viewing of dangerous military simulations

1966

Super Cockpit
visual flight simulator designed for the Air Force

1968

Sword of Damodes
created by Ivan Sutherland, considered to be the first VR headset

1977

Aspen Movie Map
MIT- designed hypermedia experience offering a virtual tour

1980

HCI (Human-Computer Interaction)
HMD with sensor-equipped full body garment, relaying spatial positioning used by NASA

1982

Atari VR Lab
led by Alan Kay, founded to explore VR

1987

DataGlove & EyePhone developed by VPL Research
Jaron Lanier founded a research team to produce a range of VR gear, as the first company selling VR products

1990

VR Arcade Systems
sit-down or stand-up arcade pod with built-in controllers to allow players to explore VR

1994

Sega VR-1
head-tracking VR device developed for home gaming consoles

2012

Oculus Rift
began development and manufacturing as a division of Facebook Inc.

figure 1.03 Timeline of Historical Events Suurounding VR Creation & Implementation

Set of Stock Images Used in VR Articles to Increase the "Promise of VR":

figure 1.04 Image depicting VR's "Amazing" Shock-Factor

figure 1.05 "Amazing New Media & Entertainment VR Experiences..."

figure 1.06 "The Amazing Ways Honeywell Is Using Virtual And Augmented Reality To..."

VR environments often integrate transmissions in the form of vibrations and sensations to create haptic experiences for the user within the artificial scene.[1] Although VR is also often used as an "umbrella term" for immersive experiences, within this thesis, it refers specifically to the unique experience of placing a viewfinder headset on a user to allow them to experience a simulated digital scene. VR environments are typically closed off from the physical world as headsets implement the use of headphones to block out sound and the viewfinder is placed directly over the eyes to block out sight. Although VR hardware may seem futuristic and the software quite complex, the end result is simply the production of two shifting images that are combined with lenses to produce the simulation of a virtual environment.[2] Therefore, these environments can be seen as "wholly new" even though the simulation environments could be based on real places that exist in the world.[3]

Virtual reality previously existed solely in the laboratory sector, specifically as a medium for research and development purposes in the educational, medical and rehabilitative fields.[4] In the past decade, the breakthrough of accessible virtual reality into the public realm has spurred a deep interest in the technology by designers, architects, and visualization specialists.[5] Consumer adoption of virtual reality technology has been eagerly brought forth by the entertainment industry, which sells the public on "the promise that VR. holds".[6] This promise is represented in the themes of blockbuster movies, books, and TV shows within the last two decades (such as Ready Player One, the Matrix, Star Trek, and Black Mirror). These media releases present utopias of almost magically-immersive experiences. The public has been sold on the principle of simplicity in virtual reality, by placing advanced panoramic viewfinder goggles on their heads to be instantly teleported into a "fully realized" holographic scenes. Any room can be created as a virtual environment into which a user can step, and which can simulate the presence of actually being within that environment. Hence, VR is now seen as a technology to be heralded in the next decade with a transcendent impact on the way human-beings work, perceive entertainment and communicate ideas.

---

[1] Paul Mealy, Virtual & augmented reality for dummies, 1st edition (Indianapolis IN: John Wiley and Sons, 2018).

[2] Michael Totzke, "The Real and Virtual Norm Li," accessed November 9, 2018, https://www.canadianinteriors.com/features/real-virtual-norm-li/.

[3] Paul Mealy, Virtual & augmented reality for dummies, 1st edition (Indianapolis IN: John Wiley and Sons, 2018).

[4] Ibid.

[5] Architizer Editors, "VR Is Now More Accessible Than Ever for Architects," accessed March 3, 2019, https://architizer.com/blog/practice/tools/vr-yulio/.

[6] Paul Mealy, Virtual & augmented reality for dummies, 1st edition (Indianapolis IN: John Wiley and Sons, 2018), 7.

It is even discussed that nearly anyone can envision a virtual reality scene, presenting the notion of the technology being simple to use and fundamentally carrying the ability to "change where we are headed as a society".[1] Most often, these grand claims are made exaggerating future possibilities. Although these claims are plausible, the challenge lies in that they are not commonly judged for practicality. In particular, within the field of architectural design, the implementation of VR is not currently routine, and its practicality has not been examined to date. The judgement of practicality involves understanding the implications of virtual reality, the status of its technological maturity and identifying its place within design.

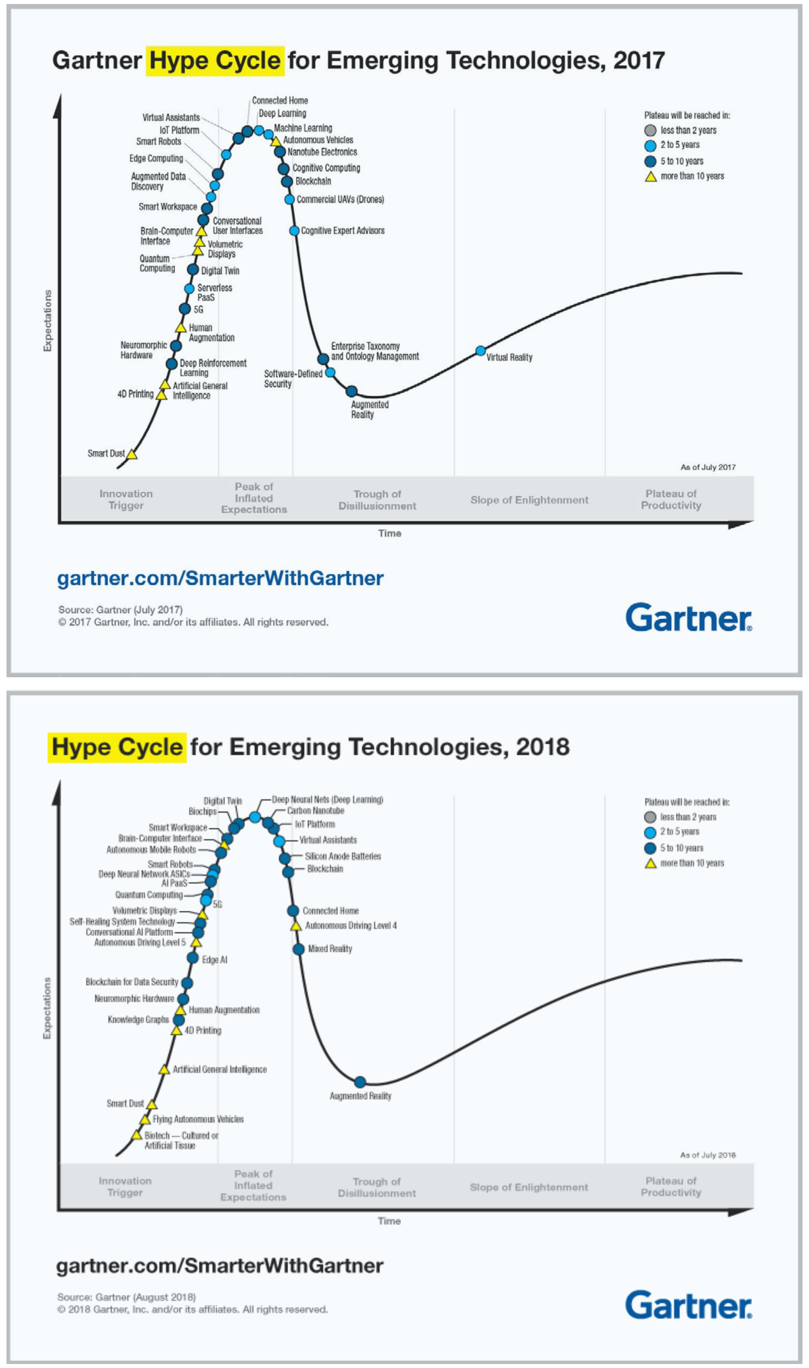
The Gartner Hype Cycle

The Gartner Hype Cycle outlines the transition periods of technological innovations as they evolve from conception to maturity.[2] The graphs developed by Gartner Inc. formulate how expectations around emerging technologies progress as they are released (beginning with the "Innovation Trigger") and reach full consumer adoption (shown in the "plateau of productivity"). The Hype Cycle charts expectations (y-axis) over time (x-axis) and asserts that technology goes through five stages. The first stage or the "innovation trigger" earmarks the release of the proof of concept and sparks media interest in the new technology. For virtual reality this occurred in March-April 2016 with the release of Oculus Rift and HTC Vive. Next, the "Peak of Inflated Expectations" occurs as media reports explore the possibilities and aforementioned "promise" of the new technology. In this stage, numerous companies also generally join in the bandwagon of exploring the technology. Yet, they often begin having expectations of the innovation that are higher than what it yields. Consequently, the "trough of disillusionment" phase occurs as interest in the technology plummets and the technology fails to meet expectations set out by the initial trigger and media buzz. Technologies that weather this trough pass to the final two stages; the "Slope of Enlightenment" and the "Plateau of Productivity". The final stages are indicative of a product achieving full adoption by the mass consumer base; technologies that have gone through the full cycle to reach this point include the Internet (world-wide web) and pre-2007 cellular mobility. Even though VR technology is making progress towards widespread consumer adoption, it is yet to achieve the critical mass adoption experienced by these technologies. In the 2017 Hype Cycle assessment, VR existed on the course to achieving the plateau stage, but the 2018 assessment does not recognize the existence of VR in the cycle.

---

[1] Paul Mealy, Virtual & augmented reality for dummies, 1st edition (Indianapolis IN: John Wiley and Sons, 2018), 8.

[2] Gartner Inc., "Gartner Hype Cycle: Interpreting Technology Hype," accessed October 18, 2018, https://www.gartner.com/en/research/methodologies/gartner-hype-cycle.



Source: Gartner (July 2017)
© 2017 Gartner, Inc. and/or its affiliates. All rights reserved.

Source: Gartner (August 2018)
© 2018 Gartner, Inc. and/or its affiliates. All rights reserved.

figure 1.07 Gartner inc.'s 2017 and 2018 Hype Cycle Predictions

The omission of VR in the 2018 Hype Cycle graph is partly due to the prediction that Mixed Reality (MR) will overtake VR and Augmented Reality (AR) in the upcoming 3 years. It is also because the 2nd and 3rd generation models of VR hardware include a more condensed setup and user interface improvements, which might re-enter the technology into the Hype Cycle as new innovations. As modern technology for VR presents itself and the uses for the technology is better understood, mainstream adoption will have a higher capacity of successful implementation. Edward Castronova, Professor of Media at Indiana University, explains that "the hype is invalid at the moment it happens, but it's pointing to something that will be happening in the future."[1] In 2017 VR was on the "Slope of Enlightenment" and was scheduled to reach mass consumer adoption in a predicted 4 years. As the technology evolves at the time of writing into new generations of devices and MR promises to be the "next big thing,"[2] predictions are being made for VR innovations to be short-lived, reaching strong consumer adoption only between 2023[3] (as the technology will be in a 3rd or 4th generation release and problems in the hardware of 2018 will be solved).

The challenge faced by VR is that it has been unable to achieve a frictionless and ubiquitous experience to the end user. The greatest success of technology will be when it becomes second nature to humans; to the effect of opening a new browser, checking emails and messaging friends. In the architect's perspective, ubiquitous concept-to-execution VR technology would be as simple as modeling on SketchUp, Rhino or Revit to conceptualize and then deploying a renderer such as V-Ray or 3dsMax to execute the design. If VR is able to achieve this level of ubiquity in allowing an architect to conceptualize and execute their designs, the technology will truly hit mainstream adoption and be successful in the modern architectural office. In order for VR to gain mass adoption, a new generation of headsets with easy-to-use functionality will be required. Although the first generation of headsets have been able to offer compelling immersive experiences, they have only been in the hands of designers with a keen interest in early explorations of technological design and VR development. The architectural design practice's usage of virtual reality heavily relies on the accessibility and ease of use of the technology, emphasizing the need for VR's technological maturity.

---

[1] Lesley Weidenbener, "2018 Innovation Issue: Is virtual reality already dead? Or just getting started?," June 1, 2018, accessed October 24, 2018, https://www.ibj.com/articles/69026-innovation-issue-is-virtual-reality-already-dead-or-just-getting-started.

[2] Jaron Lanier, Dawn of the new everything: Encounters with reality and virtual reality, First edition (New York: Henry Holt and Company, 2017).

[3] Jeremy Bailenson, Experience on demand: What virtual reality is, how it works, and what it can do / Jeremy Bailenson (New York, NY: W. W. Norton & Company, 2018).

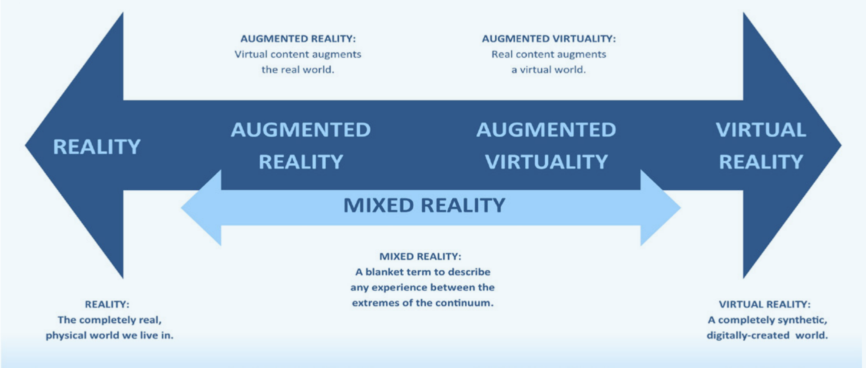


figure 1.08 Paul Milgram's Reality-Virtuality Continuum

Differentiating Between Virtual, Augmented and Mixed Realities

Extended reality (XR) is the umbrella term that encompasses the full spectrum of realities.[1] These realities, namely Virtual, Augmented and Mixed, are claimed to effectively "blur the line between reality and illusion, pushing limits of our imagination and granting us access to any experience imaginable."[2] This is accomplished differently by each of the three realities. With a strong understanding already that virtual reality comes from placing an HMD atop one's head (which include headphones), two main sensory sources (sight and sound) are taken over. Due to this, VR environments are typically closed off from the physical world. Augmented Reality is a way of viewing the real world directly (or via a device with a camera that creates a visual of the real surrounding world) and then "augmenting" this world with a computer-generated visual input. Virtual reality and augmented reality accomplish two very different things in two very different ways, despite the similar designs of the devices themselves. VR *replaces* reality, taking the user to a different place. AR *adds* to reality, projecting information on top of what one is already seeing. The virtual addition is created through still graphics, audio or video generation onto the real world. Augmented reality differs from VR in that it adds to the real world instead of creating every part of the visual experience from scratch. With augmented reality, data, instructional information, emotive objects and characters are animated over the real-world view, adding a layer on top of the existing real world.

---

[1] Paul Mealy, Virtual & augmented reality for dummies, 1st edition (Indianapolis IN: John Wiley and Sons, 2018).

[2] Jeremy Bailenson, Experience on demand: What virtual reality is, how it works, and what it can do / Jeremy Bailenson (New York, NY: W. W. Norton & Company, 2018).

AR is often implemented through smaller devices such as a mobile phone or tablet. An extremely successful app called Pokémon GO is an example of AR, as it involves virtual characters augmented into the real world and seen through the phone's camera. Another example, more viable in usage to the architectural profession, would be an engineer remotely showing a construction worker how a specific wall specification is meant to be built. Examples of this software include Apple's ARkit and Google's ARcore which scan the physical world, project a digital layer atop this world, and can even cast digital shadows on physical items.[1] AR provides no interaction with the augmented digital world, whereas Mixed Reality (MR) allows such interactions.

Mixed Reality (MR) takes the real world and integrates computer-generated content to interact with the view of the real world. It carries the ability to take fully-generated digital environments and connect them to real-world objects, making it the only technology able to combine the analog and digital realities.[2] Hence, all three of these technologies (VR, AR and MR) refer to highly immersive and virtual experiences that carry value in a discussion of virtual spaces. Toward the architectural context, this essential value of being within a virtual version of a building project at true-scale is that it allows for the intended user to appreciate, interact, or critique a project in the most natural way, whilst it is still being envisioned or constructed into the real world. To understand the difference between these technologies, we can also look to Paul Milgram's "reality-virtuality continuum," a virtuality scale used to measure a technology's amount of virtualness or inversely, realness.

Amongst these different realities, VR appears to be the best way to access the sensation of immersion, or the sense of "being," in a fully virtual environment. In the architectural realm, the majority of projects focus on conceiving and imagining new environments, thus requiring full virtual immersion when these environments are showcased to clients or peers. While VR can then be seen to apply to the majority of the architectural process to imagine designs, AR might provide a more specific ability to solve visualizations for renovations and projects that have features existing in the real world. Projects that require visualizations of additions and changes to existing scenarios might be better focused on through AR. As AR cannot achieve the level of virtual immersion VR does, it is more applicable to projects that require a layer of virtual immersion affixed to spaces existing in real life.

---

[1] Paul Mealy, Virtual & augmented reality for dummies, 1st edition (Indianapolis IN: John Wiley and Sons, 2018).

[2] Ibid.

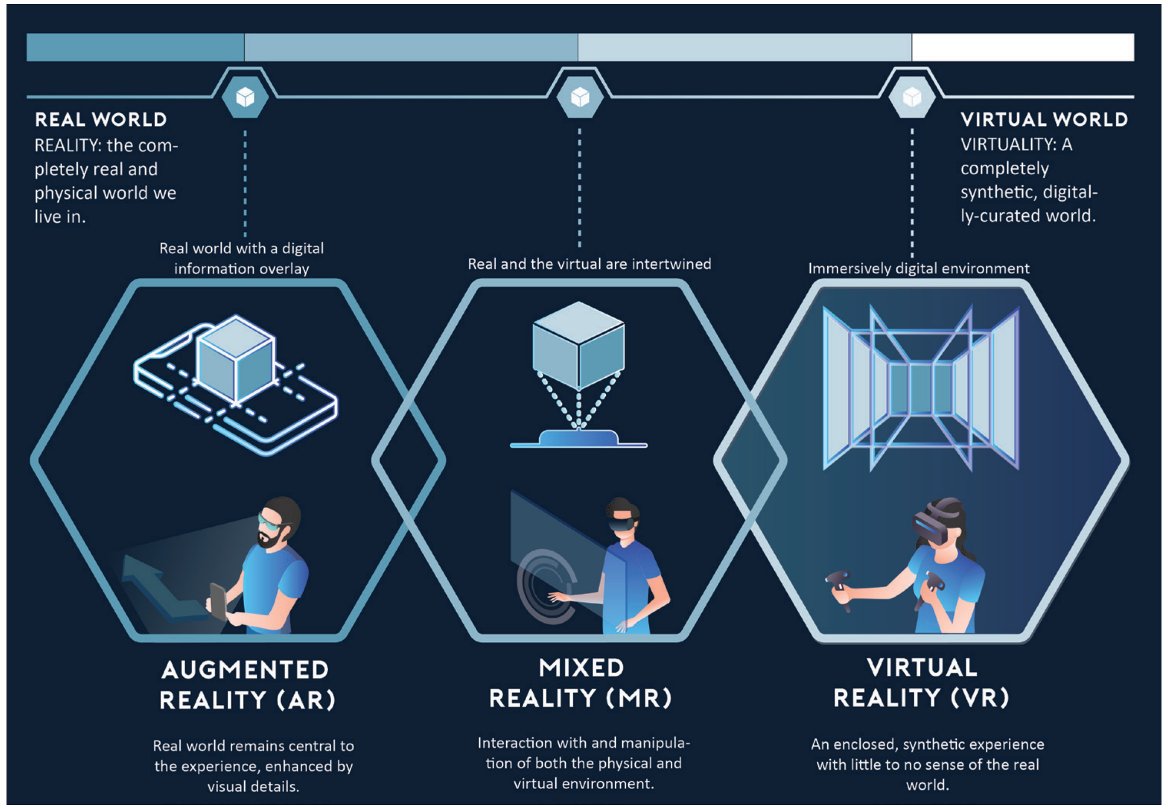


figure 1.09 Differences between AR, MR and VR (Immersion Levels)

figure 1.10 Full Immersion in Virtual Reality

figure 1.11 VR, A Fully Enclosed Experience

As VR closes a user off from the physical world by commandeering sight, it can be used to conceive any space virtually, regardless of whether or not the architectural space exists in the present. Hence, this makes it more diversely applicable to the entire range of work in the architectural realm. Lastly, although MR can be expected to provide both VR and AR (and switch between the two), the technology has yet to reach evolutionary maturity, and its evaluation within the architectural sphere can only be made once it passes through the Hype Cycle or is explored in the consumer market in the same manner experienced by VR.[1] Although market success is not imperative for MR to create an impact within the design practice, it is more importantly an indicator of the consumer or an architectural practitioner's ability to use the innovative technology. The consumer side, after all, trails far behind the enterprise world of technology.[2] Only a product that is well-received in the market is representative of technological maturity and usability within a wide demographic (including architects and designers), asserting that MR is currently too young to be evaluated within the architectural sphere.

---

[1] Gartner Inc., "3 Reasons Why VR and AR Are Slow to Take Off: Here's how technology product managers should plan for virtual reality and augmented reality adoption in the next few years." accessed October 23, 2018, https://www.gartner.com/smarterwithgartner/3-reasons-why-vr-and-ar-are-slow-to-take-off/.

[2] "Virtual Reality is reaching a mature state according to Gartner," The Ghost Howls: A Blog about Virtual Reality, Startup and Stuff, accessed October 23, 2018, https://skarredghost.com/2018/08/27/virtual-reality-is-reaching-a-mature-state-according-to-gartner/.

On Representation

Jaron Lanier, scientist and founder of VPL Research (one of the premier companies to buy and sell Virtual Reality products) convincingly asserts that VR represents the beginning of an "enormous paradigm shift" in the way humans relate and communicate.[1] Architects have long utilized drawing, model-making and visualization tools to communicate designs to both clientele and other designers. With the claims of VR being a vastly immersive and integral change in the way humans communicate ideas, the effect of VR on architectural communication is a worthwhile investigation. Representation is integral to the successful communication of ideas. Architects have determined that buildings communicate a message greater than simply the function enclosed within them. Thus, representation is a necessity in the design profession to be able to showcase projects and engage design thinking. Over several decades, technology has provided architects with a multitude of tools, allowing for computer-aided drafting, photo-realistic rendering and now, virtual reality. Achieving these technological evolutions and still using conventional representation techniques in practice such as hand-drafting, sketching and model building demonstrates that architects use a blend between old and new. Designers combine age-old techniques with the novel technology of modern day to produce designs at different scales and engage different ways of thinking through a project. Consequently, this evolution of digital tools adds a complexity both in the story-building and iterative design stages for the visual manifestation of ideas. For example, the introduction of advanced computational tools has permitted architects and designers to generate complex forms virtually, toggle between scales and views and work at a greater speed to tangibly represent ideas on paper with unprecedented ease.[2] However, these computer-aided design tools currently lack the contribution of an even deeper understanding of a user's experience of space that is achieved beyond the formal, functional and structural aspects of architecture.[3] The core of the architectural practice is centered around iteration and visualization of beautifully-designed spaces, highlighting the driving force of representation through communication within the discipline.[4]

---

[1] Jaron Lanier, Dawn of the new everything: Encounters with reality and virtual reality, First edition (New York: Henry Holt and Company, 2017).

[2] Romullo Baratto, "Trends in Architectural Representation: Understanding the Techniques," ArchDaily, accessed October 23, 2018, https://www.archdaily.com/867060/trends-in-architectural-representation-understanding-the-techniques.

[3] Ibid.

[4] Lorraine Farrelly, Basics Architecture 01: Representational Techniques, Basics architecture 1 (Lausanne: AVA Academia, 2008).

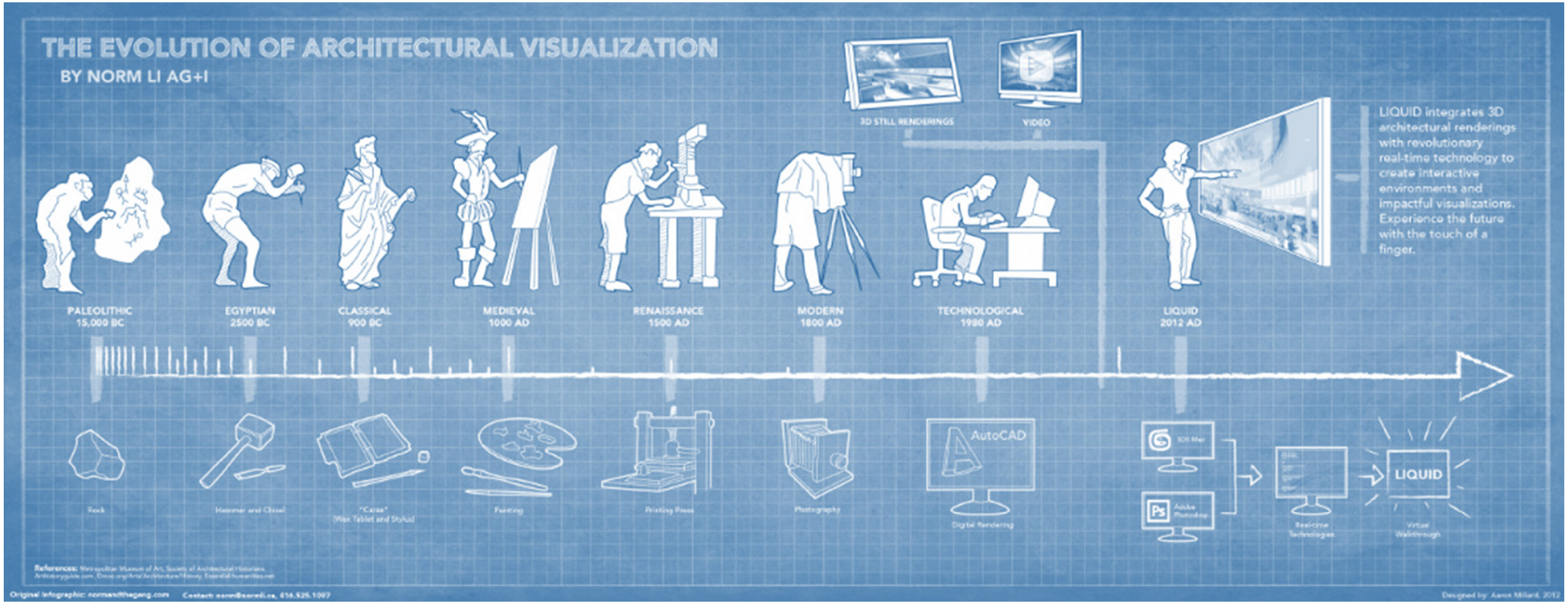


figure 1.12 The Evolution of Architectural Visualization

According to Norm Li, “the technological reality emulation definition of Virtual Reality, is the natural evolution of the visualization realm field from tools we’ve used in the past”. From hammer chisels, to painting presses, to the computer-aided design tools today, architects are believed to be in the natural process of using VR as the next modem of communication and representation of designs, according to visualization specialists.

Chapter 2: Foundation of Investigation

Design Process Integration

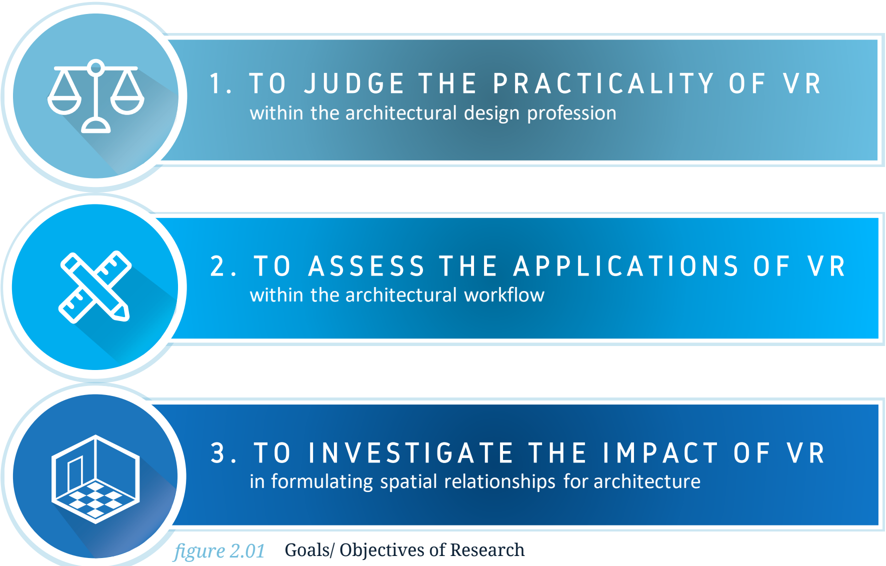


figure 2.01 Goals/ Objectives of Research

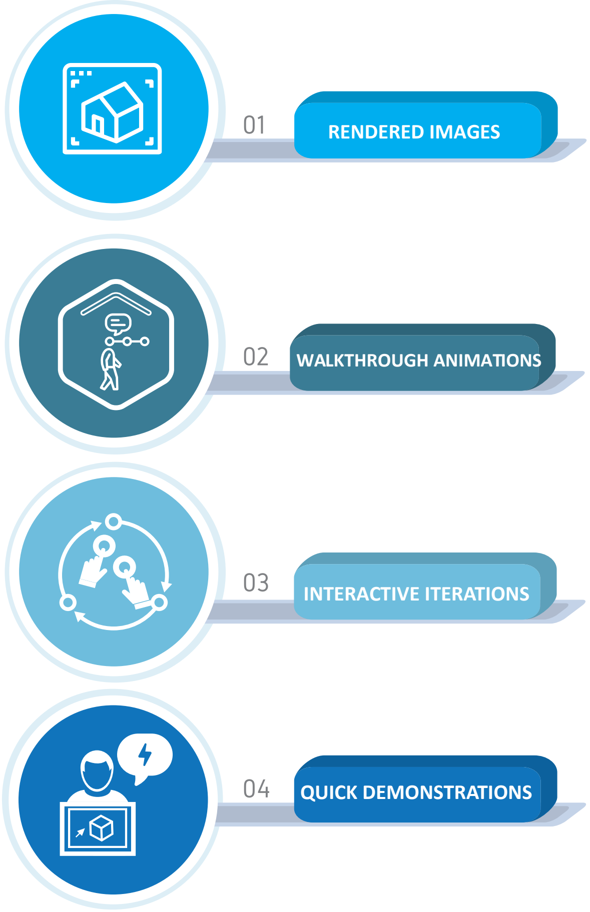


figure 2.02 4 Types of Visualzation Experimentation Explored

VR has found its place in numerous professions including, but not limited to, science,[1] sports,[2] and filmmaking.[3] This leaves a great deal yet to be understood about VR's place within the architectural field. Specifically, we are presented with the need to determine the best way to integrate VR into the architectural design process. It has been stated by Gartner Inc. that within the next five years, VR will reach "technological maturity".[4] Within this timeframe, it is imperative for the design field to discover new ways to utilize the capabilities of VR and move the architectural practice forward. VR developers constantly emphasize how real and visceral a VR experience feels.[5] With the praise VR technology receives, a discussion of the medium's shortcomings within the architectural practice is often left undocumented. There is a need to verify the viability of VR in the architectural context. To address this, this thesis aims first to judge the practicality of Virtual Reality within the architectural design field, before determining an assessment of how VR can best be applied within the architectural workflow. The goal of this thesis is to determine how VR might be able to change the way architects learn, play, and communicate designs. VR also carries the potential to change the way that architects think about designs. In this regard, this thesis primarily focuses on the visualization aspect of the design process. Design visualization is how the architectural practice has communicated any work that is in progress to teams, clients and other architects. It is integral to the process architects use to think through ideas, resolve problems and make decisions within a project.

---

[1] Eyal Ophir, Clifford Nass, and Anthony D. Wagner, "Cognitive control in media multitaskers," Proceedings of the National Academy of Sciences of the United States of America 106, no. 37 (2009), https://doi.org/10.1073/pnas.0903620106, http://web.uvic.ca/~dbub/Cognition_Action/Essaytopicscontrol_files/Mediamultitaskers.pdf.

[2] Jane Zorowitz, "It Just Got Real: Coaches like Bret Bielema and Bill Belichick are getting on the virtual-reality wave," accessed March 1, 2019, https://sportsworld.nbcsports.com/virtual-reality-sports-arkansas-kentucky/.

[3] Celine Tricart, Virtual reality filmmaking: Techniques & best practices for VR filmmakers / Celine Tricart, 1st (New York: Focal Press, 2017).

[4] Gartner Inc., "Gartner Hype Cycle: Interpreting Technology Hype," accessed October 18, 2018, https://www.gartner.com/en/research/methodologies/gartner-hype-cycle.

[5] Michael Abrash, "Welcome to the Virtual Age," Oculus, accessed October 24, 2018, https://www.oculus.com/blog/welcome-to-the-virtual-age/.

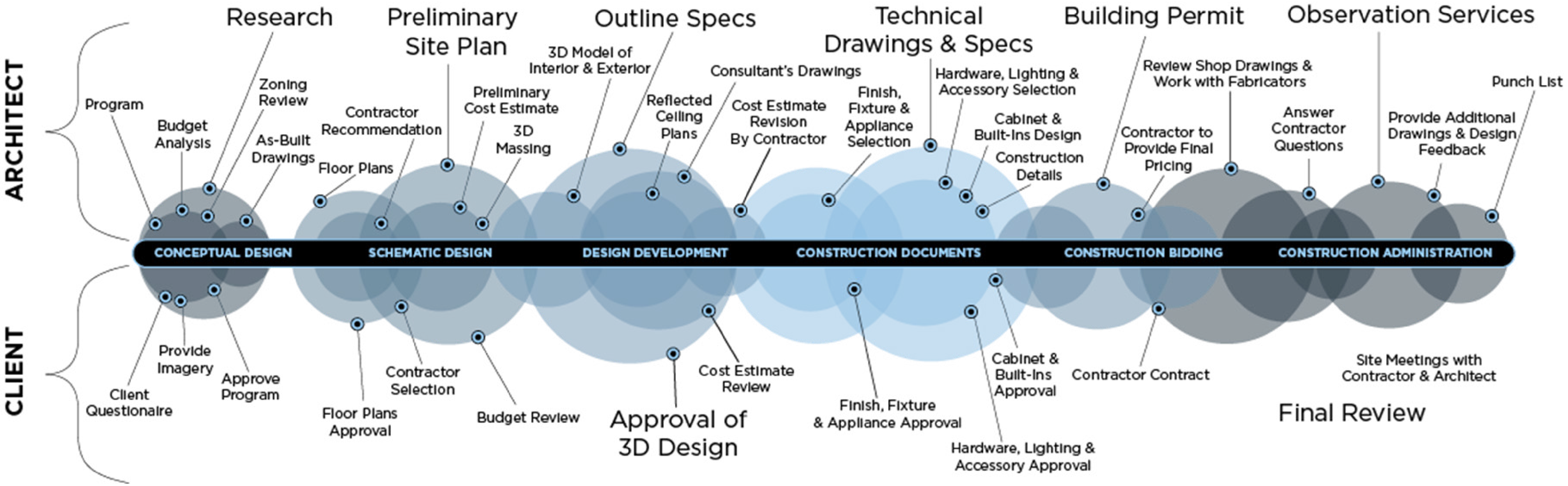


figure 2.03 Design Stages & The Architectural Process

Identifying Underlying Questions for Creating Content in VR

1. What problem does a VR project uniquely solve?

The ability to immerse oneself within the environment of an idealized design.

2. Who is the target market?

Clientele, Stakeholders or Design Peers within the architectural design practice.

3. What is the vision for the end-user experience?

A seamless experience of a simulated environment created from an existing model. This simulation should be able to display interactive options for the iterative process of design or provide a hyper-realistic finished product that is usable as a marketing medium.

Strengths and Weaknesses in Context

The practice has long formulated the workflow of using drawings and models in initial stages of conceptualization before evolving to model-making and the use of digital design tools. Renderings, animations and walkthroughs are most commonly used in practice to formalize ideas and communicate projects within the conceptual design, schematic design and design development stages of a project.[1] Therefore, this thesis seeks to examine the use of VR with rendering images, animations and the creation of walkthroughs to understand the full spectrum of visualization features' workflow with VR.

Before any assessments can be made of VR within the design profession, the strengths and weaknesses of the technology must be determined. This can be achieved by understanding the value it provides to the architectural field. Undeniably, the fact that VR is able to showcase environments well before they are physically built is extremely beneficial to communicating design specifics in a life-like scale. VR technology is successful because it unlocks the factor of embodiment; "the state of existing, occurring or being present in a place or thing."[2] Embodiment allows a user to directly activate two experiential factors. The first of the two is presence, which is the feeling of encompassing an environment. Activating presence is the brain's way of telling the body that an experience is real and that it is different from simply looking at an image or at a screen.[3] Scientifically speaking, it involves the activation of the brain's motor cortex and the body's sensory system in a manner similar to their activation during a real life experience.[4] The second factor brought forth by embodiment is the experience of the real-life scale of objects within the environment. Though the sensation of embodiment brings forth both presence and scale, these factors are extremely fragile and may not be guaranteed for every experience. VR can thus enhance a project when done correctly, but also has the capability to obfuscate a project when done incorrectly. Therefore, VR warrants exploration simply due to the fact that constructing a project is egregiously expensive and requires years to formalize. As virtual reality is successful in the creation of immersion and transporting a user to a simulated environment, it presents value to clients and project stakeholders.

---

[1] Romullo Baratto, "Trends in Architectural Representation: Understanding the Techniques," ArchDaily, accessed October 23, 2018, https://www.archdaily.com/867060/trends-in-architectural-representation-understanding-the-techniques.

[2] Konstantina Kiteni and Raphaela Groten, "The Sense of Embodiment in Virtual Reality," Event Lab, accessed March 1, 2019, http://diposit.ub.edu/dspace/bitstream/2445/53294/1/634024.pdf.

[3] Andy Wilson et al., eds., Proceedings of the 2nd ACM symposium on Spatial user interaction - SUI '14 (New York, New York, USA: ACM Press, 2014).

[4] Thong Nguyen, Can Virtual Reality Change Your Mind? TEDxMinneapolis, with the assistance of TEDx (2018), Video, accessed March 1, 2019, https://www.youtube.com/watch?v=eFHj8OVC1_s.

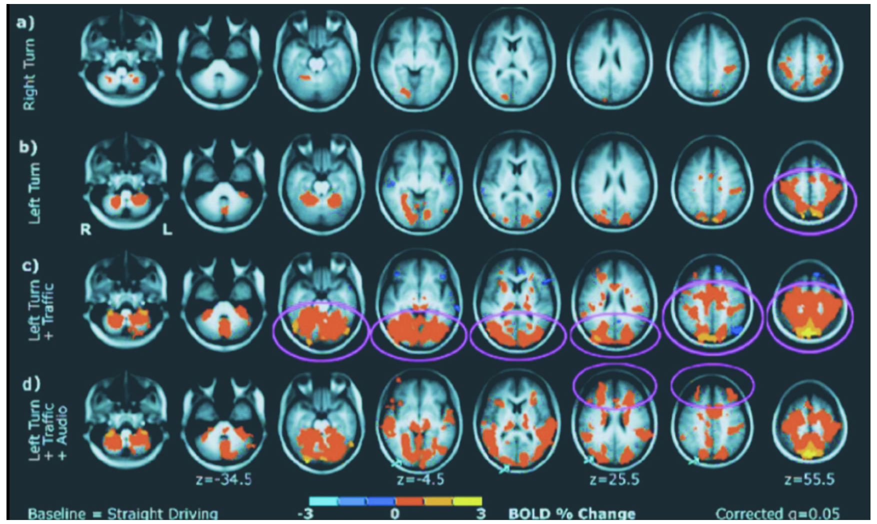


figure 2.04 Brain activations from the bottom to the top of the brain (left to right figures) of participants when performing various simulated driving conditions in Virtial Reality.

However, a substantial weakness is that a VR experience might not be entirely frictionless. Though VR carries enormous potential to experience an unbuilt design, the quality of an experience can quickly decline due to several reasons. Namely, problems can arise surrounding loss of presence, unrealistic features, unintuitive features and general discomfort due to motion sickness. As VR is being used as a medium to evaluate designs, it could potentially represent projects in an unflattering light. If the experience of a represented design is soured, it might be detrimental to the very design being showcased. Additionally, hyper-realism and high-end experiences require an immense attention to detail along with focus on precision. The substantial time, effort, and expenses required to build quality VR experiences may be limiting. A consideration of this weakness may partly explain why VR has not achieved widespread adoption of higher-end experiences to date. With the release of Garner's research on the 2018 Hype Cycle, it was revealed through a conducted critique that the greatest barrier to adoption is "lack of good user experience design".[1] This likely correlates with VR's applicability in the architectural field. VR has been created by developers and primarily focused towards facilitating a workflow suitable for those with a programming background. A programmer might approach content creation in a different manner and workflow than an architect. The strengths and weaknesses of VR in an architectural workflow requires further exploration.

---

[1] Gartner Inc., "3 Reasons Why VR and AR Are Slow to Take Off: Here's how technology product managers should plan for virtual reality and augmented reality adoption in the next few years." accessed October 23, 2018, https://www.gartner.com/smarterwithgartner/3-reasons-why-vr-and-ar-are-slow-to-take-off/.

Relevance of VR in the Architectural Field: Communicating Intent

VR head-mounted displays (or HMDs) can change the process by which we design and communicate buildings during the stages they are envisioned simply because of their ability to engage visual immersion. Michael Abrash, Chief Scientist at Oculus asserts that, “the human perceptual system has evolved to capture massive amounts of data from our environment.”[1] Yet, until today, each form of communication has used “only a small fraction of that capability, equivalent to sipping information from a straw.”[2] It follows that all currently used mediums of representation provide limiting descriptions of a built environment. Beit drawings, literature, or physical models, the full experience was ultimately always reconstructed by the human mind. With the immersive power of VR, this changes, and the sense of spatial perception is engaged. In cases where the subconscious mind is engaged in virtual reality, spatial cognition is also able to be triggered.[3] This ability to trick the human sense of sight is a powerful advantage in the architectural field. The lenses of a headset are responsible for mapping the display to the wide field of view a user sees through the headset. Since the wearer is instantly immersed in a true three-dimensional environment, they are given a greater sense of scale, depth and spatial awareness that is unmatched by traditional renders, video animations or physical-scale models.[4]

This visceral sensation of actually being inside an architectural space also makes VR an incredibly powerful tool for communicating design intent. Clients often do not have the ability to perceive simple spatial relationships and scale just by looking at a 2D plan or 3D model, like architects train themselves to. VR can thus induce an intuitive and coherent response realistic to that evoked by physical architecture.[5] This added physical dimension in 4D might also be used to notice aspects of a project inaccessible by other mediums of representation. Each medium used in architectural representation allows for designers to access different views of the proportions of a project, cultivating unique layers of information. For example, orthogonal drawings allow architects to witness layouts represented through drawing various 2D cuts and perspectives.

---

[1] Michael Abrash, “Welcome to the Virtual Age,” Oculus, accessed October 24, 2018, https://www.oculus.com/blog/welcome-to-the-virtual-age/.

[2] Ibid.

[3] Michael J. Proulx et al., “Where am I? Who am I? The Relation Between Spatial Cognition, Social Cognition and Individual Differences in the Built Environment,” Frontiers in psychology 7 (2016), https://doi.org/10.3389/fpsyg.2016.00064.

[4] AEC Magazine, “Virtual Reality for architecture: a beginner’s guide,” Building Information Modelling (BIM) technology for Architecture, Engineering and Construction, accessed October 24, 2018, https://www.aecmag.com/59-features/1166-virtual-reality-for-architecture-a-beginner-s-guide.

[5] Ibid.

Viewing scaled models (both physically and digitally) enables architects to understand spatial relationships by orbiting around a project. The access of the fourth layer of dimensionality through VR interaction therefore unlocks the unique ability of first-person interaction with space. Architectural designer and visualization specialist Norm Li asserts that architects and designers “may envision something in the mind, but the physical resolution of the thought- the actual building- can wind up being completely different”.[1] With VR, the perception of the space is constructed as the architect is within the space, allowing for “no excuse for not understanding the eventual result”. The capabilities HMDs hold to affect our sense of perception of a project demonstrates the importance of the apparatus. VR is relevant because perception plays such an integral role in the way architects, clients, investors and stakeholders validate assumptions carried about a design.

Importance of VR in the Architectural Field: Presence & Scale

The two key things that separate traditional visualizations from Virtual Reality experiences are presence and scale, as mentioned previously in the Strengths and Weaknesses subsection. These two factors will be discussed at a greater depth with relation to the architectural practice in this section. Presence is, exactly as the term implies, the effect of feeling as though one is truly inside a virtual space. It is a feeling that transports a user from their physical surroundings into a virtual world. Experiencing presence in any VR scene aids in providing the basis for a positive experience within a simulation. This sense of presence is extremely fragile. When this aspect of presence is lost or absent within the rendered environment, the experience tends to be negative, unfulfilling and simply unviable for the architectural practice.[2] Presence is thus highly relevant to architectural VR and provides the basis of experiencing space. The second factor is scale and it engages the practice with even more applicability. The experience of a built environment at true scale allows any client or designer to understand the true implications of their creation in reality. Whether it is a deep cantilever over 2 storeys or a mass scale 100-story tower, experiencing a design is null without the engagement of its actual size. Experiencing any design at true scale represents a major step forward in architectural communications. Designers can thus recognize the merit in showcasing designs through VR, as they may thereby be able to accurately showcase dimensions, represent collisions in a mechanical system or even depict the ergonomics of a design. However, given the complexity of VR, the two factors of presence and scale are not guaranteed in every VR experience.

---

[1] Michael Totzke, “The Real and Virtual Norm Li,” accessed November 9, 2018, https://www.canadianinteriors.com/features/real-virtual-norm-li/.

[2] Paul Mealy, Virtual & augmented reality for dummies, 1st edition (Indianapolis IN: John Wiley and Sons, 2018).

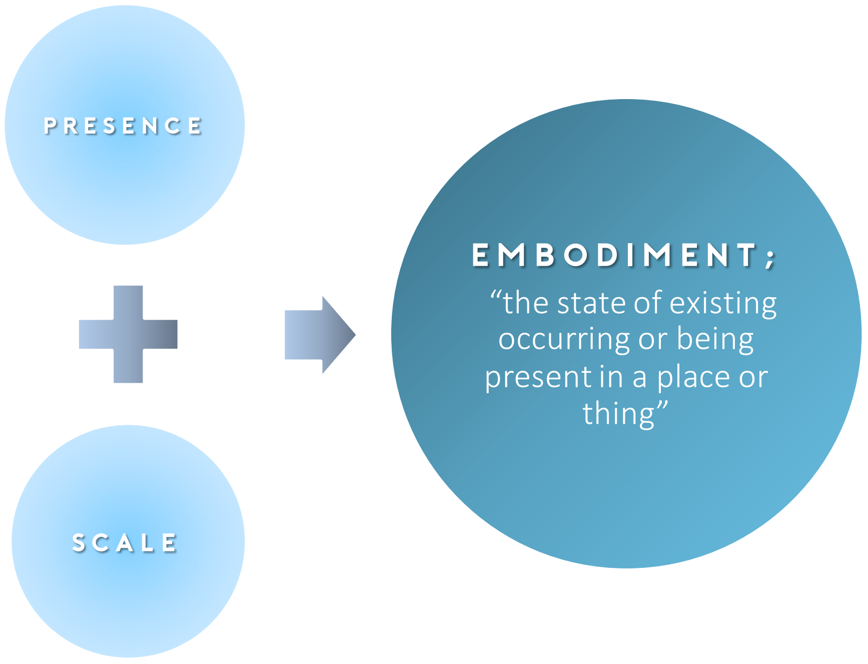


figure 2.05 Underlying Factors Behind Embodiment

A VR simulation can only be as successful as its implementation, which is a challenging task.[1] The goal of VR is that the user feels subconsciously present in a virtual world. The human mind has evolved over millions of years to perceive the natural reality. Being able to present a user a virtual environment that their brain can accept as a subconscious reality during an experience remains the greatest challenge of VR.[2] When executed correctly, VR can connect architects to their full power of perceptual capabilities, creating a new means of interacting with digital information.

---

[1] William Sherman and Alan Craig, Working with virtual reality (Morgan Kuafmann, 1998).

[2] William R. Sherman and Alan B. Craig, Understanding virtual reality: Interface, application, and design, Second edtion, The Morgan Kaufmann series in computer graphics (Cambridge, USA: Morgan Kaufmann Publishers, 2019).

Chapter 3: Hypothesis

Experimental Aim & Scope

The aim of this thesis is to develop an understanding of creating VR experiences from the reference point of an inexperienced programmer in the architectural practice. A designer might question the reason for this exploration in an era where architects and designers can invest in firms that specialize in visualization to create VR experiences for them. Architects are able to pay for the output they require but that negates learning through practice and can often hinder the ability to customize an experience. Many practitioners may choose that course of action and often do, however it is important to test design development in VR because of the value it can provide. Architects exploring the creation of virtual reality experiences can benefit from learning complex visualization concepts to achieve autonomy over the design and visualization process. The ability to put a viewer into a design and visualize data in real-time for a fully customized design project is a boon for the architectural practice. Whether VR is explored through rendered imagery, interactivity or animated walkthroughs, the architectural field now possesses an added layer of perception for a conceptualized design, expanding the possibilities of communication. Evaluating VR in a design context can allow architects to understand the true potential and pitfalls of the medium. This thesis tests the process of designing in VR, where documentation is used to evaluate the hardware and software options used in stages. As such, this experimentation aims to understand the design of virtual reality spaces by assessing the creation of 3 outputs: rendering, interaction, and walkthrough animations. It will evaluate the integration of VR into the architectural workflow by using models of architectural design projects (created with common 3D architectural design software) that are imported into game engines (the primary software resource for designers to create VR experiences). Although push-button VR software and architectural plugins exist, they are commonly used to conceptualize projects in a standardized format and will be explored within Chapter 8 of this thesis. These programs and plugins represent a “microwave” method for VR creation and are restricted to specific output formats, graphics levels, and functionality. As this thesis aims to design for virtual reality from scratch to achieve customizability in the aforementioned 3 outputs, visualizations will be created with the use of game engines. The technical goals within this process will involve differentiating between building VR scenes in multiple game engines and hardware options. This will accommodate a greater breadth of knowledge towards hardware and software workflows prevalent within this process. With the creation of each experiment, an evaluation will be made to assess the ease of VR technology integration into the architectural design process.

State of the Art: Experimentations with Spatial Representation

The onset of Virtual Reality usage in studies has spurred new opportunities in the development of state-of-the-art neuropsychological assessments.[1] VR has enabled neuropsychologists to assess and measure more accurately, factors such as users' cognitive, sensory and motor abilities alongside behavioural or self-regulatory functions, all while users experience a virtual environment.[2] Successful neuropsychological experimentation with virtual reality allows for an understanding of how positive VR experiences can be created. Though this experimentation is conducted in a laboratory setting, it features positive VR experiences that were able to activate a sense of embodiment and use it to receive a response from users. Due to this, these investigations can aid in correlating VR to spatial understanding, by outlining a set of experiential design principles. This research can also allow architects to better understand successful and necessitous factors to spaces represented in virtual reality.

1. WAYFINDING: NAVIGATION REQUIRES SPATIAL CUES

When landing in a new space, people analyze their environment to search for signals that clue them in on how to get where they want to be. Likewise, as a virtual reality experience is a relatively unknown and unfamiliar space, it requires spatial cues. Cell neurobiology research analysts Rizzo, Schultheis, Kerns and Mateer conducted research to analyze VR in cases that support disabled individuals in wayfinding.[3] In this study, VR technology was used to aid developmentally-disabled teenagers navigating a supermarket and to assist children navigating schools in wheelchairs.[4] The success rate of participants in wayfinding these spaces with the help of VR indicates that the medium engages spatial navigation in a realistic fashion. Through multiple rounds of running through the simulations, users were able to learn how to navigate the public spaces in question.[5] As users were placed into the VR space, they were given navigational cues in the form of markers to guide them through their environment.[6] This is useful for architects to understand, as the development of VR experiences requires substantial attention to the ways a user navigates the environment.

---

[1] Steve Aukstakalnis, Practical augmented reality: A guide to the technologies, applications, and human factors for AR and VR (Upper Saddle River, N.J.: Pearson Education, Inc, 2017).

[2] Anna B. Boyum, "Virtual Reality in Neurology," ModernMedicine Network, accessed December 6, 2017, https://www.neurologytimes.com/stroke/virtual-reality-neurology.

[3] Albert A. Rizzo et al., "Analysis of assets for virtual reality applications in neuropsychology," Neuropsychological Rehabilitation 14, 1-2 (2004), https://doi.org/10.1080/09602010343000183.

[4] 2016 IEEE Symposium on 3D User Interfaces (3DUI) (IEEE, 2016 - 2016).

[5] Albert A. Rizzo et al., "Analysis of assets for virtual reality applications in neuropsychology," Neuropsychological Rehabilitation 14, 1-2 (2004), https://doi.org/10.1080/09602010343000183.
6 2016 IEEE Symposium on 3D User Interfaces (3DUI) (IEEE, 2016 - 2016).

As VR headsets obstruct human vision in place of a simulated environment, the boundaries of an architectural space are replaced, requiring the need for navigational cues in the simulation. In reality, the design of any large-scale spaces such as shopping malls, airports or residences is facilitated with signs and navigational cues. In the same way, these cues have to be presented in virtual reality simulations to allow a user to successfully navigate the space.

figure 3.01 Common Wayfinding Signals in High Traffic Urban Environments

figure 3.02 Adding Navigation and Points of Interest to a VR Simulation

2. PERCEPTION: REFERENCE POINTS, SCALE AND SIZE MATTER

The human perception of oneself is heavily tied into the perception of physical space, which is the reason behind people feeling giddy in certain perspectives or claustrophobic in small rooms. Virtual reality experiences are easily uncomfortable for users and thus VR's long journey from conception has involved a constant battle with simulator sickness.[1] Although a self-assessment study has demonstrated that simulator sickness does not carry any long-term side effects,[2] the experience of a VR environment can be extremely unsatisfying if simulator sickness is present. To combat this, researchers at Purdue University implemented a virtual nose into VR simulations and found that it improved the effects of the visual display, reducing simulator sickness by 13.5 percent.[3] Researchers often point to sensory conflict as a primary cause of simulator sickness.[4] However, it was discovered in this experimentation that inaccurate bodily measurements and scale also play a significant role in inducing sickness. Fixed reference points can thus tend to reduce these feelings of discomfort. During the Purdue University research study, 41 participants used a multitude of diverse VR applications (ranging from the user walking around a Tuscan villa to a user riding a roller coaster), half of the participants took part playing games with the virtual nose and half played without a virtual nose. Participants with the virtual nose were able to play the game for 94.2 seconds longer than those without the virtual nose.[5]

This study indicates the importance of physical reference points, accurate bodily measurements and scaling in virtual simulations. The merit behind having the ability to experience virtual environments at 1:1 scale can be easily understood. However, it should not be forgotten that humans have proprioceptive senses of their human scale in reference to their environment in reality.[6] People already have an understanding of their bodily measurements prior to being inserted into a virtual environment, indicating that if they are presented with jarring changes from those reference points, they may feel sick.

---

[1] Sue V. G. Cobb et al., "Virtual Reality-Induced Symptoms and Effects (VRISE)," Presence: Teleoperators and Virtual Environments 8, no. 2 (1999), https://doi.org/10.1162/105474699566152.

[2] Frank Steinicke and Gerd Bruder, "A self-experimentation report about long-term use of fully-immersive technology," in Proceedings of the 2nd ACM symposium on Spatial user interaction - SUI '14, ed. Andy Wilson et al. (New York, New York, USA: ACM Press, 2014).

[3] Ajoy S. Fernandes and Steven K. Feiner, "Combating VR sickness through subtle dynamic field-of-view modification," in 2016 IEEE Symposium on 3D User Interfaces (3DUI) (IEEE, 2016 - 2016).

[4] Joseph J. LaViola, "A discussion of cybersickness in virtual environments," ACM SIGCHI Bulletin 32, no. 1 (2000), https://doi.org/10.1145/333329.333344.

[5] Ajoy S. Fernandes and Steven K. Feiner, "Combating VR sickness through subtle dynamic field-of-view modification," in 2016 IEEE Symposium on 3D User Interfaces (3DUI) (IEEE, 2016 - 2016).

[6] Gary E. Riccio, Eric J. Martin, and Thomas A. Stoffregen, "The role of balance dynamics in the active perception of orientation," Journal of Experimental Psychology: Human Perception and Performance 18, no. 3 (1992), https://doi.org/10.1037/0096-1523.18.3.624.

This highlights the innate necessity of including reference points, ensuring an avatar is rendered to scale and that the size of the environment is not constraining. Building an environment to scale, size and with reference points allows designs to be better conveyed to clients in practice. After all, the human perception of oneself is central to the perception of physical space.

figure 3.03 An Alterning Shift in Our Real-life Perspective

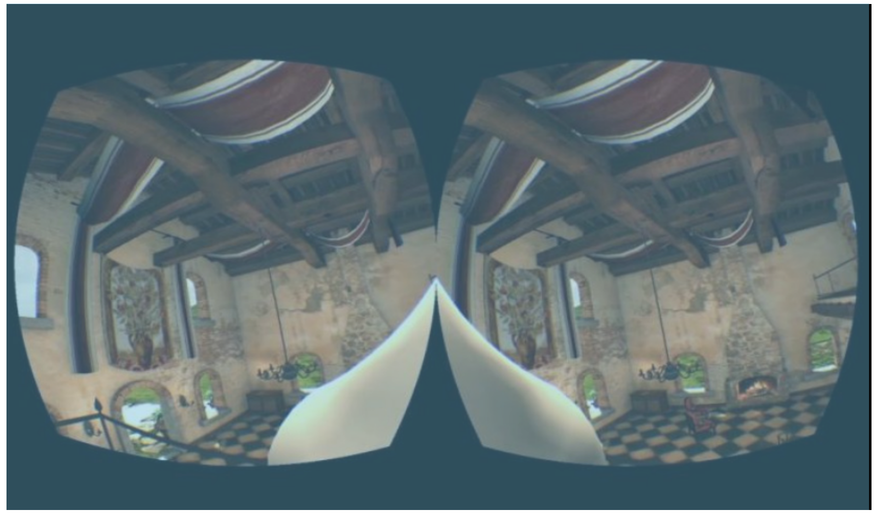


figure 3.04 The Implementation of a Virtual Nose to Connect a Viewer to Physical Refernce Points

3. PROMPTING: NARRATION PROVIDES VALUE

Designers educate clients on spaces and gauge their focus towards different areas of a project. In reality, a demonstration of an architectural project rarely, if ever, occurs alone. In this way, a focus on design principles is gauged through prompting. Translating this knowledge into a virtual experience of space, a user might not know where to focus in the new unknown environment. An Oxford University experiment conducted by Professor Daniel Freeman studying alleviating acrophobia found success in the implementation of a virtual coach. Participants of this study in acrophobia engaged in different height-related virtual reality scenarios.[1] The experiment places users within a 10-storey virtual office building (by administering an HTC Vive headset) where participants, with the guidance of a virtual coach tackle challenges of increasing difficulty. Participants of the study were asked to walk along a shaky walkway, rescue a cat from a tree within the building's atrium, conduct tasks while on the edge of a balcony, and finally, ride a virtual whale around the atrium space. Participants were asked to walk around and activate the hand-controllers during the experiment as per the dictation of their virtual coach.[2] The integral component of this experiment was the virtual coach, who was programmed with voice and animations to act as an avatar simulated in the environment. Their guidance and encouragement to engage in the activities prompted participants' success through the experiment's challenges. This proves that any presence in a virtual scene, whether in the form of a digital avatar or narrational guidance, can be beneficial and reassuring for a user. This experiment demonstrates that narration or prompting through audio might often be recognized as a second-class feature to VR, when indeed it can make or break a virtual experience. Compelling virtual environments often stimulate all the senses, but VR currently best stimulates a virtual and auditory experience.[3] Background audio, voiceovers and sound effects are the three ways sound transforms into spatial audio in VR. Architects need to design in VR with the consideration of sound as a real-world experience. Sound is used effectively when prompting a user to conduct tasks in a simulation. The reassurance that comes from a digital avatar can aid a user within a simulation to focus on their goal. In reality, when designs are physically presented to an unfamiliar eye, the same auditory guidance is provided by a tour guide. This implementation can thus be valuable in the design of VR simulations.

---

[1] Daniel Freeman et al., "Automated psychological therapy using immersive virtual reality for treatment of fear of heights: a single-blind, parallel-group, randomised controlled trial," The Lancet Psychiatry 5, no. 8 (2018), https://doi.org/10.1016/S2215-0366(18)30226-8.

[2] Ibid.

[3] William Sherman and Alan Craig, Working with virtual reality (Morgan Kuafmann, 1998).

figure 3.05 Tour Guides Explaining Space to Viewers in Reality

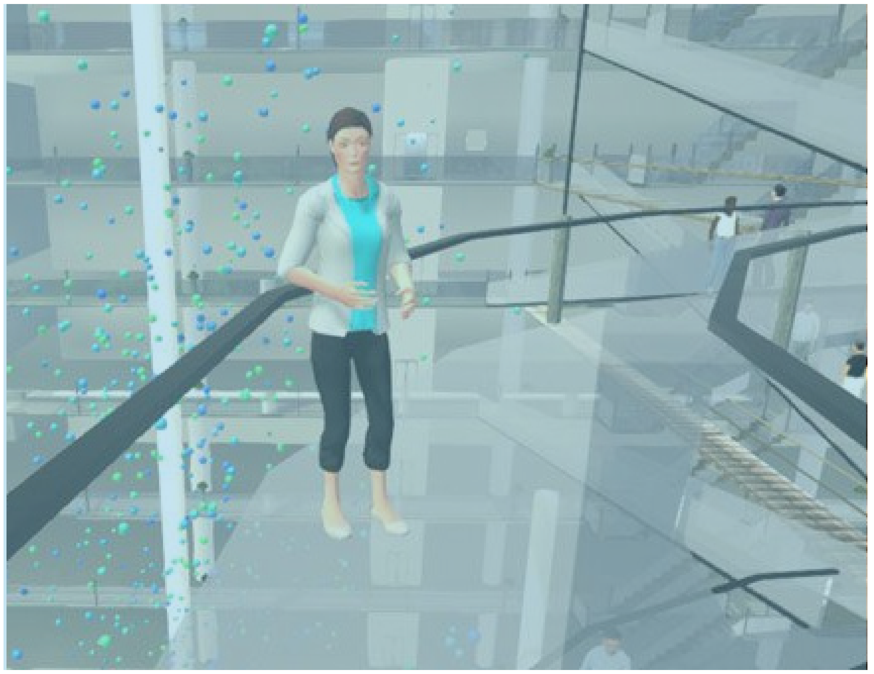


figure 3.06 Utilizing a Virtual Guide to Assist and Encourage Users

4. UNFAMILIAR ACTION: INTUITIVE INTERACTION IS KEY

A well designed architectural space can induce an instinctive physical response. For example, the experience of a design installation can encourage walking through, jumping on or sitting a specific area. Although virtual reality is an experience of a virtual space, it can still evoke physical responses. The Cochrane Review outlined some of the first breakthrough cases comparing the effects of VR against alternative methods of rehabilitation on participating stroke victims.[1] The study focused on the opportunity to recover victims by testing arm function, walking speed and the ability to manage daily functions following the experience of a traumatic stroke. The review studied 72 cases involving 2,470 people after the experience of a stroke. The study found that users were able to regain arm function by practicing gait and balance VR scenarios.[2] The VR therapy designed in these cases gave victims the opportunity to practice everyday activities that were not and could not be recreated in hospital environments. Although the quality of the evidence gained in this study was deemed as "low to moderate quality," fifty of these studies had positive findings that VR used in addition to rehabilitation, or on its own, resulted in better walking ability, arm functionality and ability to dress oneself and shower.[3] This success of using VR to assist rehabilitation represents one of the most important set of cases showing the ground-breaking opportunities that exist with VR technology's affordance with interaction. This influence can be tied back to that of architects, in possessing the ability to shape interactive spaces for their inhabitants. Designers are urged to facilitate creating intuitive spaces, allowing users to easily and naturally figure out how to engage with their simulated environment. Without intuitive design elements in their environment, users in VR spaces can be lost, distracted or unable to focus on the design of the space in question. Intuitive interaction is key in designing VR spaces because it demonstrates the mode of action to the unfamiliar audience. If a user is meant to walk, sit or jump through a space in reality, it is beneficial to have a simulation of that architectural space designed to encourage that.

Amongst all the research presented, the main premise of representing designs in VR is that the design matters above all.[4] Placing design intent into each feature created in a virtual space is necessary to achieve a successful virtual experience.

---

[1] Kate E. Laver et al., "Virtual reality for stroke rehabilitation," The Cochrane database of systematic reviews, no. 2 (2015), https://doi.org/10.1002/14651858.CD008349.pub3.

[2] Ibid.

[3] Ibid.

[4] Donald A. Norman, The design of everyday things, Revised and expanded edition (New York, New York: Basic Books, 2013).

figure 3.07 Users Intuitively Interacting with an Installation By Stepping on Lit Up Areas

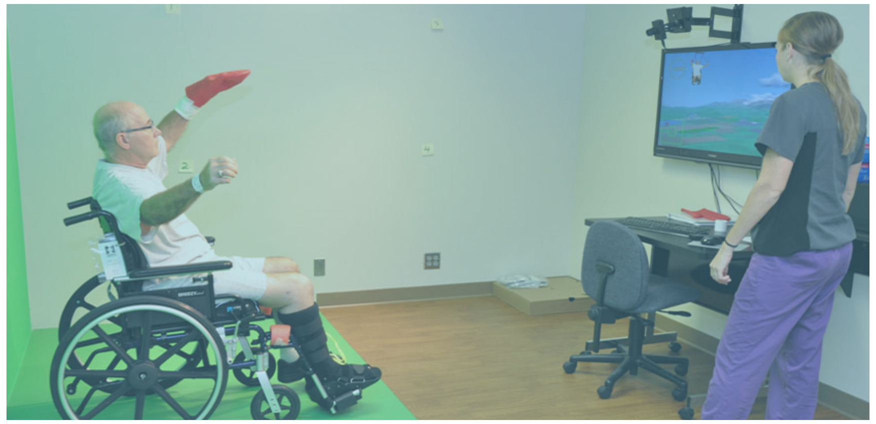


figure 3.08 Using Gesture Control Technology in Virtual Simulations to Rehabilitate Patients

This realization is further emphasized in the work of psychologist Sally Augustin who studies person-centered design.[1] Augustin believes in the formation of sensory stimuli from physical and virtual environments. That is to say that the design of our environments impacts us directly as humans. Architectural scholar Steven Holl, similarly attests to the importance of intuition in the construction and experience of built space.[2] The visualization of lighting effects, textures, and interactions can all evoke visceral sensations. All of these formal and spatial principles come together to convey that everything makes a difference in the perception of a virtual space. Presenting a good architectural design effectively takes a great deal of thought and effort.

---

[1] Sally Augustin, Cindy Coleman and Neil Frankel, Place advantage: Applied psychology for interior architecture (Hoboken, N.J.: Wiley, 2015).

[2] Fred A. Bernstein, "Steven Holl Leaves His Mark: With blockbuster projects nearing completion all across America, architect Steven Holl prepares for the biggest year of his career—and many more milestones to come," accessed December 24, 2018, https://www.architecturaldigest.com/story/steven-holl-leaves-his-mark.

Contextual Viability: Creating Immersive Representation

In an experiment conducted at the University of Waterloo School of Architecture, spatial relationships through designing in virtual reality were tested.[1] The experiment reviewed the differentiation between determining distances from virtual reality simulations as opposed to orthographic architectural drawings. This study was the foundation to understand whether or not VR might be effective in conveying spatial relationships. Participants in the study were asked to approximate distances from an orthographic drawing and move walls in a virtual reality scene to those set distances. Users were able to move the walls further away from or closer to them to affect their perception of interior spaces with regard to shape, details and population density. Although drawings were found to have a greater accuracy in measurement approximations, participants were found to create rooms with dimensions similar to one another. This suggests that VR, as a representational tool, has the potential to “impart a common understanding of space to different people”[2], positively reassuring designers that VR is an effective medium in design communication. Successful design communication and the representation of spaces in virtual reality can be tied deeply to phenomenology, as the results of this spatial perception study can show. Phenomenology is the “manipulation of space, material, lights and shadows to create a memorable encounter through an impact on the human senses”[3]. It directly deals with the human experience of structures by understanding consciousness. Phenomenology can be attributed to activating temporal, spatial awareness and self-consciousness through purpose or intention in action. Understanding phenomenology and being able to actively participate in interacting with a virtual world makes VR enticing in the design field. Architectural scholars, Alberto Pérez-Gómez and Steven Holl directly correlate human perception to the phenomenological experience of architecture.[4] They believe in representational tools having a direct influence on conceptual development of projects, specifically with relation to the generation of forms. From the evidence provided justifying the scientific and philosophical nature of virtual reality, research points positively to the practicality of VR as a medium for immersive representation.

---

[1] Ronald Tang, Step into the Void: A Study of Spatial Perception in Virtual Reality (UWSpace, 2019), http://hdl.handle.net/10012/14468.

[2] Ibid.

[3] Kraus, “Theory of Phenomenology: Analyzing Substance, Application, and Influence,” Arch 630: Theory and Context, accessed December 25, 2018, https://cte.ku.edu/sites/cte.drupal.ku.edu/files/docs/portfolios/kraus/essay2.pdf.

[4] Steven Holl, Juhani Pallasmaa and Alberto Pérez-Gómez, Questions of perception: Phenomenology of architecture (San francisco: William stout, 2008).

Applicability versus Practicality

The research of this thesis began following the praise surrounding virtual reality in an appreciation of the collaborative nature of the medium. VR is constantly acclaimed as interdisciplinary by developers and virtual reality enthusiasts in media. It might be easy to be persuaded that VR is diversely applicable, especially by the successful results of research documented in this chapter. There may in fact be a multitude of ways virtual reality has applications toward the architectural practice and design workflow. However, this doesn’t necessarily guarantee the practicality of the medium. Practicality comes from a judgement of the expense of effort, time and resources that it takes to make Virtual Reality a successful representational medium in conveying the intent of an architectural design.

applicability ≠ *practicality*

Chapter 4: Methodology

Method Behind the Medium

There are a multitude of experiences that can be created in VR and various ways to go about creating them. The advent of major VR software and hardware releases from 2016 challenged the industry's VR standards and spurred a mass production of immersive HMDs. The increase in both hardware and software options for VR has brought forth the accessibility of VR technology, both financially and physically. Although the accessibility of VR technology present today brings enormous merit, a harrowing task of selecting and differentiating between tactics for the creation of VR experiences now ensues. This chapter seeks to address the methodology behind experimentation in the following chapters and to differentiate between hardware and software options used within practice. To address the practicality of VR in architecture, there is a need to systematically compare and contrast methods and choose the most reasonable mediums to facilitate this experimentation. This chapter addresses the design framework for experimentation, VR hardware and software, along with the current limitations of the technology.

Experimental Framework

After the release of Gartner's 2018 Hype Cycle, an article was released claiming "the biggest barrier to wide adoption of immersive technologies" as the "lack of good user experience design".[1] This critique links directly to the general understanding of VR practicality in architecture. For a long time, the user experience of VR has been primarily geared towards programmers, to facilitate a coding or gaming background. However, a gaming approach does not capture the full scope of architectural approach to content creation[2]; a design-based and iterative approach is more applicable to the architectural practice.[3] To be able to judge practicality, the architectural practitioner's experience of creating VR experiences must be challenged.

To establish the parameters of experimentation, this thesis evaluates VR content creation from the standpoint of an average design consumer or end-user of VR. The study is done in the context of research of a postgraduate student with qualifying experience and education in design. As such, the standards involve possessing a knowledge of architectural software without having the background of a developer or adept programmer. In addition, this research is framed around a student's experiences within the architectural profession, and is grounded in the architectural design process - rather than within the programming field. While personal knowledge of game engine software is limited, this can be attributed to that of an entry-level architectural designer. To produce a VR experience for architectural standards, this thesis is segregated into small tasks, each of which seek to evaluate hardware and software parameters. To evaluate success in VR progress, a decision of using small stages to evaluate progression through education of the software/hardware was made. Also, to facilitate this, each chapter to follow includes a precise research question, as well as a corresponding experiment completed within the context of the architectural practice. Each experiment is supplemented with steps, an allotted timeframe for documentation of efficiency and a set of evaluation parameters to evaluate efficacy.

Assessing the ease and accessibility of VR usage requires strength in both efficacy and efficiency for VR to be a convincing imperative to architectural practitioners. Uncompromising quality or effectiveness is quintessential for designers to showcase a project to any presentation to clients or peers. Efficiency is integral as the design profession thrives on completing tasks to meet short deadlines. Ideally, it is hypothesized that a short amount of time yields a highly effective virtual reality simulation for the architectural practice within investigation.

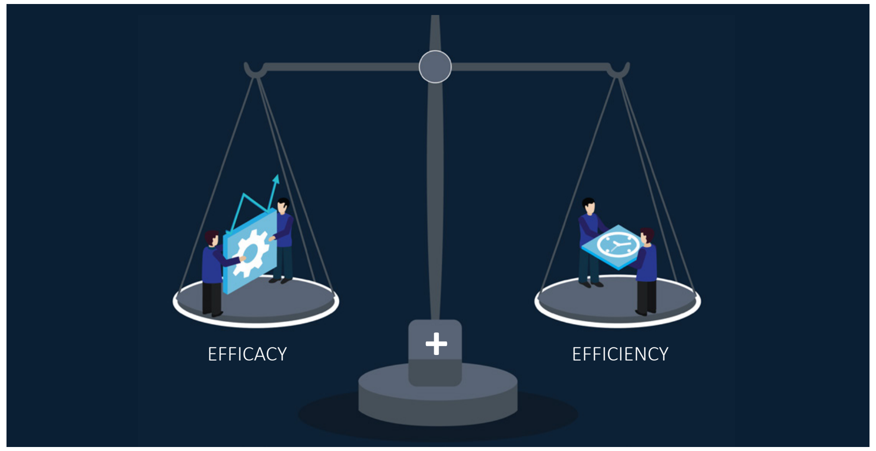


figure 4.01 Efficacy and Efficiency As Equal Requirements

---

[1] Gartner Inc., "3 Reasons Why VR and AR Are Slow to Take Off: Here's how technology product managers should plan for virtual reality and augmented reality adoption in the next few years." accessed October 23, 2018, https://www.gartner.com/smarterwithgartner/3-reasons-why-vr-and-ar-are-slow-to-take-off/.

[2] Erica Brett, "Architects in the Design of Virtual Reality Spaces" (UC Berkeley College of Environmental Design, 2015-2016), accessed October 10, 2018, http://ced.berkeley.edu/downloads/thesis/arch/2016/Brett_Erica_Architects_in_the_Design_of_Virsua__Reality_thesis.pdf.

[3] Gilles Delalex, Go with the flow: Architecture, infrastructure and the everyday experience of mobility, Publication series of the University of Art and Design Helsinki A55 ([Helsinki, Finland]: [Univ. of Art and Design Helsinki], 2006).

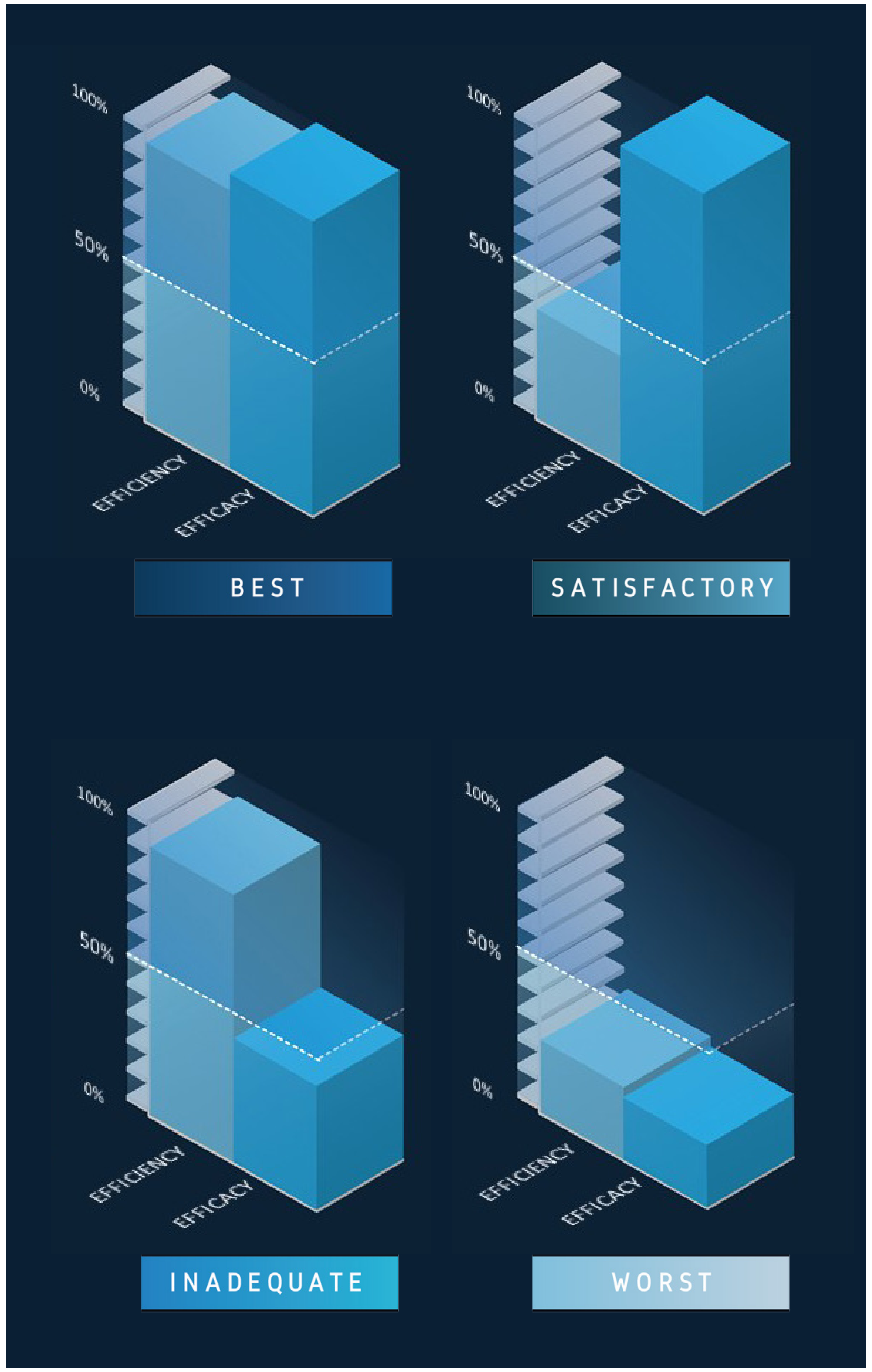


figure 4.02 Case Scenarios for Evaluating Efficiency & Efficacy

VR Hardware: Room-Scale (tethered) vs. Stationary (mobile)

Current consumer-based headsets in the market fit between two categories; mobile or tethered. Mobile headsets are simply shells with lenses into which a smartphone is placed.[1] The user while able to view and hear the VR simulation, will not be able to physically move anything but their head to access other views within the rendered scene. Hence, this experience is physically stationary as the user will remain seated or standing for the majority of the experience. First-generation mobile headsets allow for exclusively stationary experiences. Gyroscopic sensors and accelerometers present in mobile devices sense head rotation, allowing a user to look around for their stationary head-rotation based VR experience. Some common examples of mobile headsets include the Samsung Gear VR, Google Daydream and Google Cardboard.[2]

Conversely, if a user requires being able to move their head and walk around at the same time, additional external sensors and video processors are required to track the physical position of a user. Experiences that track this physical position along with head rotation are referred to as room-scale VR experiences. Room-scale experiences are created with tethered headsets, which come with built-in motion sensors and external hardware in the form of camera trackers to allow for a more complex VR experience. Tethered headsets such as the Oculus Rift and HTC Vive are connected to personal computers (PCs) and are able to place all the computational load of video processing into the PC itself.[3] A room-scale VR experience thus offers more promise in assessing applicability in architecture as it allows a user to freely walk around their play area of VR space, translating physical movements from reality into their digital environment.[4] As the cameras monitor a user's environment in 3D space, the user can feel a greater sense of immersion with the digital translation. Walking across a room digitally is equated to walking around a physical room and this simple factor can provide power to architectural visualizations. However, as architects usually have to showcase projects that are larger than room-scale, teleportation becomes a necessary additional factor of implementation. Tethered headsets allow for users to remain in control of teleporting through their environments. Additionally, their dedicated display rather than that of a smartphone as used in mobile headsets improves image fidelity and tracking.[5]

---

[1] Diversified Internet Holdings LLC, "The Ultimate Guide to Understanding Virtual Reality (VR) Technology," RealityTechnologies.com, accessed October 25, 2018, http://www.realitytechnologies.com/virtual-reality.

[2] Mealy, Virtual & augmented reality for dummies.

[3] Ibid.

[4] Bailenson, Experience on demand.

[5] middleVR, "The challenges of creating a VR application," accessed October 23, 2018, https://www.middlevr.com/resources/the-challenges-of-creating-a-vr-application/.

Table 1 Mobile VR Systems

MOBILE VR SYSTEMS	Samsung Gear VR	Google Daydream	Google Cardboard
MSRP CAD (2018)	$99.00	$139.00	$15.00
Platform	Android	Android	Android, iOS
Experience	Stationary	Stationary	Stationary
Field of View	101 degrees	90 degrees	Varies (90 degrees)
Resolution	1,440 x 1,280 Super AMOLED	Varies (Pixel XL 1,440 x 1,280 AMOLED)	Varies
Headset Weight	0.76 pounds without phone	0.49 pounds without phone	0.2 pounds without phone
Refresh Rate	60 Hz	Varies (minimum 60 Hz)	Varies
Controllers	Headset touchpad, single motion controller	Single motion controller	Single headset button

****OLED: organic light-emitting diode*

Table 2 Tethered VR Systems

TETHERED VR SYSTEMS	HTC Vive	Oculus Rift	Windows Mixed Reality	Playstation VR
MSRP CAD (2018)	$799.00	$599.00	$695.00	$399.00
Platform	Windows, Mac	Windows, Mac	Windows	Playstation 4
Experience	Stationary, Room-Scale	Stationary, Room-Scale	Stationary, Room-Scale	Stationary
Field-of-View	110 degrees	110 degrees	Varies (100 degrees)	100 degrees
Resolution per Eye	1,080 x 1,200 OLED	1,080 x 1,200 OLED	Varies (1,440 x 1,440 LCD)	1,080 x 960 OLED
Headset Weight	1.2 pounds	1.4 pounds	Varies (0.375 pound)	1.3 pounds
Refresh Rate	90 Hz	90 Hz	Varies (60-90 Hz)	90-120 Hz
Controllers	Dual-motion wand controllers	Dual-motion controllers	Dual-motion controllers, inside-out tracking	Dual PlayStation move controllers

****AMOLED: "active-matrix' organic light-emitting diode*

figure 4.03 Leading VR Headsets

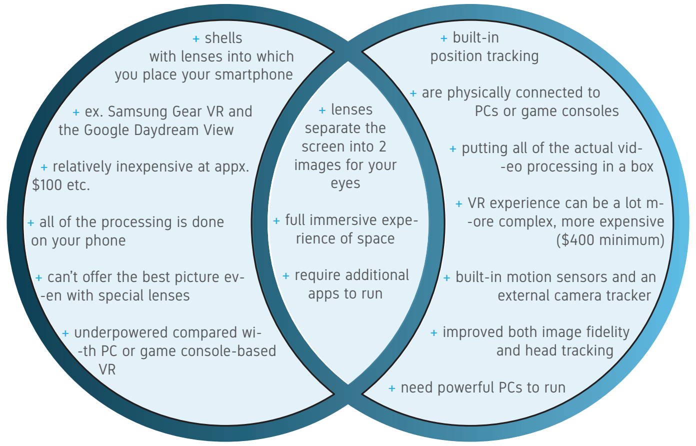


figure 4.04 Venn Diagram Differentiating Headsets

Oculus Rift, HTC Vive, Samsung Gear VR, Google Daydream and Google Cardboard (along with any other HMDs mentioned previously in this chapter) represent the first generation of devices.[1] HTC Vive and Oculus Rift (being tethered and well-tested consumer based headsets) represent the higher end of VR experiences available at the time of writing. Between the two higher-end devices, the Rift was selected as the medium for this thesis given its greater consumer base. As only first-generation devices are out in the market and have been thoroughly tested by consumers, the VR hardware used in this thesis is prone to drastic change. Recently, for example, hardware manufacturers have released specifications for headsets with integrated processing, audio and motion controllers as a base setup for high-end experiences.[2]

---

[1] Mealy, Virtual & augmented reality for dummies.

[2] Ibid.

Norm Li, architect and visualization specialist, claims that the primary success of VR is within mobility. He claimed in an interview, “Why spend a million and half on a presentation centre when you can – with VR – take your project right to the buyer? Or create a pop-up in an empty storefront?”[1]. Hence, the variety in hardware executions of VR experiences is subject to a form factor change in the upcoming years, in an effort to achieve greater mobility. This change is forthcoming with the release of headsets such as the Oculus Rift S, Project Alloy, Snapdragon VR, Tango, Windows MR Tracking and WorldSense, which are all “marker-less inside-out” tracking systems.[2] These systems eliminate the need for external sensors, allowing for unconfined movement and unlimited physical transitions (provided the user has the empty physical space for it).[3] The elimination of external sensors comes with a cost, unfortunately, as environmental tracking accuracy will suffer to an extent. The controllers can easily fall out of the line of the user’s gaze, resulting in an inability to track controller input. This can lead to a user losing their sense of place easily within a virtual scene.

As Oculus Rift Founder Palmer Luckey has stated, the VR experience is incomplete with “just the visual side”. Palmer claims the absolute need for a fully-integrated input and output system, “so you not only have a natural way to view the virtual world, but also a natural way to interact with it.”[4] Natural interaction can be amplified through factors such as haptic feedback and 3D audio, which activate touch and sound senses.[5] Haptic feedback allows for the user to experience touch sensations by using vibrations to provide contextual information.[6] Haptic hardware systems are retrofitted suits and gloves, that translate sensations and vibrations to a user, as an expensive addition to virtual environments.[7] 3D audio on the other hand, is more commonly employed in scene creation as the hardware requirements are simply a pair of integrated headphones. 3D audio engages a user by enabling them to sense realistic sounds in a simulated environment and activate their sense of presence.[8]

---

[1] Totzke, “The Real and Virtual Norm Li”

[2] X in Reality, “Markerless inside-out tracking,” accessed October 24, 2018, https://xinreality.com/wiki/Markerless_inside-out_tracking.

[3] Ibid.

[4] Mealy, Virtual & augmented reality for dummies.

[5] James J. Cummings and Jeremy N. Bailenson, “How Immersive Is Enough? A Meta-Analysis of the Effect of Immersive Technology on User Presence,” Media Psychology 19, no. 2 (2016), https://doi.org/10.1080/15213269.2015.1015740, https://vhil.stanford.edu/mm/2015/cummings-mp-how-immersive.pdf.

[6] Chris Dede, “Immersive interfaces for engagement and learning,” Science (New York, N.Y.) 323, no. 5910 (2009), https://doi.org/10.1126/science.1167311.

[7] Cummings and Bailenson, “How Immersive Is Enough? A Meta-Analysis of the Effect of Immersive Technology on User Presence”

[8] Thackery I. Brown et al., “Prospective representation of navigational goals in the human hippocampus,” Science (New York, N.Y.) 352, no. 6291 (2016), https://doi.org/10.1126/science.aaf0784.

Sensations through haptic feedback and spatial/ binaural sounds can aid in an understanding of space and presence by providing directional, auditory and visual cues.[1] Unfortunately, advanced haptics are extremely difficult, expensive and effort intensive to implement in virtual reality experiences that exist outside of a laboratory. Haptic feedback technology has also not evolved to a mature state for easy implementation with standard controllers like the HTC Vive or the Oculus Rift. Yet, as any aid in activating presence is beneficial to creating virtual scenes, 3D audio implementation will play an integral role in scene creation experimentation.

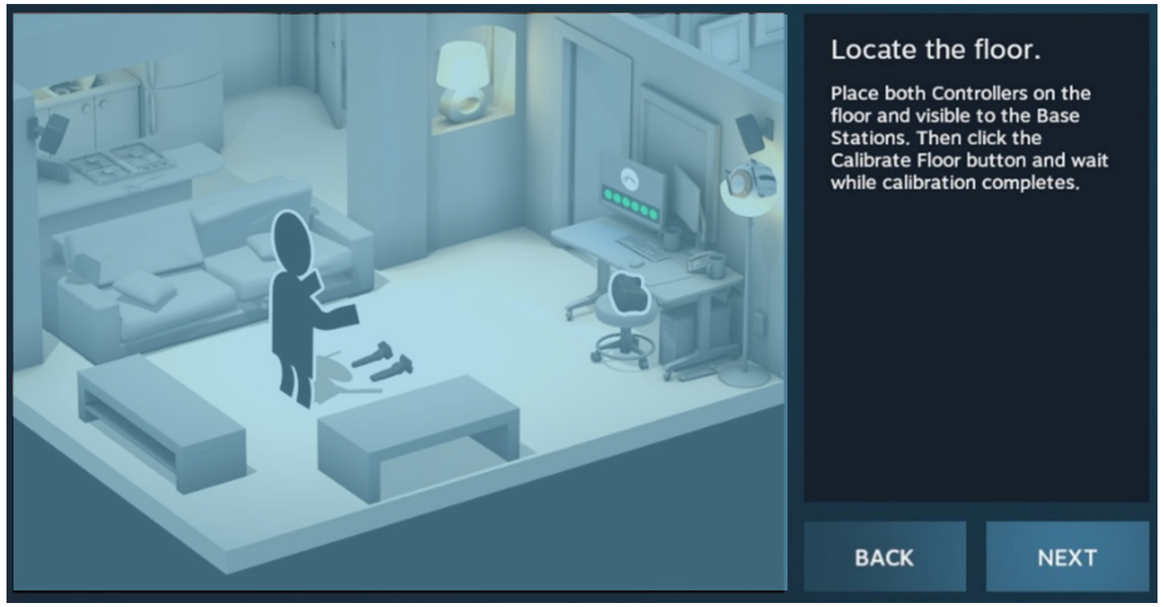
Hardware Options: Differentiating Immersion

This thesis primarily focuses on the use of entry-level and high-end consumer VR hardware, to examine the range of hardware options available to architectural designers. However, advanced hardware currently used in the laboratory setting is an important consideration for future research done within this topic. Once advanced VR hardware reaches a level of accessibility for the architectural practice, this experimentation would be worthwhile to recreate. As of the time of writing, VR has been shown to successfully trick the human visual sense into seeing a different environment than the one physically inhabited.[2] Even so, there are other factors that contribute to the illusion of being in another location. Entry-level VR hardware all activate rotational head tracking. This allows a user to be drawn to in interacting with their environment through sight. To add a layer of immersion, high-end consumer VR, with the use of sensors and controllers, is able to offer positional tracking. Positional tracking lets a user to interact with a scene and employs human proprioceptive cues. This is often achieved with the use of binaural audio, which allows a user to hear sound naturally as it is spatialized (which makes sounds louder or softer depending on the sound source). Advanced VR adds the final layer of immersion to an experience, by allowing for haptic and sensorial tracking, which seeks to map physical sensations in the form of vibrations and temperature changes both to objects and users. Beyond the sense of sight, sound and touch uncovered with each level of immersion, VR has a difficulty in recreating smell and taste sensations. Yet, even without smell, taste and -with current consumer VR- touch, VR has been able to successfully represent space more naturally. The most natural and comfortable VR solutions currently are advanced VR options, which are extremely expensive and labour-intensive to implement.

---

[1] Matthew Lombard and Theresa Ditton, “At the Heart of It All: The Concept of Presence,” Journal of Computer-Mediated Communication 3, no. 2 (1997), https://doi.org/10.1111/j.1083-6101.1997.tb00072.x, http://dx.doi.org/10.1111/j.1083-6101.1997.tb00072.x.

[2] Jason Jerald, The VR Book: Human-centered design for virtual reality, First edition, ACM Books # 8 (New York, NY: Association for Computing Machinery and Morgan & Claypool Publishers, 2016).



Locate the floor.

Place both Controllers on the floor and visible to the Base Stations. Then click the Calibrate Floor button and wait while calibration completes.

figure 4.05 SteamVR Tethered Headset Tracking Setup

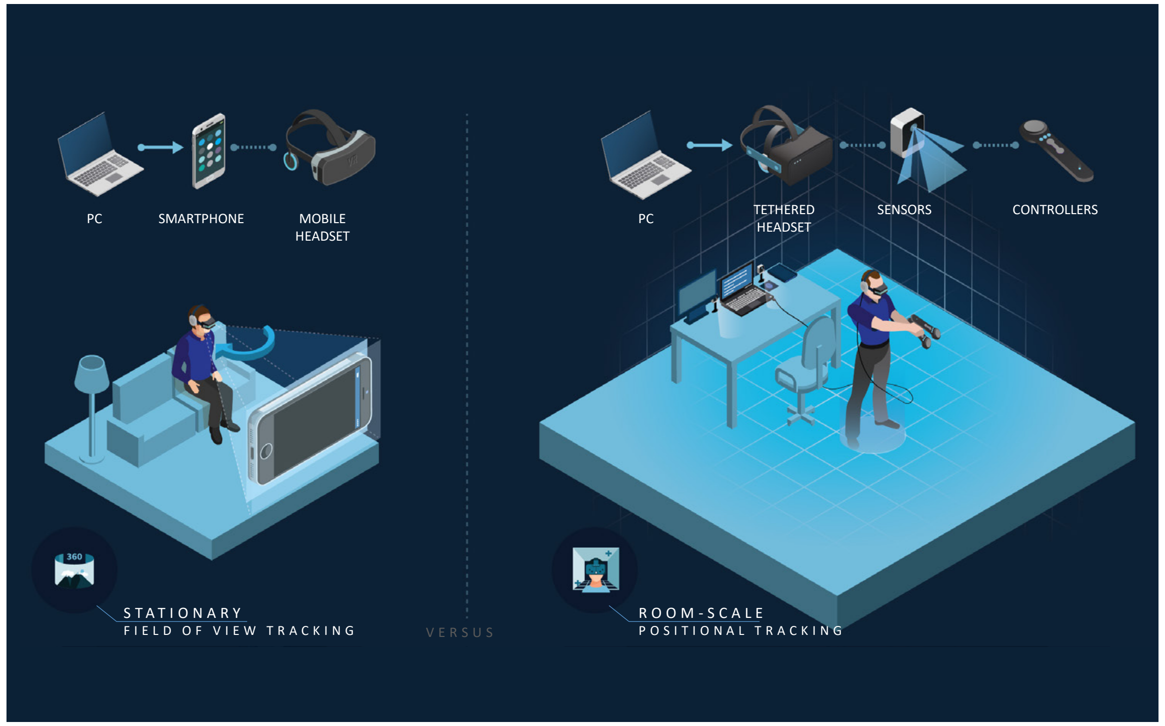


figure 4.06 Stationary vs. Room-Scale Experiences

Some of these implementations include an omnidirectional treadmill[1], full haptic suits[2] or "warehouse-scale" experiences with backpack computers[3]; all of which allow a user to freely walk within their environment and experience the added sensations of their environment. The higher level of immersion that advanced hardware can provide is an important consideration from the perspective of a designer. However, these implementations are often expensive and difficult to physically recreate for the architectural design profession. Bearing this in mind, this thesis focuses on the entry-level and high-end consumer levels of immersion with VR hardware, rather than advanced VR options to justify practicality within the architectural field.

VR Software: 3D Image Creation and 4D Scene Manipulation

The architectural design process has undergone transformations due to the advent of computer-aided design tools.[4] In today's world, design within any conceptual phase is based on iterative 3D model-making which then derives the 2D manifestations of designs (namely plans, sections, elevations).[5] VR has the capability to enhance client offerings by improving the communication of ideas to clients.[6] If a 3D viewable model can be easily transported into a 4D scene for VR viewing, it can enhance a client's understanding of the unbuilt environment by allowing them to interact with it in a more realistic way, without having to envision that design just from orbting around 3D model on a desktop screen or from looking at image boards. If the most common computer-aided design tools can export a model for import into a game engine for VR scene creation to be used to create a simple walkthrough, the design process might gain tremendous value in using VR as part of the architectural design process.

The experimentation of this thesis will therefore focus on converting 3D architectural models into 4D experiential content. For many individuals, VR is as simple as downloading the Oculus Rift plugin and starting up an application for a game they already have downloaded from an online server. For architects, VR will be more complex than that simply because each application will be created by oneself for one's own architectural project.

---

[1] Virtuix, "Move Around in VR," accessed February 1, 2019, https://www.virtuix.com/.

[2] Teslasuit, "Teslasuit: Ultimate Tech in Smart Clothing," accessed February 1, 2019, https://teslasuit.io/.

[3] Hypercell, "Hypercell Theme Park: Global Franchise Package: Wireless Full Action Up to 5000 Sq.m." accessed February 1, 2019, http://www.hypercell.com/.

[4] Botchway E. A. Abanyie SA, "The Impact of Computer Aided Architectural Design Tools on Architectural Design Education. The Case of KNUST," Journal of Architectural Engineering Technology 04, no. 02 (2015), https://doi.org/10.4172/2168-9717.1000145.

[5] Dino Bouchlaghem et al., "Visualisation in architecture, engineering and construction (AEC)," Automation in Construction 14, no. 3 (2005), https://doi.org/10.1016/j.autcon.2004.08.012.

[6] Bouchlaghem et al., "Visualisation in architecture, engineering and construction (AEC)"

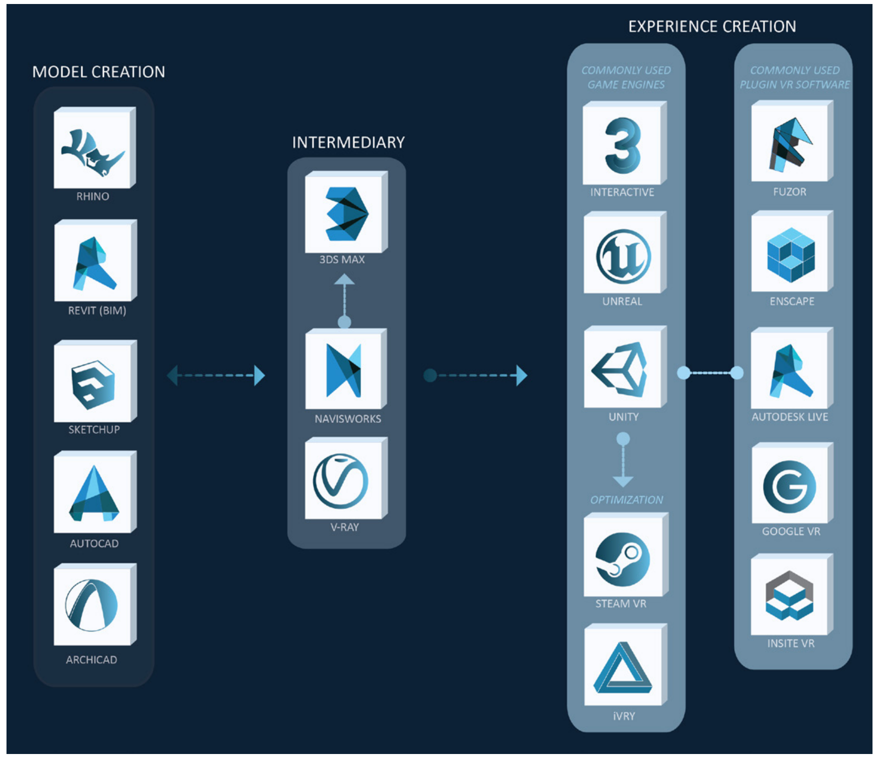


figure 4.07 Architectural Workflow from Digital Model to Experience

Starting with an architectural model built by a common architectural modeling software (such as Revit, SketchUp, Rhino3D, ArchiCAD, Maya, AutoCAD, 3dsMax and VectorWorks)[1] and passing the model into a 4D content creation option, VR experiences will be built. 4D Content for VR experiences is primarily developed using game engines, such as Unity 3D and Unreal Engine 4, making them the primary focus of experimentation in this thesis. Each existing project in any software used for model creation will have to be exported, passed through an intermediary to convert textures and material properties (ensuring retention during translation) and then imported into Unity/Unreal for scene creation and deployment onto one's own specific hardware. As an alternative method, direct integration programs are also an available option for 4D content generation (discussed further in Chapter 8). By exploring architectural project exports into game engines/ direct integration programs, the practicality of VR content creation in the architectural field will be evaluated throughout this thesis.

---

[1] Easy Render, "Latest advancements in architectural rendering software," accessed November 8, 2018, https://www.easyrender.com/architectural-rendering/latest-advancements-in-architectural-rendering-software.

Software Options: Differentiating Between Game Engines

Table 3 Leading Game Engines

	CryEngine	Unreal Engine 4	Ogre	Unity 3D	Project Anarchy
Entry Level	Very High	Medium	High	Low	Medium
Language	C++	C++/Blueprint (Unreal Script)	C++	C#, JavaScript	C++
Built-in AI System	Yes	Yes	No	Yes (in Pro)	Yes
Community	Small	Large	Small	Huge	Medium-sized
PC Requirements	High	High	Low	Medium	Medium
Cost	Yes	No	Yes	No	No
Graphic Quality (i.e. features)	Very High	Very High	Medium	Medium	Medium

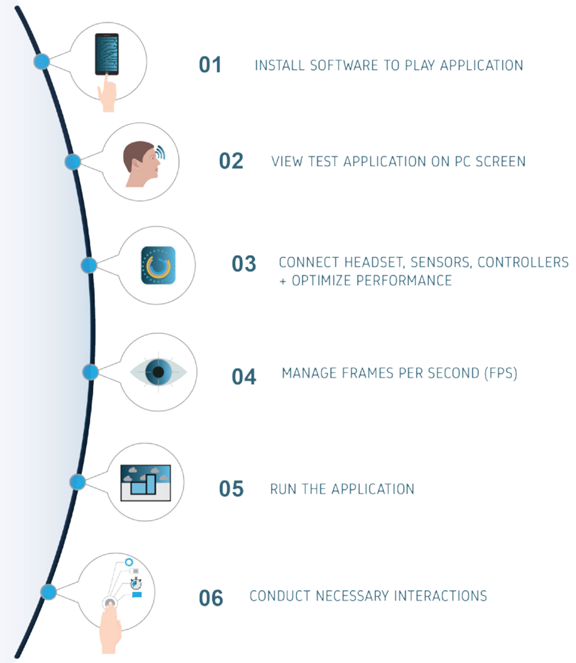


figure 4.08 VR Deployment Process

Tactics of the Trade

The usage of Virtual Reality is primarily geared towards focusing a user's attention to the screen before their eyes. This is often easier said than done, because in a VR simulation, a user still has the freedom of choice to look wherever they please. However, with a narrative or storytelling feature, a user may be propelled to conduct specific actions or focus their attention towards a specific feature. To facilitate user focus, adept content creators have noted successful techniques to keep a user engaged in the virtual simulation. Introducing tactics such as subtle 3D audio cues, guided narration, lighting cues (e.g. brightening objects that should be looked at, darkening objects that are to be avoided) and content reorientation can add to "telling the story" of the environment for the user. When presenting objects within a VR scene, there are maximum, minimal and optimal distances for objects such that they achieve their intended purpose. This directly links to the theory that as an object gets closer, the human eye begins to strain to focus. Oculus developers recommend a minimum distance of viewable objects to be placed at 0.75m to prevent eye strain, the effects of which fade considerably between 10 and 20m. Similarly, a comfortable range of motion for a user to rotate their head vertically or horizontally is between 30 and 55 degrees.[1]

When focus objects are framed within these viewing distances, the simulation is able to provide the proper feedback and a user's gaze can follow their environment adequately. As VR is a highly engaging experience, a user needs to be directed to pay attention to the important things. Along with best practices, there are also common guidelines for what should not be done with VR. Simulator sickness has to be avoided at all costs within any VR experience. To facilitate this, any application has to maintain a frame rate of 60 fps or higher and avoid fast acceleration/deceleration.[2] In the physical world, physical sensations are often accompanied by visual cues. As VR facilitates a primarily visual response, a user needs to always be moving at a constant velocity and avoid fixed-view items so as not to get sick. Similarly, bright colours and environments can trigger a physical response akin to staring at the sun. When done right, the cumbersome equipment vanishes along with any physical discomfort, giving a user the space to enjoy and focus on the virtual environment before them.

---

[1] Paul Mealy, Virtual & augmented reality for dummies, 1st edition (Indianapolis IN: John Wiley and Sons, 2018).

[2] RealityTechnologies.com, "Virtual Reality: The Ultimate Guide to Understanding Virtual Reality (VR) Technology," accessed September 5, 2018, https://www.realitytechnologies.com/virtual-reality/.

Current Limitations and Issues in VR

Although VR is readily available to the consumer market, it has not passed being owned by just a niche market.[1] Gartner claims VR requires strength in two sectors: convenience and control. To be a very user-friendly product and reach its untapped market, it needs to gain traction in availability and access. The hardware has to be intuitive and offer experiences that don't compromise on affordability to create the sensation of reality. By current standards, not all VR experiences have been proven to achieve levels of gesture and movement control people expect.[2] This disappointment can be recognized simply because in reality, every person's actions are usually met with a certain physical response, visual or otherwise. For example, if hot garlic bread is touched or a toe is slammed on a door, a burning or tingling sensation is gained in the form of tactile feedback to the human nervous system.[3] A recreation of all these types of physical responses while in a virtual simulation, is challenging to achieve. Currently, haptic cues that go farther than vibrations have not been developed or executed safely in VR.[4] Also, since physical movement in VR is only tracked in a single room and not any farther, movement over a long distance is unprecedented without the use of teleportation.[5] Large distances, like that of a walkthrough in even a simple residential housing project are extremely difficult without the use of teleportation. Even through accessing teleportation, a user has to have a large empty physical space in the room to experience the visualization of a small room without bumping into physical objects. Each scene also requires the creation of digital barriers (or colliders) to prevent a user from exhibiting "ghost-like" tendencies in the simulation. Overall, VR walkthroughs require users to physically travel much farther than what their physical space can generate.[6] Using controllers for locomotion solves this issue of physical movement to an extent by allowing the user to have the experience of locomotion while in one place.[7] Yet, the user can still only move around in the virtual world by distances allowed in their physical setup. Understanding the maximum distances and haptic cues that the Oculus Rift provides is quintessential to walkthrough scene experimentation in this thesis.

---

[1] Gartner Inc., "3 Reasons Why VR and AR Are Slow to Take Off: Here's how technology product managers should plan for virtual reality and augmented reality adoption in the next few years." accessed October 23, 2018, https://www.gartner.com/smarterwithgartner/3-reasons-why-vr-and-ar-are-slow-to-take-off/.

[2] Bailenson, Experience on demand.

[3] Mealy, Virtual & augmented reality for dummies.

[4] Nick Yee and Jeremy N. Bailenson, "Walk a mile in digital shoes: The impact of embodied perspective-taking on the reduction of negative stereotyping in immersive virtual environments." (Cleveland, Ohio, Stanford University, 2006), https://vhil.stanford.edu/mm/2006/yee-digital-shoes.pdf.

[5] middleVR, "The challenges of creating a VR application"

[6] Ibid.

[7] Abrash, "Welcome to the Virtual Age"

The greatest struggle of VR technology is its accompaniment with simulator sickness. Although this aspect was discussed briefly in Chapter 2, there is a need to define this term specifically and explain its constraints. Simulator sickness can occur as there are incoherent vestibular signals between what the eyes see and the human inner ear's vestibular sense of motion.[1] Latency occurs if there is a delay between natural head motion and a shift in the virtual perspective, which can result in a disruption of balance. When there is latency present in a simulation, the human body sends signals to the brain indicating the entrance of a toxin or affliction.[2] As a reaction to the inconsistency, the brain subsequently induces headaches, dizziness, disorientation and in severe cases, nausea. Hence, latency has to be avoided both while developing VR applications and choosing VR hardware. Latency is measured in frames per second, or FPS, which calculates how fast the user's movement in relation to their visual world can be optimized. To achieve a successful VR application, the FPS must be as high as it can be.[3] The greater the power of the device, the greater the ability of achieving a smooth and well-processed visual scene. A higher frame rate indicates a faster response and better feedback. Smooth VR experiences without latency occur above 60 Hz. Yet, due to the severity of precautions against simulator sickness, in the creation of any application for VR, it is advised to monitor the FPS and keep the rate at 75-90 Hz for maximum efficacy. Furthermore, due to the risks associated with latency, the user market is reduced due to cautions against persons who are pregnant, elderly, fatigued or diagnosed with cardiac ailments. As far as any research shows, any adverse health effects experienced are short-term, and have no lasting effect on the user. In the decades following the invention of VR, the latency gap present in devices made VR unusable. Currently, the onset of mass VR adoption exists because present hardware has overcome this issue by providing stronger processing speeds, allowing for lower latency VR content development.

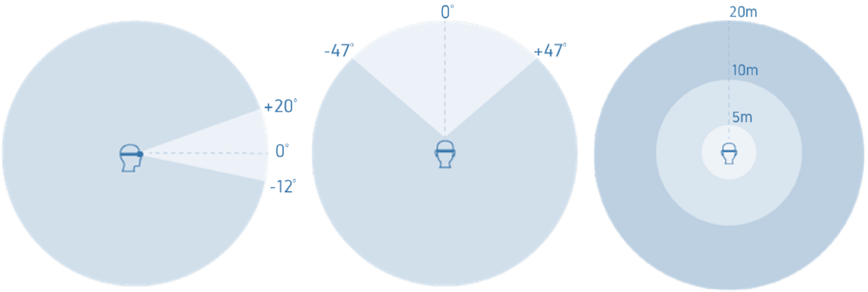


figure 4.09 Viewing Distances and Affordances

---

[1] Lombard and Ditton, "At the Heart of It All: The Concept of Presence"

[2] Bailenson, Experience on demand.

[3] middleVR, "The challenges of creating a VR application"

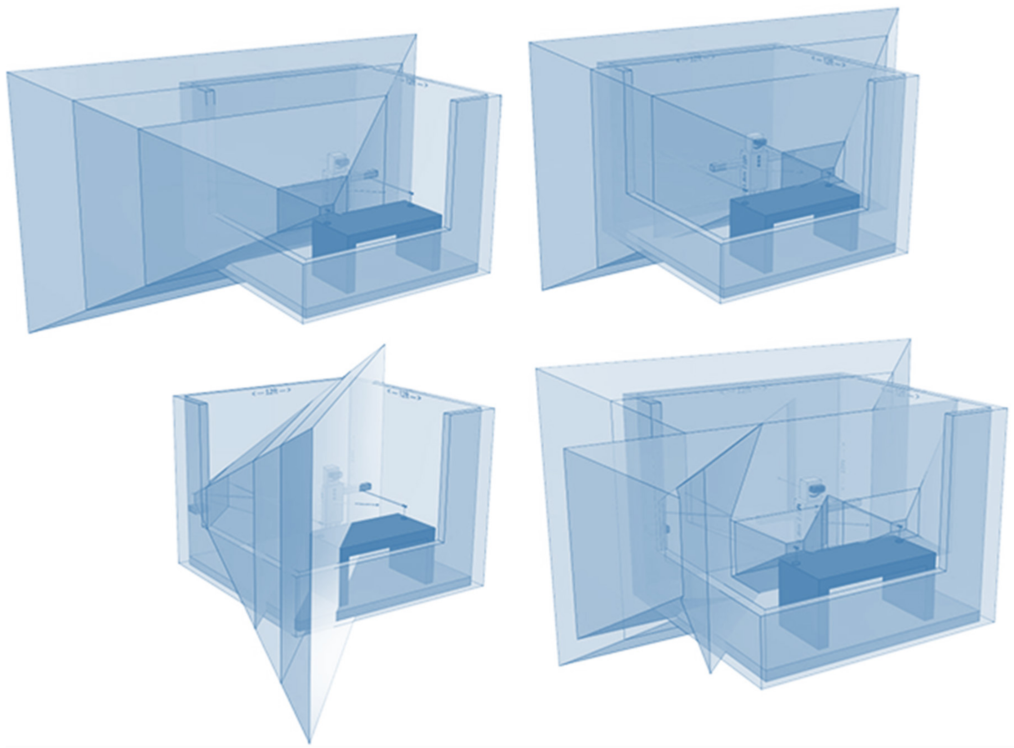


figure 4.10 Oculus Room-scale "Tips for Setting Up a Killer VR Room"

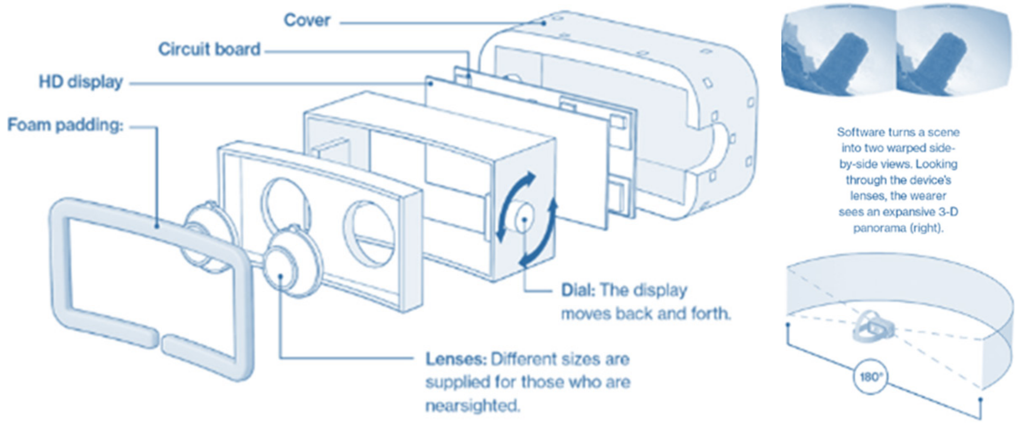


figure 4.11 Inside View of Oculus VR's First Commercial Headset

Chapter 5: Stereoscopic Scene Creation

Research Q1: Is a Unity3D VR application simple to create and easy to access?

To learn how VR experiences are built, this first experiment attempts the creation of a stationary VR experience by converting a render into viewable VR content. In this experiment the primary goal will be to produce a stereoscopic scene given a basic rendered architectural image for deployment in any accessible mobile headset. Within this chapter, an evaluation will be made of the ease and accessibility of creating content in the first game engine explored, Unity 3D, with a render created from a project of a living room space. This task represents the first attempt at understanding virtual reality's connection to visualization by creating a stationary VR experience. The method used to convert a 2D image into 4D content follows the principles of stereoscopy. Stereoscopy is the means by which images are converted into 4D content viewable with a VR headset. Creating a stereoscopic scene is the first stage of experimentation as it will allow for an understanding of the basics of the Unity game engine, a navigation of the user interface and assessment of the program's difficulty level for VR experience creation. As the premise of this experiment, stereoscopy can achieve the depth perception of an image and provide additional value in allowing a client to see a render at a real-life scale. Stereoscopic pictures are produced as image pairs showing the same environment from slightly different angles made. These images will then correspond directly to the physical properties of binocular vision (where the left-eye view and right-eye view are angled in the vision of when a person looks at an object).[1] In the brain, stereoscopy is achieved by combining these separate perceptions from each eye and interpreting them in terms of depth, distance and objects viewed.[2] Hence, by feeding two spherical images (a right-eye image and left-eye image) within Unity and deploying it for a mobile phone (for mobile headset usage), an assessment can be made of the ease of application creation, accessibility with Unity 3D and success of stereoscopic scene creation. Though this VR experience does not offer the ability to walk around a virtual environment, a user can still look around and feel a part of the space, aptly making this experiment the first step to understanding visualization in virtual reality.

---

[1] The Editors of Encyclopaedia Britannica, ed., Encyclopaedia Britannica: Stereoscopy (Encyclopædia Britannica, inc., 2013); Optics, https://www.britannica.com/technology/stereoscopy, accessed November 12, 2018.

[2] Ibid.

As this experiment features a stationary VR experience, the only interaction a user has is visual. This strictly visual interaction can be understood at a greater depth in this chapter by defining field-of-view (FOV) and Stereoscopy. All virtual reality headsets track field-of-view, which is the area or range a user can be reasonably expected to see with their headset at any one time. The wider the FOV, the more positionally accurate objects are in their placement with regard to a user. This can impact the feeling of presence within a space and also helps determine setup of objects or cameras within a VR scene. While a wider field-of-view is important to achieve immersion and presence, a stereoscopic binocular area of the full human field of view is where most visual perception occurs. Stereoscopy, and binocular vision is the principle that is required for images and content in VR to provide a 360-degree view and enable a user to look around. By delivering two images from two unique cameras (which are slightly offset from each other), it mimics what the real human vision achieves. Stereoscopy can be configured for scenes as well to provide a 360 degree view when a user stays in a still position[1]. Nonetheless, when it comes to a scene created within a game engine, binocular vision is already achieved and the scene is fed into each eye in real-time as it is built for virtual reality, dismissing the need too actively associate separate perceptions of a scene to each eye. Stereoscopy is, however, necessary to implement to be able to look at pre-rendered images or content in 4D. Any pre-rendered scene built with a game engine is not automatically stereoscopic in 360 degrees, as it has to be associated to each eye. Both FOV and stereoscopy can be used to understand how a user given the perception of a virtual experience.

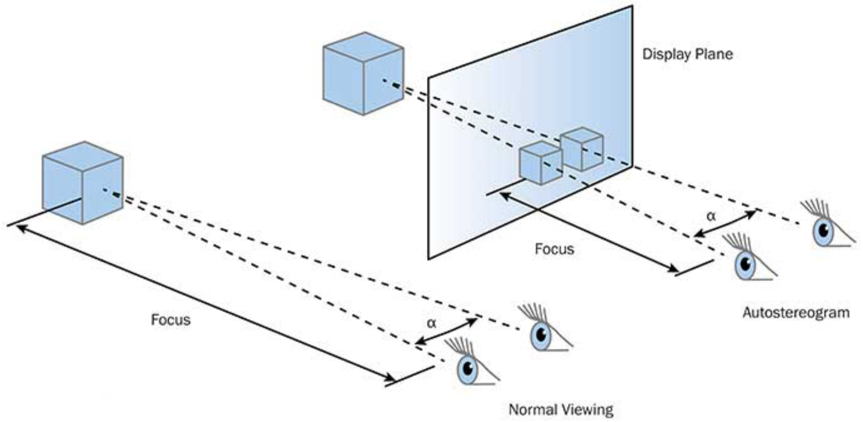
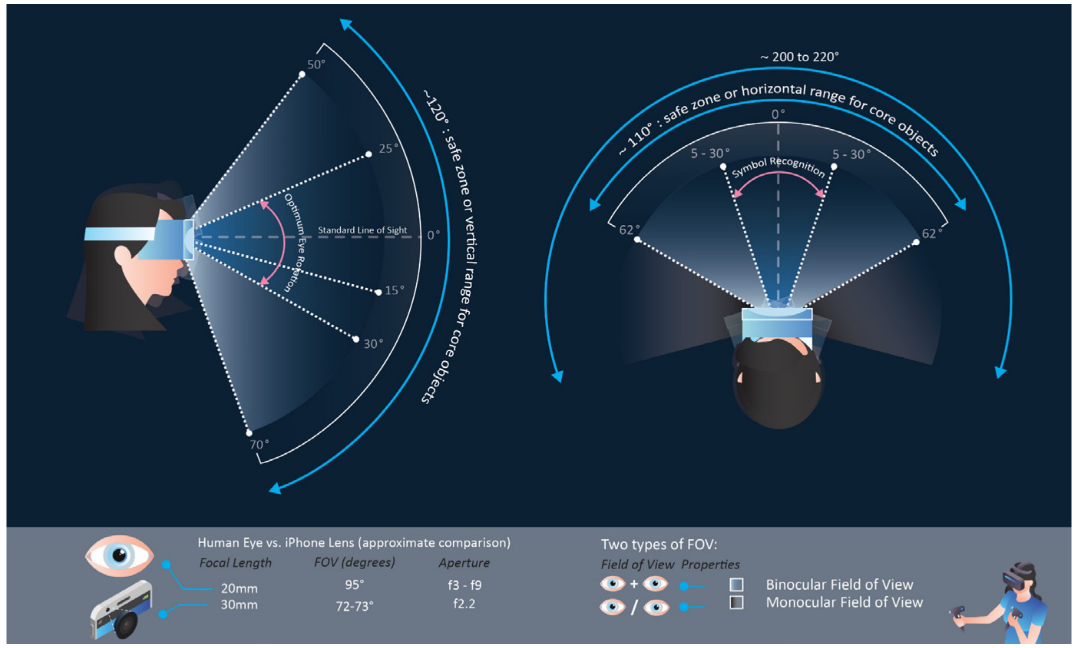


figure 5.01 Autostereogram vs. Normal Viewing

---

[1] Unity Technologies, "Stereo 360 Image and Video Capture," accessed March 2, 2019, https://blogs.unity3d.com/2018/01/26/stereo-360-image-and-video-capture/.



Human Eye vs. iPhone Lens (approximate comparison)

Focal Length	FOV (degrees)	Aperture
20mm	95°	f3 - f9
30mm	72-73°	f2.2

figure 5.02 Field of View (FOV) in Virtual Reality

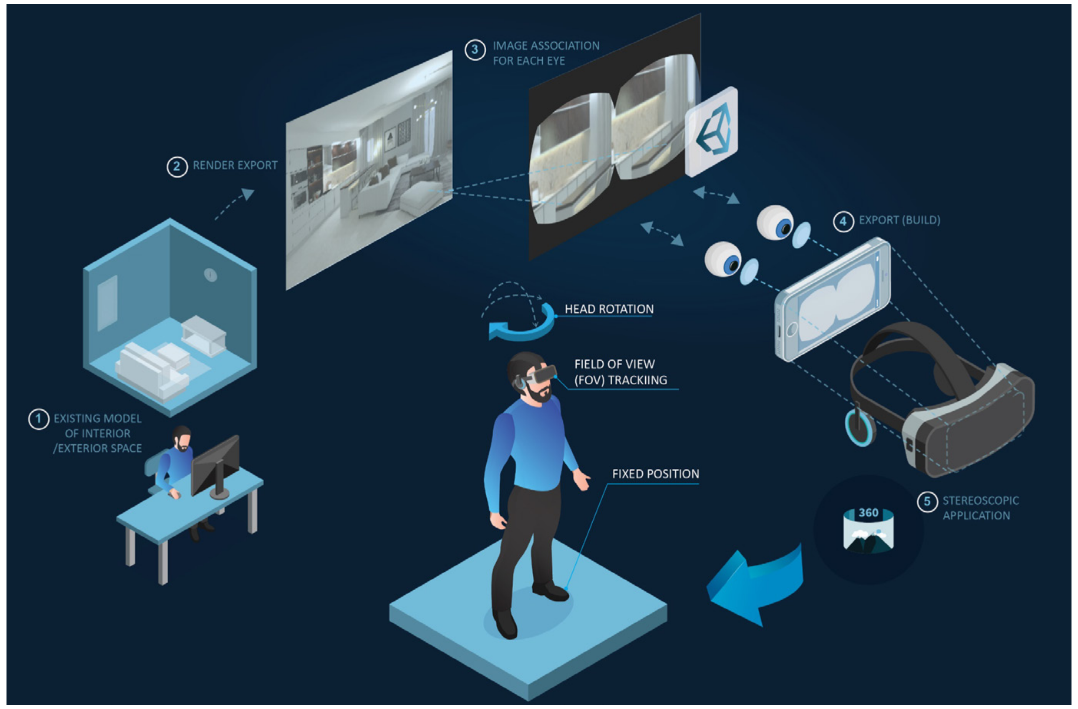


figure 5.03 Stereoscopic Scene Creation Process

Definitions[1]

1. Stereoscopy: seeing an image in three dimensions using principles of binocular vision.[2] Stereography is the means of delivering this, achieved through the creation of two images from separate cameras (slightly offset from one another) to mimic what the eyes would see at their respective angles. This act provides the experience of something that appears to be 3D, as mimicked to be from one's own eyes.[3]

2. Interocular Distance (IOD): an essential factor in determining a scene's scale in relation to its viewer. This factor measures the distance between the two left and right cameras that render in stereo. If the distance is too small or too large, the viewer's body is scaled disproportionately to the world around them.[4] A VR experience can't exist with disproportions because a viewer experiences what is occurring as if through their own eyes. IOD thus needs to be established correctly in a scene to resemble accurate physical proportionality.

3. Cubemaps: an efficient way to process 360 degree images. Cubemaps place six images on the inner surfaces of a virtual cube. The net of these six images is then unraveled to be placed side-by-side as a single 6:1 aspect image. The image stitched together from these six then appears as a full 90-degree FOV image.[5]

4. GameObjects: the essential building tools in Unity. Every object in the game is a GameObject, including (but not limited to) characters, props, lights, special effects and cameras. They are empty vessels for which properties need to be assigned. Once assigned, they contain functional attributes, which are referred to as components, that allow it to become a character, environment or special effect. GameObjects can be compared to the creation toolbars/toolkits prominent in any modeling or drafting software; that allow for the conception of simple surfaces, lines, or solid shapes (baseline objects).

5. Components: functional capabilities attributed to GameObjects in Unity. A GameObject can contain any number of components. Unity has a variety of built-in components, and unique components can be created by writing scripts that inherit commands from MonoBehaviour. Basically, components are the nuts and bolts of objects and behaviors in a game.

---

[1] Unity Technologies, "Unity User Manual (2018.3 beta)," accessed November 13, 2018, https://docs.unity3d.com/2018.3/Documentation/Manual/.

[2] Merriam-Webster, "Stereoscopy," accessed November 13, 2018, https://www.merriam-webster.com/dictionary/stereoscopy.

[3] ChaosGroup, "Guide to VR: Understanding and creating pre-rendered Virtual Reality content in V-Ray," accessed November 13, 2018, https://static.chaosgroup.com/documents/assets/000/000/023/original/Guide-to-VR_Chaos-Group-Labs.pdf?1472326790.

[4] Ibid.

[5] Ibid.

A GameObject acts as a container for many different components. In computational architectural design tools, components can be compared to the “Properties” toolbar assigned to any object.

6. Prefabs: a reusable asset in Unity. The Unity Prefab system allows for the creation, configuration and organization of GameObjects with associated components, values and child GameObjects to be reused. A Prefab is a template from which to duplicate GameObjects and apply changes to all of them at once. In this manner, they act like “blocks” used in computer-aided architectural design tools.

7. Projects: created through any game engine exist as self-contained units.[1] Projects compile all the aspects of a game such as any content or code used and subsequently act as a directory of information.[2] In a similar fashion, architectural projects are comprised of models, elements and drawings all compiled together to form the entire design.

8. Assets: represent any data item that is used in a game engine’s projects. Assets are organized directly as a self-contained unit and correspond to a filing structure located on a local disk. Assets can come from files outside of a game engine, such as a 3D model, an audio file, an image, or any of the other types of file that the engine supports. Assets also represent the hierarchy of project management in game engines, acting as the hub for all imports and links within the project file. Hence, they act similarly to the link manager that is commonly found in architectural computational tools.

Parameters of Experimentation

Software: 3DsMax, Unity 3D (Game Engine), SteamVR, iVRY

Hardware: PC, ETVR 4.0 Headset with Stereo Headphones (Mobile Headset), iPhone

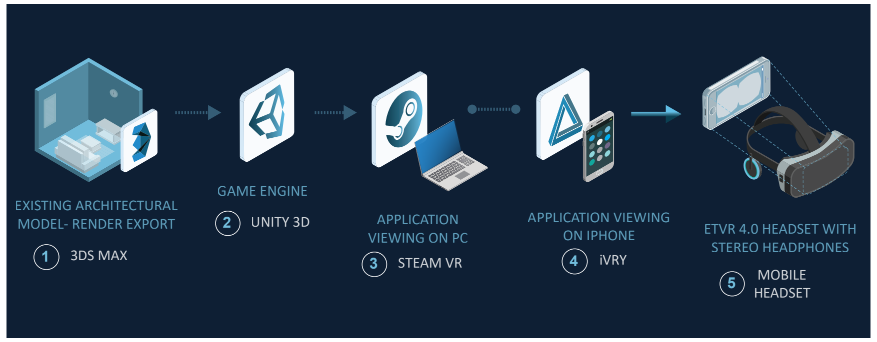


figure 5.04 Workflow: Render to VR

---

[1] Epic Games Inc., “Unreal Engine 4 Terminology: Get Started with UE4,” accessed January 12, 2019, https://docs.unrealengine.com/en-us/GettingStarted/Terminology.

[2] Ibid.

Process & Documentation of Creating a Scene in Unity

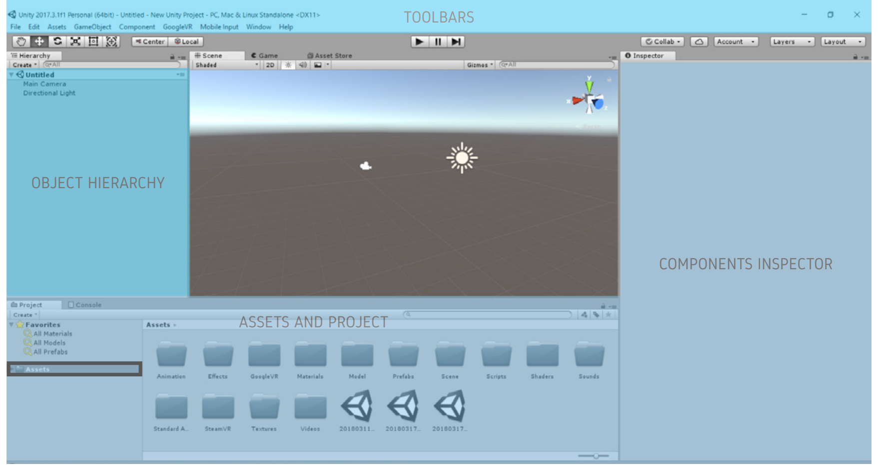


figure 5.05 Software Navigation in Unity

Step 1 [10 hr duration]: Software Setup and Navigation

The goal was to develop a VR scene for deployment onto a personal iPhone device. With the intent to deploy a stereoscopic scene on a mid-range mobile headset, workflow research had to be conducted. The first step involved researching the order of action for mobile experiences and a familiarization of online tutorials to allow for scene creation to be administered on a mobile device. This included online tutorial research[1] on navigating the Unity Interface and common project set-up compilations. Other empirical research conducted during this stage is that of stereoscopy and the creation of stereoscopic rendered images. Creating asset folders, practicing scene navigation and exploring the interface. As such, installing Unity with no base knowledge assumes a greater focus on learning while working with the software. It was in the best interest of maximizing the speed of personal learning to combine tutorials, empirical research and hands-on exploration.

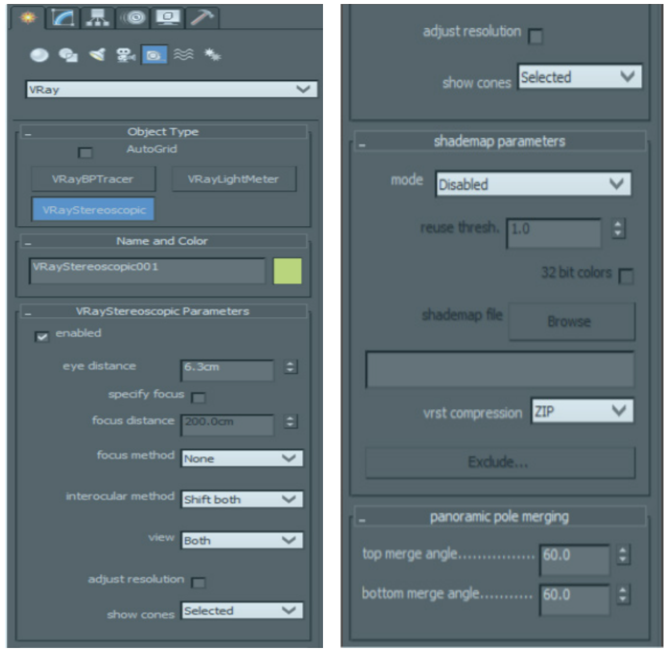
Step 2 [2 hr duration]: Stereoscopic Image Export

VR enables a viewer to experience their environment in true scale. Therefore, proper scale within a render export is crucial to the VR experience successfully simulating the illusion of reality. To allow a user to also experience what it’s like to be inside their virtual environment, scale is an essential factor to making that experience feel correct. Hence, the camera has to be placed at a correct height

---

[1] Lynda.com, “Unity for Architecture and Visualization Courses,” accessed March 20, 2018, https://www.lynda.com/.

in export, and the interocular distance (IOD) must be accurately set depending on the viewer. Both of these features determine the simulation's scale in relation to the user. References from anthropometric studies conducted in the US suggest a seated height average of 117 cm and standing height average of 157 cm[1]. Due to this evidence, studies indicate that a good interocular distance setting was determined to be set at 6.3 cm. Stereoscopic helpers and attributes are available as add-on software in many common architectural visualization programs such as 3dsMax and Maya. From these render settings, the cube 6x1 selection was made. From the resolution settings, the highest resolution of 9216 x 1536 was selected to achieve the final stereoscopic resolution of 18432 x 1536. Lastly, before rendering the image, under the image sampler, all filters were turned off, to produce a raw image output. Before exporting the render, camera positioning was adjusted one last time to make sure all aspects regarding a user's view were considered. This included verifying the front-facing position of the camera and deciding whether or not to create a seated or standing experience. By finalizing the camera position accordingly, the render was exported.



VRay

VRayStereoscopic001

☑ enabled
eye distance 6.3cm
specify focus ☐
focus distance 200.0cm
focus method None
interocular method Shift both
view Both
adjust resolution ☐
show cones Selected

mode Disabled
reuse thresh. 1.0
32 bit colors ☐
vrst compression ZIP
top merge angle............... 60.0
bottom merge angle.......... 60.0

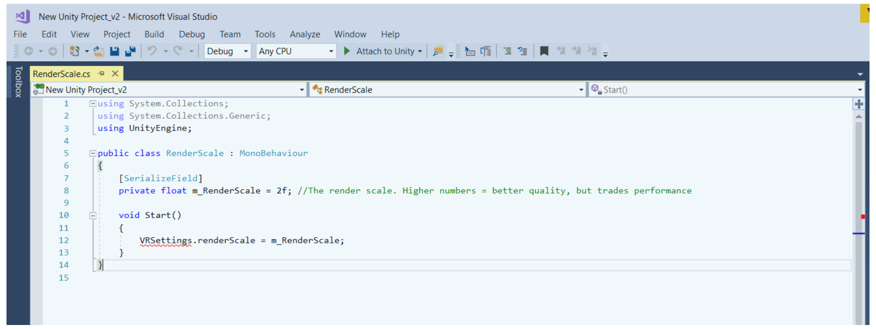
figure 5.06 Stereoscopic Image Export Settings in 3ds Max

---

[1] ChaosGroup, "Guide to VR: Understanding and creating pre-rendered Virtual Reality content in V-Ray," accessed November 13, 2018, https://static.chaosgroup.com/documents/assets/000/000/023/original/Guide-to-VR_Chaos-Group-Labs.pdf?1472326790.

Step 3 [1 hr duration]: Managing Image Resolution

As images are brought into Unity, they are down-sampled. Through conducting an online search, a script that altered the resolution was retrieved to heighten the render scale of the stereoscopic panorama. This script affected a property known as "Render Scale". After extensive efforts to make the script functional in MonoDevelop, the code used was eventually deemed unsuccessful. Error messages indicating the use of improper code were generated. Although troubleshooting was attempted, ultimately a decision was made to abandon the task and keep the panorama at the down-sampled resolution.



```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class RenderScale : MonoBehaviour
{
    [SerializeField]
    private float m_RenderScale = 2f; //The render scale. Higher numbers = better quality, but trades performance

    void Start()
    {
        VRSettings.renderScale = m_RenderScale;
    }
}
```

figure 5.07 Scripting Render Scale in MonoDevelop

Step 4 [2 hr duration]: Unity Scene Creation and Image Association

The first part of this process involved removing the default cameras and lights and downloading an asset utilities package from Oculus. While VR production is supported in Unity, this package allows access to a created camera that can associate images to each eye individually (as opposed to the ones prevalent in Unity which associate cameras to both eyes). The selective production of what is shown to each eye provides the effect of stereoscopy. The portion of this package used is the Oculus Virtual Reality (OVR) camera rig; it was added onto the stereo renders imported as assets in Unity (one for each eye). Once the eye associations were made for each eye and in the Cubemap format, the scene was deemed ready for final modifications.

Step 5 [1 hr duration]: Export for iOS

"Building the scene" is Unity's terminology behind exporting. This step was conducted to convert the scene into the desired application platform. This process also involved rendering the scene in real time, to accommodate the export medium. Ideally, the project's first step would have been to set up the build for iOS,

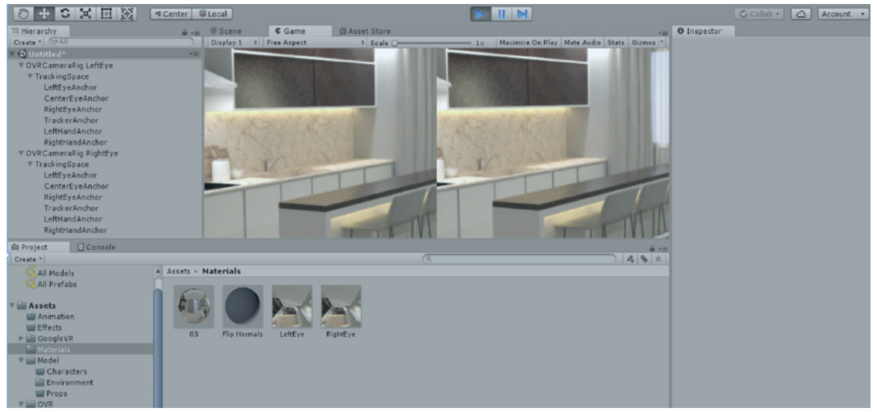


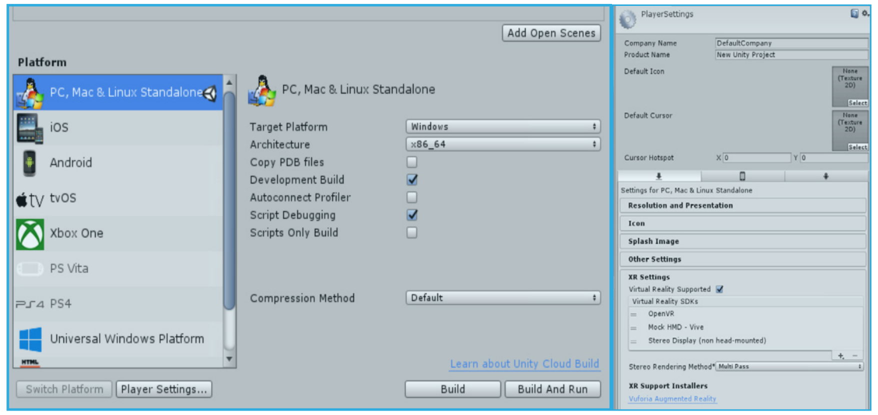
figure 5.08 Demo of Stereoscopy in Unity's Game Mode

figure 5.09 Image Warping According to Mouse Position on Screen Demo in Unity

as it builds for the default while the scene is being constructed by the user. The conversion process takes an additional amount of time for the PC to process if it is assigned at the end of developing the scene. Upon building the scene for iOS, it was determined the application required XCode to be run on an iOS device.

Step 6 [2 hr duration]: Export for PC

Upon resolving "compile errors" during the previous step, the application was built and run for PC settings as a test. A PC-based viewing platform for game applications was required for opening and viewing the game on a PC without having to keep Unity open constantly. SteamVR, an application commonly used for PC gaming and the launching of downloaded games for use, was discovered and downloaded. XCode is the paid platform that Mac-based content creators purchase to run VR games on iPhones. In searching for a feasible, accessible (non subscription-based) and PC option, the driver for an application called iVRY was installed. Using iVRY allows the Steam-run application on the desktop to be translated into the iPhone app for mobile viewing.



PC, Mac & Linux Standalone

Target Platform: Windows
Architecture: x86_64
Copy PDB files ☐
Development Build ☑
Autoconnect Profiler ☐
Script Debugging ☑
Scripts Only Build ☐
Compression Method: Default

Company Name: DefaultCompany
Product Name: New Unity Project
Default Icon: None (Texture 2D)
Default Cursor: None (Texture 2D)
Cursor Hotspot: X 0 Y 0
Settings for PC, Mac & Linux Standalone
XR Settings
Virtual Reality Supported ☑
Virtual Reality SDKs
OpenVR
Mock HMD - Vive
Stereo Display (non head-mounted)
Stereo Rendering Method*: Multi Pass

figure 5.10 Build Settings in Unity

figure 5.11 Full View Composed from Each Eye

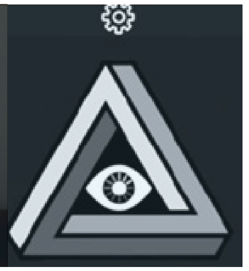


figure 5.12 iVRY Logo Loading Phone App

Table 4 Efficiency Documentation

TASK	DATE	START TIME	BREAK TIME	FINISH TIME	TOTAL DURATION
1. Software Setup and Navigation	2018-03-16	6:00 pm	2 hours	12:00 pm	4 hours
	2018-03-17	12:30 pm	2.5 hours	9:00 pm	6 hours
2. Stereoscopic Image Export	2018-03-18	12:00 pm	0.5 hours	2:30 pm	2 hours
3. Managing Image Resolution	2018-03-19	7:45 pm	0 hours	8:45 pm	1 hour
4. Unity Scene Creation and Image Association	2018-03-24	12:15 pm	2 hours	4:15 pm	2 hours
5. Export for iOS	2018-03-25	9:00 pm	2 hours	12:00 pm	1 hour
6. Export for PC	2018-03-26	10:30 pm	2 hours	2:30 am	2 hours

Result

This experiment yielded a stereoscopic scene of an interior space developed from Unity and demonstrated on an ETVR 4.0 Headset with Stereo Headphones housing a 6" iPhone. The evaluation of the practicality overall in the creation of this VR experience will derive from weighing efficiency (speed & duration of scene creation) against efficacy (in creating a compelling stereoscopic client/presentation friendly application).

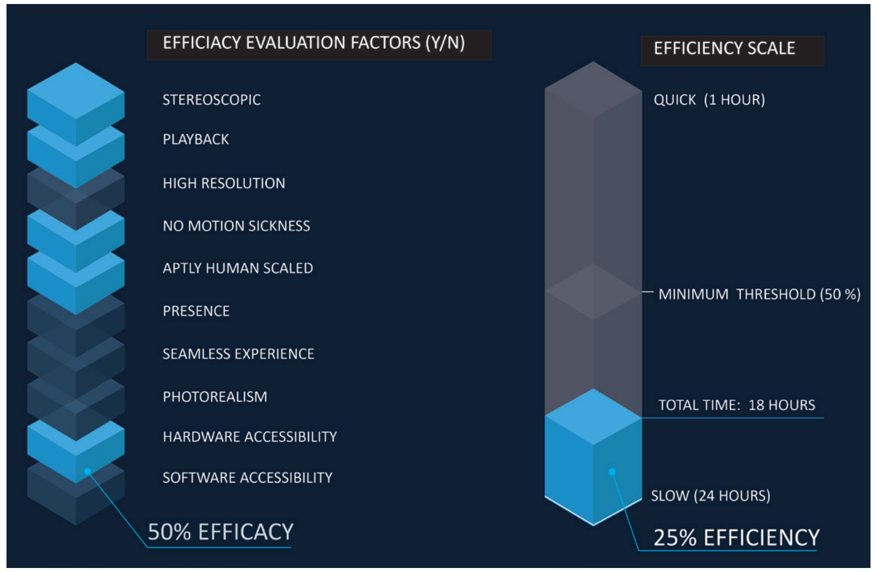


figure 5.13 Stereoscopic VR Scene: Practicality

Evaluation of Process

EFFICIENCY:

The experiment yielded a documented result of 18 hours of work. The majority of effort put into this task was based on initial research and troubleshooting problems. Although the most relatively accessible and entry-level hardware/software options were selected (as defined in Chapter 5), a seamless workflow was difficult to achieve. Consequently, research and tutorials were integral to the completion of the entire process. The "learning curve" of new software also became a significant consideration in adding time to the initial research phase. This is simply due to Unity being used with no prior knowledge of navigating 3D game engines. As the evaluation of Unity's ease of use implements existing architectural software to output a stereoscopic render, the process thereon seemed half complete. Coming into Unity with predictions of the simplicity of the task and merely a background in architectural visualization software was misinformed. This is demonstrated by the fact that learning the process of creating this scene took a generous amount of time. The initial research phase also delivered mass amounts of information from guides and tutorials that were not useful to the production of this scene. Substantial amounts of copied code, folders, empty assets (with nested files that were not deleted for fear of importance) and file structures in the working project file resulted from this. Troubleshooting also played a significant role in increasing time as problems (in the form of error messages) arose quickly with simple changes made.

This experiential learning process indicated that many things can go wrong easily within the setup of the scene. Missing a small step from a tutorial or forgetting to switch a simple parameter as the result of another, made every step integral to the success of application creation.

There were also numerous programs tested to streamline the process of export from Unity to the iPhone. In testing these different subsidiary programs, to ultimately achieve running the app on the phone, additional time was added. Towards future considerations of efficiency, as new VR apps are released, the process of conversion and understanding the methodology behind exporting the application becomes all the more complex. This factor is highlighted as more apps were downloaded and tested to end up being deleted, and attempts to bypass Apple's app licensing and XCode usage increased. Overall, research in tutorials, getting an understanding of the software and troubleshooting problems increased the duration of this experiment. The stereoscopic scene was therefore designed in an unrealistic amount of time, which might be generally unaffordable to the common fast-paced architectural environment. Due to the fact that a stereoscopic experiment can be compared to a render, it might be more practical in the design profession to create multiple renders in same timeframe.

EFFICACY:

A stereoscopic image is fixed and allows for limited interaction, restricting its usage to the final design stage. The primary use for a fully envisioned render in stereoscopy is in obtaining client buy-in for a compelling, expensive and realistic design. Thus, presenting an environment in stereoscopy might not be valuable unless it provides a generous resolution. Any VR experience that is facilitated as an architectural visualization tool must build upon the high-resolution, photorealistic baseline that the industry has set for imagery and animation.[1] Compromising on visual quality in the achievement of the aforementioned presence and scale would bode an unsuccessful and therefore, unfruitful visualization experience. The efficacy of building this experience was challenged at every stage a conversion was completed. Notably, conversions cause the overall image quality in stereoscopy to suffer. Firstly, in the start of scene creation, the build to the desired platform must be completed. As this task was done in the end (in converting from the iOS build to the PC build), assets along with the overall file size expanded and the conversion down-sampled nested files. Inherently, this showcased the difficulty of having to converting back and forth between platforms. A reason behind this can be attributed to the fact that an Unity can not build directly to an iPhone without a Mac since XCode is required.

---

[1] Barry Dineen, "Stereoscopic Renders in Unity3D for GearVR," accessed November 19, 2018, https://vrandarchitecture.com/2016/07/19/stereoscopic-renders-in-unity3d-for-gearvr/.

Due to this, a third-party application was necessary. Looking into third-party software to build to the PC, connect the iPhone, and have similar drivers installed on both PC and the iPhone to play the app on the iPhone, is a complex task. Eventually, with the discovery of iVRY, the ability to play the app on the iPhone directly presented itself. However, the resolution quality was noticeably lowered once again as it had to be converted through iVRY's own platform. While VR is developing quickly and new applications for execution are being released, efficiency of completing this task is hindered as anything "custom" or "novel" is attempted. The efficacy of the process of exporting from PC to Android is much simpler than that of exporting from PC to iPhone. Although iPhones themselves carry high resolutions, the Unity platform itself is unable to build directly to iOS without supplementing third party software, which unfortunately depletes the resolution at another stage from the original render output. Overall, the loss in resolution quality presents a major issue in efficacy of stereoscopic render creation ease and accessibility.

Limitations & Current Issues

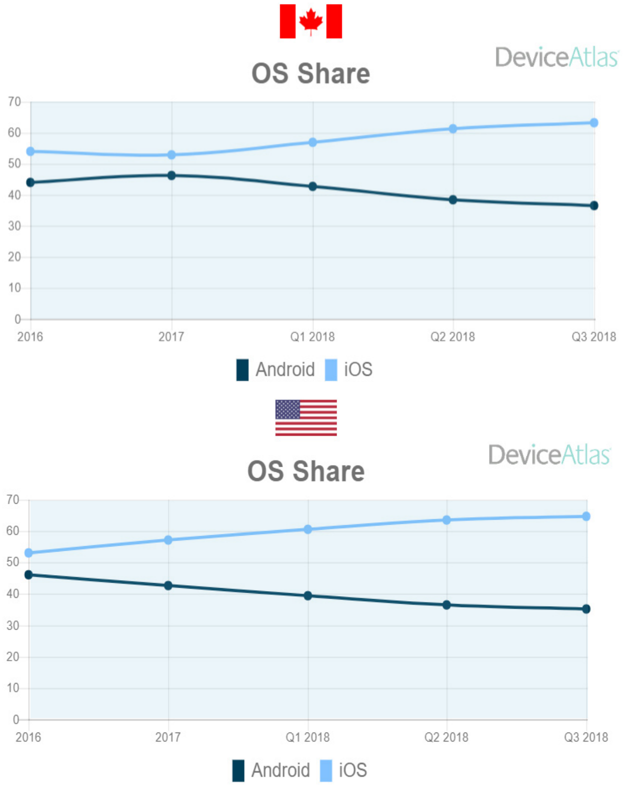


figure 5.14 Android vs. iOS Market Share

The export process is the greatest current limitation within this experiment to determine the ease and accessibility of VR app creation. The platforms build exclusively through their own systems (i.e. Mac to iPhone/ PC to Android) increasing the difficulty and inaccessibility of the entire process. iPhones currently carry greater market shares than their Android counterparts in North America.[1] This statistic indicates that iPhones are the more popular smartphone. Therefore, a significant flaw exists with application creation from game engines if building to iPhones directly continues to pose challenges. Additionally, as explained in Chapter 3, the future is promising towards mobile headsets, increasing the urgency and importance of iOS connectivity.

Technical Conclusions

1. The learning curve is steeper than expected. The process of navigating game engines proved to be unlike that of common architectural software, presenting unforseen challenges and requiring an understanding of diverse technical principles. Assumptions in the ease of use with an architectural background were dismissed as game engines operate with a specific step-by-step course of action. Architectural software, on the other hand, derives from an iterative and change-based approach.

2. There is a need to be able to render scenes at higher resolutions. If scenes are able to be created with lossless resolution, the effort put into the complex workflow is justified. However, when the process of conversion continually results in a slightly lower resolution, these decreases accumulate to display an overall substantial loss in resolution.

3. It is almost impossible to lock the viewer into the scene with the hardware at it's current state. If the user's head mistakenly turns all the way to the right or left, a shift in the virtual perspective results. Hence, the fragile notion of presence is then hindered as the user becomes aware of their artificial environment.

4. Multiple steps and software platforms were required to facilitate a functional workflow from render to smartphone. Each phase of the experiment carries a different program to execute a specific task. The programs used were as follows; exporting a stereoscopic render (3dsMax), utilizing package imports (Oculus developer aids), building the scene with a game engine (Unity3D), playing the application on PC (Steam) and finally activating smartphone viewing (iVRY). For this process to be hastened, tasks from the list need to be combined and completed by single programs.

---

[1] J. Kielty, "Android v iOS market share 2018," accessed November 19, 2018, https://deviceatlas.com/blog/android-v-ios-market-share.

5. The possibilities for playback on a wide range of devices increases the complexity of project to device translation. After the project is built from Unity, there are still steps to be completed before the device is ready for viewing the scene. The transition of an app from Unity to a phone can be simplified with further developments in Unity's build settings.

Conceptual Conclusions & Revisiting Research Q1

VR is a way of experiencing a 3D environment that, when done correctly, allows the brain to recognize embodiment and activate feeling present in a scene. Creating content that feeds left and right spherical images to each eye is a compelling way to depict a visualization. If done incorrectly, the environment brings no value to a viewer anymore than looking at a simple image. Thus, a stereoscopic VR scene is only a worthwhile achievement if the application is visually engaging. A compelling VR experiment is much more difficult to achieve than what was presumed prior to conducting this experiment. This is proven by both evaluation factors, namely efficiency and efficacy, resulting in unfavourable outcomes from this experiment. Prior to conducting the experiment, the timeframe range set for this experiment was 1 hr (deemed short) until 1 day (deemed long). With the result of an 18-hour duration, a stereoscopic scene took the equivalent of a little over 2 workdays (8 hours each) to complete. In the architectural practice, a task of this measure would presumably be conducted in a shorter amount of time. As the lengthy duration is indicative of the extensive necessary effort, this experiment can be concluded as difficult. In addition, the efficacy determining factors resulted in a negative effect upon image resolution. In an ideal scenario, the scene would yield exclusively positive responses from the efficacy factors. Yet, as the efficacy of the scene was compromised due to the iPhone playback complexity, this experiment can represent the challenges of inaccessibility. Hence, this experiment answers the initial research question of "Is a Unity VR application simple to create and easily accessible?" unfavourably.

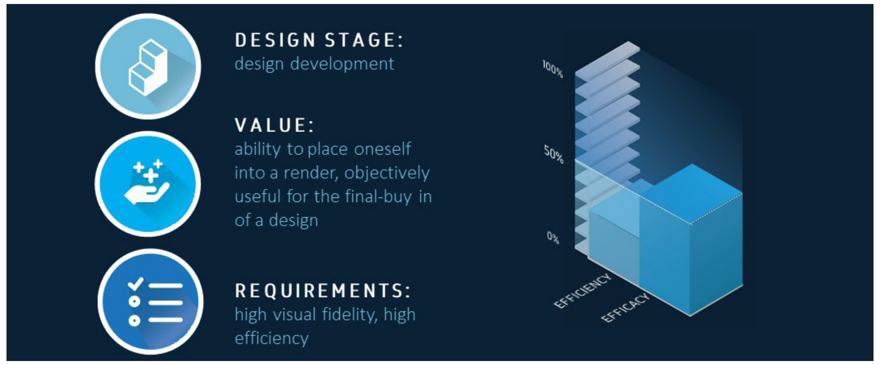


figure 5.15 Stereoscopic VR Design: Applications

Chapter 6: Narrative VR Journey Creation

Research Q2: Are narrated walkthrough VR demonstrations simple to create in Unity3D?

The focus of experimentation within this second investigation will be in determining the ease of creating an animated walkthrough in virtual reality with Unity3D[1]. Walkthroughs and narration are both essential and valuable factors in design communication. An architectural walkthrough is found to be superior tool to static 3D renderings when it comes to presenting projects (due to the increased capability of activating presence within a design) making this a reasonable second investigation.[2] Adding narration to a walkthrough might propel an understanding of the rationale behind a design. VR is a free environment wherein a user can conduct themselves as they please within the virtual space. When a narrative is added to the VR environment, a user can be guided to complete certain actions and focus on the context of their surroundings. Aside from pure aesthetics, most individuals are not trained to interpret nor understand any architectural design on their own without the knowledge of design principles executed within the scheme.[3] Thus, narration in a VR experience will essentially supplement the virtual experience with an auditory one.[4] This disengages a user from the distractions within their virtual environment and focuses their attention on the design elements they are meant to prioritize within a project. Lastly, animations in walkthroughs offer a more comprehensive interactive visualization experience for designs before they are built. With a more comprehensive interactive visualization experience, speculative fears ahead of a project for a design team can be alleviated.[5] They require the ability to design and coordinate object-to-user interaction as part of a timeline. Animations also require an understanding of programming and the principles of physics to facilitate creation in game engines, which are also explored in this chapter. Therefore, the objective of this experiment is to create a functioning animated narrative walkthrough scene with the Unity game engine from existing architectural models and provide a room-scale interactive VR experience for a user.

---

[1] ArchiCGI, "Architectural Walkthrough: How to Sell Your Project Without Words," accessed January 2, 2019, https://archicgi.com/architectural-walkthrough-for-selling-projects/.

[2] Erica Brett, "Architects in the Design of Virtual Reality Spaces" (UC Berkeley College of Environmental Design, 2015-2016), accessed October 10, 2018, http://ced.berkeley.edu/downloads/thesis/arch/2016/Brett_Erica_Architects_in_the_Design_of_Virsua__Reality_thesis.pdf.

[3] Barry Dineen, "Collaborative VR Journeys: A new way to experience unbuilt architecture!," accessed October 20, 2018, https://vrandarchitecture.com/2017/12/13/collaborative-vr-journeys-a-new-way-to-experience-unbuilt-architecture/.

[4] Paul Mealy, Virtual & augmented reality for dummies, 1st edition (Indianapolis IN: John Wiley and Sons, 2018).

[5] ArchiCGI, "Architectural Walkthrough: How to Sell Your Project Without Words," accessed January 2, 2019, https://archicgi.com/architectural-walkthrough-for-selling-projects/.

Ease will be determined by converting a model file from existing architectural software (i.e. Revit, SketchUp, 3dsMax) to an fbx file, importing the file into Unity3D to animate the virtual environment. To determine an objective account of difficulty, this experiment is split into two parts, which test the implementation of these features in VR in two different projects of differing scales. Part A deals with working from Revit into Unity by using a small-scale loft project and Part B deals with working from SketchUp into Unity, by using a well-documented and highly resourced large-scale residential project. The implementation of both parts will allow for a greater comprehension in architectural modeling software compatibility and will therefore attempt an unbiased review of architectural software integration. The process of using these two architectural modeling mediums into Unity will also provide critical knowledge of workflow management.

figure 6.01 Animated Walkthrough Scene Creation Process

Definitions[1]

1. Animations: movements created through the principles of physics stored as clips for objects/a hierarchy of objects in a scene. Animations are created as sequences of events are played and organized within a game engine's timeline.[2]

---

[1] Unity Technologies, "Unity User Manual (2018.3 beta)," accessed November 13, 2018, https://docs.unity3d.com/2018.3/Documentation/Manual/.

[2] Ibid..

Animations use colliders to define the physical space an object takes up and define the interaction between objects with actions. Rigid bodies are objects within a scene that activate animations to move and interact with other objects are called rigid bodies. Animations are created by rigid body components which, when acting as mass is subjected to the physical principles of drag and gravity trigger kinematics.[1] Animations thus work through understanding the theory of relativity which asserts that objects move relative to each other. Hence, no object within a game engine scene is ever at absolute rest or absolute motion.

2. Rendering[2]: a process that can be described as image synthesis as it is the automatic process of generating a photorealistic or un-photorealistic image from a 2D or 3D model (or models in what collectively are called a scene file) by means of a computer program. In this way, real-time rendering refers to animations rendered at high speeds to mimic the appearance of being generated in absolute real-time. It involves three stages; application, geometry and rasterization. The end result of real-time renderings is an animation rendered by the graphics system processing image frames quickly enough to show realistic motion. Due to this, real time rendering is measured in frames per second.

3. Lightmap UVs[3]: control how simulated lighting hits different objects, by reading it's surface and extrapolating a texture to it. Lightmap UVs process the surfaces of objects and pre-calculate lighting for them, saving those textures as results of "light baking".

4. Baked Lighting[4]: is the first of two times that a game engine processes lighting. Light Baking occurs during scene creation when an object turns static, which notifies the game engine of lighting staying in place. At runtime, the engine loads the lightmaps and applies their pre-calculated illumination back to the objects in the level, instead of having to calculate the lighting in real time every frame of the engine's main loop. Baked lighting is thus used more conventionally than dynamic lighting, so as not to compromise the performance of a scene. Baked lighting can, however, include processes that are too intricate to complete during run-time.

5. Dynamic Lighting[5]: the second of two times a game engine processes lighting. Dynamic Lighting is used exclusively for objects or lights that are in motion.

---

[1] Jonathan Linowes, Unity Virtual Reality Projects (Birmingham, England: Packt Publishing, 2015).

[2] Easy Render, "What is Real Time Rendering and Why It Matters," accessed November 8, 2018, https://www.easyrender.com/3d-rendering/what-is-real-time-rendering-and-why-it-matters.

[3] Unity Technologies, "Unity User Manual (2018.3 beta)," accessed November 13, 2018, https://docs.unity3d.com/2018.3/Documentation/Manual/.

[4] RealityTechnologies.com, "Virtual Reality: The Ultimate Guide to Understanding Virtual Reality (VR) Technology," accessed September 5, 2018, https://www.realitytechnologies.com/virtual-reality/.

[5] Ibid.

It calculates light for the moving object in real-time, while the engine processes the animation of that object during runtime. All calculations for dynamic lighting are calculated when the visualization is rendering or running.

6. Shaders[1]: tell the Unity Engine how to render a material and give the engine a set of material parameters.

7. Light baking[2]: the process of pre-calculating the lighting for the objects in a given level and saving it to textures on the local disc. These textures are called lightmaps.

8. Image maps[3]: represent how the texture of an image is visible on object with settings of smoothness, specularity, shine and reflectivity.

9. Normal maps[4]: represent the shape and texture of a surface with respect to how light bounces off of it. This effect is further created by height maps and occlusion maps which alter the emission of light emanating from any object in any scene.

10. Ambient occlusion[5]: is a property that adds small shadows to the corners of all objects, to give a scene depth and to increase the photo-realistic effect of objects by mimicking those in reality. This shading technique is created through the calculations made of how each vertex is exposed to lighting. For example, the interior of a tube is typically more occluded (and hence darker) than the exposed outer surfaces, and the deeper one travels inside the tube, the more occluded (and darker) the lighting becomes. It is, therefore, a value that is calculated for each surface point.

11. Final gar[6]: is a property that increases the precision and refinement of the light baking process within a game engine.

Parameters of Experimentation

Software: Revit, 3DsMax, SketchUp Pro, Unity 3D (Game Engine)

Hardware: PC, Oculus Rift (Tethered Headset)

---

[1] Unity Technologies, "Unity User Manual (2018.3 beta)," accessed November 13, 2018, https://docs.unity3d.com/2018.3/Documentation/Manual/.

[2] RealityTechnologies.com, "Virtual Reality: The Ultimate Guide to Understanding Virtual Reality (VR) Technology," accessed September 5, 2018, https://www.realitytechnologies.com/virtual-reality/.

[3] Ibid.

[4] Ibid.

[5] RealityTechnologies.com, "Virtual Reality: The Ultimate Guide to Understanding Virtual Reality (VR) Technology," accessed September 5, 2018, https://www.realitytechnologies.com/virtual-reality/.

[6] Ibid.

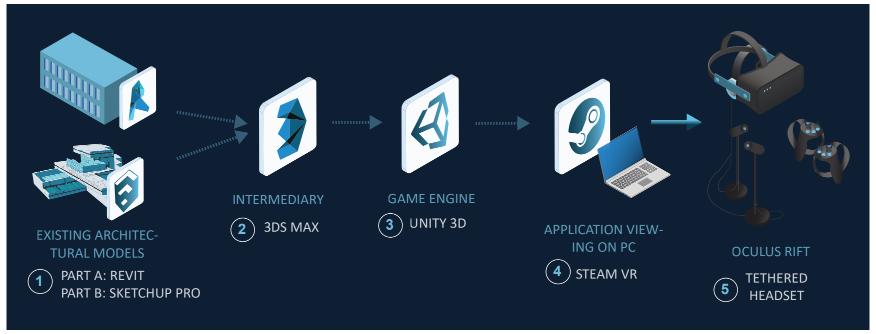


figure 6.02 Workflow: Animating for VR

Process & Documentation of Creating a Scene in Unity

Step 1 [6 hr duration]: Research and Tutorials

Similar to the previous chapter, tutorials and research were conducted in order to understand complex navigation within the Unity interface.[1] This involved understanding how a project is set up and exploring controls or component functionalities. In this stage, tutorials were accessed in order to understand functions, actions and capabilities beyond the basic interface of the program. This stage was conducted in order to understand the principles surrounding animation within Unity and the application of programming commands required to perform actions.

Part A: Revit to Unity

Step 2 [2 hr duration]: Export Preparation in Revit

The objective of this step was to simplify the model for import into Unity, thereby condensing information to improve application performance and run the Unity scene at a faster framerate. This step began with altering the existing Revit model for import into Unity. Any geometry that needs to be seen in the final model in Unity must be modeled in Revit. As Unity understands models as meshes, programs that work with modeling in other formats (such as nurbs, BIM etc.) need to be passed through an intermediary. Before bringing the model into the intermediary, the model had to be exported. Exports of 3D geometry from Revit are done through 3D views due to the model following settings within the 3D views. Thus, 3D views were set up for export within this stage. This entailed setting up the view as a template and splitting it into multiple views for export. Within the new 3D view created for export, any unnecessary elements of information that might not need to be present in the virtual scene were turned off using the cat-

---

[1] Lynda.com, "Unity for Architecture and Visualization Courses," accessed March 20, 2018, https://www.lynda.com/.

egory function of applying the temporary hide/isolate tool. Finally, the essential geometry was cropped using the section-box tool. Online tutorials suggested a method which involved separating 3D views exports into 3 categories; static (building), dynamic (doors and objects requiring animation) and furniture[1]. The static export contained simply the building with turned off furniture, systems, generic models, lighting and doors. The dynamic export included only the doors and the furniture included all interior modeled objects.

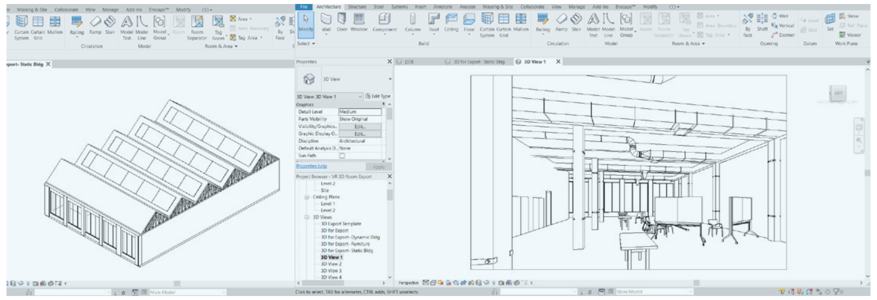


figure 6.03 Revit Project Setup Pre-Export

Step 3 [1 hr duration]: 3ds Max Import

Revit materials are not immediately translated into Unity by exporting a Revit file type into the common fbx format. Hence, to send a model into Unity effectively, 3ds Max needs to translate the model. The fbx file format is common for translating models from one source into another. 3ds Max was used in this scenario because it allows for file readability amongst architectural software (especially from the same vendor), allowing for greater functionality and access to an fbx file before it is brought into Unity and can read Revit materials and textures. Hence, a 3D view export of the 3 aforementioned views as fbx files was imported and exported from 3ds Max.

Step 4 [1 hr duration]: 3ds Max Fixes

The goal of this step was to translate information effectively from Revit into Unity. As 3ds Max is primarily a translation tool for this purpose, the Revit file was imported and linked in and then alterations were required for UV mapping and material textures so that these aspects could be read correctly by Unity. Depending on details of exports, UV mapping information is often lost in the translation of a file into a different format. UVs tell an engine how any textured image is displaced on the surface of a volume. To fix this in 3ds Max's modifier for the object, under UVW map, box mapping was selected. This process was done for each different view export created previously.

---

[1] Lynda.com, "Unity for Architecture and Visualization Courses," accessed March 20, 2018, https://www.lynda.com/.

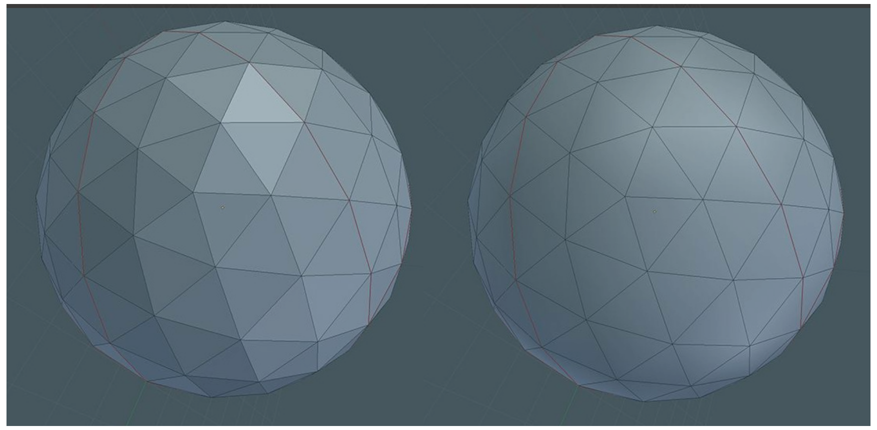


figure 6.04 Texture Stretching on Surfaces: UVW Map Diagram

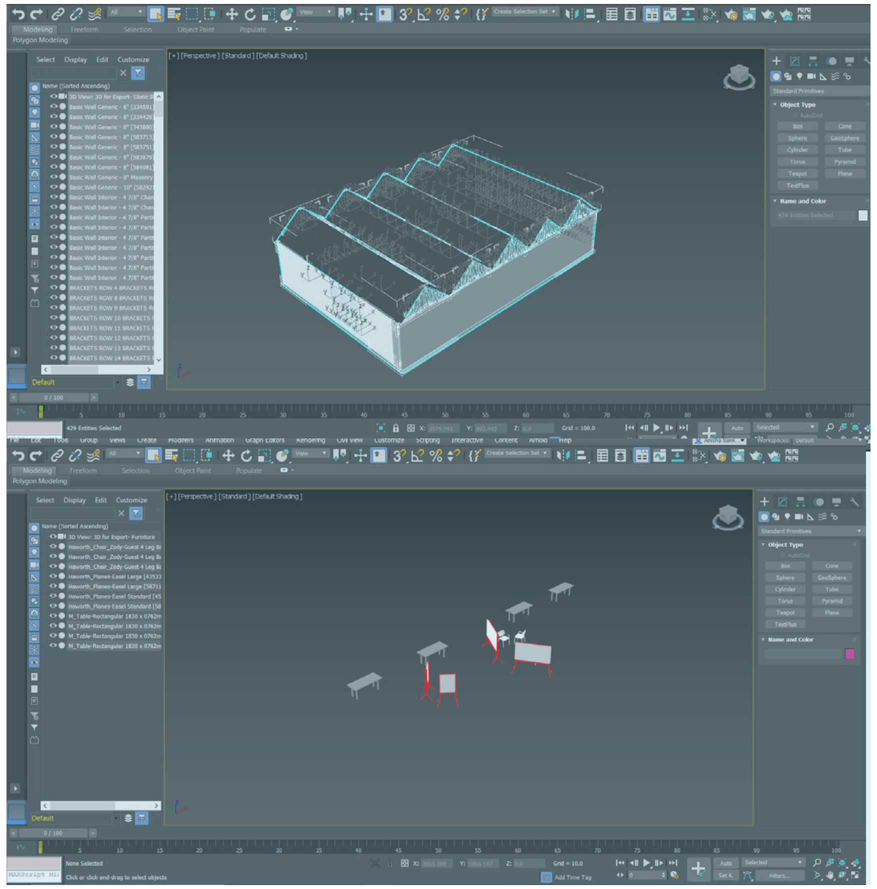


figure 6.05 3ds Max File Segregation

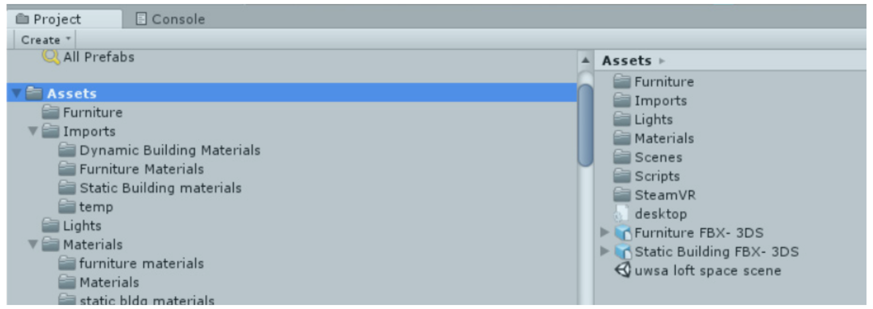


figure 6.06 Project Organization in Unity

Step 5 [3 hr duration]: Unity3D Project Creation

This step comprised of creating the project in Unity by using the 3D template from the given options. From the project tab, "import new asset" was selected. Importing files directly from 3ds Max allows a user to develop links to the original Revit file, which is useful during the design process if changes are made to the original Revit file. Once the model was brought in as an asset, the next step involved changing the scale factor. The scale factor was changed by a measure of 0.0254 since a Unity Unit is equivalent to 1 meter and the model was built using the measurement in inches in Revit. The next change made was to turn on "generate light map UVs" to make Unity calculate light map generation. The next step was to turn on mesh compression in Unity (to speed up components slowing down the speed of the typical application). After the model was imported, the materials section in the inspector tab was used to show embedded textures and materials from Revit and 3ds Max. Hence, these materials were organized into different folders according to each model imported.

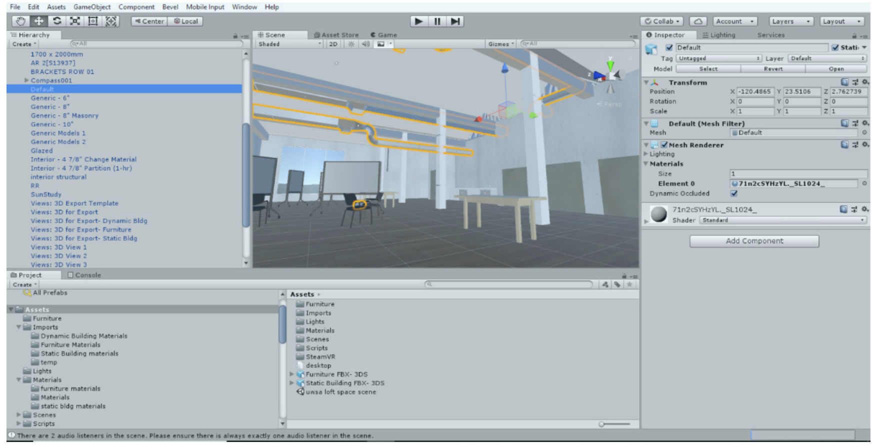


figure 6.07 Objects/Mesh Placement in Unity

Step 6 [4 hr duration]: Object Placement

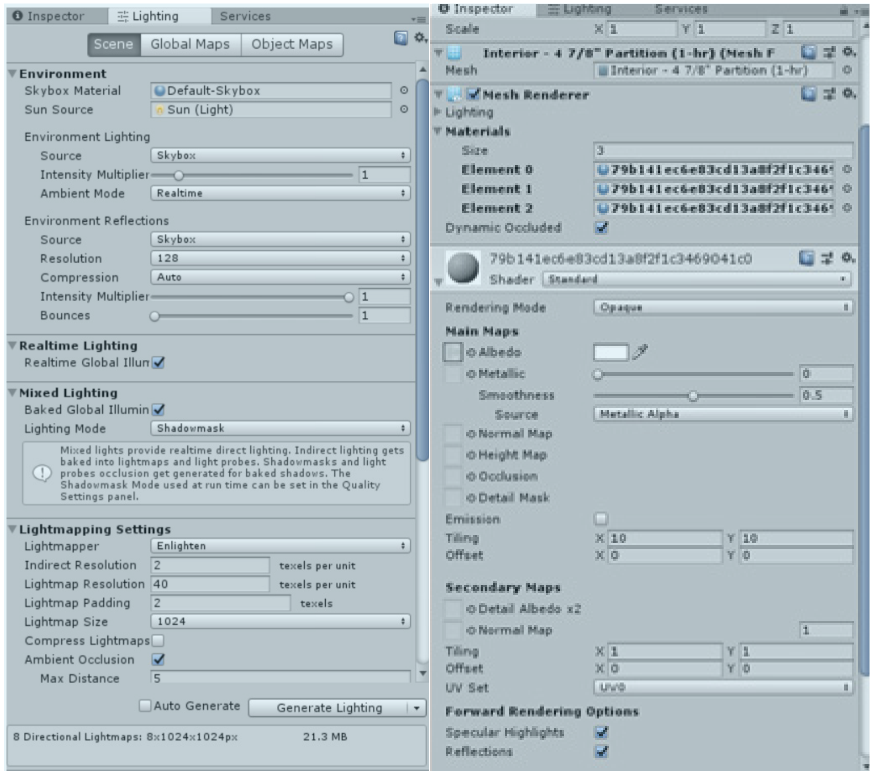
From the assets tab, objects were dragged into the hierarchy tab and converted into game objects. Initially when they are brought in, appeared in the 0,0,0 position on the coordinate grid. The next task was to ensure all imports were lining up with one another, checking scale and deleting unnecessary elements imported from objects after looking at them cohesively within the scene. Elements that provide no visible geometry to the scene were turned off rather than deleted so as not to damage the prefab link that existed between Revit and the Unity model. In this stage, a scene test was also done to check if an adequate framerate was achieved.

Step 7 [7 hr duration]: Materials and Prefabs Adjustments

After conducting search Internet queries for a script to convert objects into prefabs in Unity, a script was applied to ensure any change made to a single prefab would affect all others, without damaging the linked object from Revit. After prefabs within the model were adjusted, materials within prefabs were adjusted accordingly. Details were added to materials used in order to heighten their photorealistic qualities in Unity. Materials were sourced back from Revit's own extensive library. These material image files were then brought into the assets folder location. Displacement maps, ambient occlusion maps and normal maps were then switched on in Unity to have Unity process unique lighting conditions in real-time. Each material's tiling factor was then adjusted to find the optimal setting. Next, the tone of any image was altered to enhance either light or darkness within the specific material. Finally, the image resolutions were brought down to a maximum size of 512 so as not to increase the framerate of the application. Materials were dragged onto objects in the hierarchy to apply all the settings made, and to test out the material quality on each object. After the main tasks of this step were completed, some texture maps within the file were missing as image files. This resulted in the need to return to 3ds Max to open the file and apply the UVW mapping in box format again. This solved the problem and eventually brought the texture of the linked file back to visibility in Unity.

Step 8 [8 hr duration]: Lighting Adjustments

In this step, the objective was to achieve the desired aesthetic for lighting by exploring and manipulating the way a Unity scene interprets light. Lighting was first turned on through the window toolbar. From rendering in the past, it was understood that global or environmental lighting is the first way lighting can be affected within a Unity scene. The second way is through adding actual lights as objects in the environment. Hence, scene lighting was the first aspect adjusted.



Environment
Skybox Material: Default-Skybox
Sun Source: Sun (Light)
Environment Lighting
Source: Skybox
Intensity Multiplier: 1
Ambient Mode: Realtime
Environment Reflections
Source: Skybox
Resolution: 128
Compression: Auto
Intensity Multiplier: 1
Bounces: 1

Realtime Lighting
Realtime Global Illum ☑

Mixed Lighting
Baked Global Illumin ☑
Lighting Mode: Shadowmask

Mixed lights provide realtime direct lighting. Indirect lighting gets baked into lightmaps and light probes. Shadowmasks and light probes occlusion get generated for baked shadows. The Shadowmask Mode used at run time can be set in the Quality Settings panel.

Lightmapping Settings
Lightmapper: Enlighten
Indirect Resolution: 2 texels per unit
Lightmap Resolution: 40 texels per unit
Lightmap Padding: 2 texels
Lightmap Size: 1024
Compress Lightmaps ☐
Ambient Occlusion ☑
Max Distance: 5
☐ Auto Generate

8 Directional Lightmaps: 8x1024x1024px 21.3 MB

Scale X 1 Y 1 Z 1
Interior - 4 7/8" Partition (1-hr) (Mesh F
Mesh: Interior - 4 7/8" Partition (1-hr)
☑ **Mesh Renderer**
Lighting
Materials
Size: 3
Element 0: 79b141ec6e83cd13a8f2f1c346
Element 1: 79b141ec6e83cd13a8f2f1c346
Element 2: 79b141ec6e83cd13a8f2f1c346
Dynamic Occluded ☑

79b141ec6e83cd13a8f2f1c3469041c0
Shader: Standard
Rendering Mode: Opaque
Main Maps
Albedo
Metallic: 0
Smoothness: 0.5
Source: Metallic Alpha
Normal Map
Height Map
Occlusion
Detail Mask
Emission ☐
Tiling X 10 Y 10
Offset X 0 Y 0
Secondary Maps
Detail Albedo x2
Normal Map: 1
Tiling X 1 Y 1
Offset X 0 Y 0
UV Set: UV0
Forward Rendering Options
Specular Highlights ☑
Reflections ☑

figure 6.08 Lighting and Texture Adjustments in Unity

The sun icon was accessed through turning on scene lighting. Directional light was added into the scene and renamed as the sun, adding it to the sun source. Next, as scene lighting was understood to work from both global illumination and intensity, the environment simulation had to be modified by altering reflections and light map settings. Within these settings, specifically ambient occlusion and final gar were adjusted to simulate additional natural lighting and shadow effects in the scene. After, the debug settings were turned on to automatically rebuild lightmaps within the simulation. Next, to mimic where lighting fixtures would naturally be placed to emit light in a space, fixtures were added. Depending on the type of lighting fixture or lighting within the scene, light objects were added (not to every existing light fixture), and the parameter settings of those lights were adjusted. Point lights were mainly used for this task to light the scene well and achieve any necessary light within interior spaces. Since multiple point lights were used, a single point light was made into a prefab, duplicated and multiplied into the scene. This was achieved through creating a new folder for lights within the assets and dragging a point light into that folder. Then, that light was placed near the structural bays of the space. Within the inspector window, the render mode settings were turned automatic. The range of these lights were altered, and the colour was turned warmer. After the scene had been set to include point lights and overall dynamic sunlight, the lightmap static setting was checked off from the lighting section under mesh renderer. Finally, all geometry was set to static or dynamic by setting the floor to receive shadows and doors and furniture to dynamic.

Step 9 [6 hr duration]: Scene Interaction

The objective of this step was to develop a full first person walkthrough within the scene. Unity developers have a built in first person controller within the character sector of importable asset packages. The rigid-body first person controller was grabbed from the prefabs sector of the character folder. After being dragged into the scene the first person controller height was adjusted to achieve a reasonable eye level. The main camera was deleted with the addition of this controller due to the fact that two different cameras in the scene wouldn't allow the scene to function properly. The first-person controller will be interacting within the space.

To interact, the geometry within the space needed colliders, which act as barriers in a virtual simulation. For objects within the space, a mesh collider component was added to define the boundaries of where a user could travel. Movable objects instead used box colliders as they increase processing time and have a lower frame rate. Navigational audio was then added to the recording track to be played for the first person controller. Then the scene was checked to ensure multiple audio listeners or cameras weren't enabled, which would hinder the functionality of the audio recording.

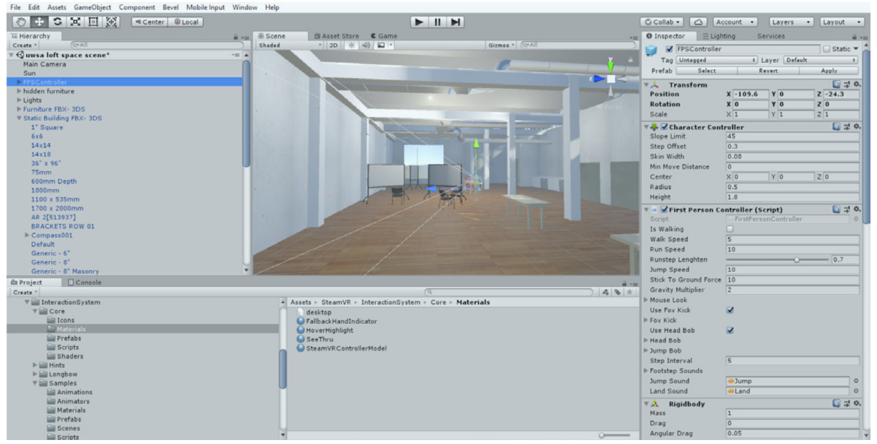


figure 6.09 First-person Controller Adjustments in Unity

Step 10 [4 hr duration]: Creating Teleportation

The first task to create teleportation was to download Oculus Utilities and the SteamVR package. It was also important to ensure the build settings were setup adequately. Under the player settings for the build, it was checked to ensure that the openVR SDK was toggled on. Next, the rigid body controller was replaced. The most crucial factor in the package is the camerarig, which contains the properties, script and functions of the camera to control it within the animation. Under steam's asset package for prefabs, the camerarig was dragged into the scene hierarchy and ordinated to sit on the floor surface. The animation was then played and deemed to require a tracked controller script to be assigned to each hand-held controller. After conducting an online search, a tracked controller script was retrieved for use in the scene. The Camerarig was opened and added to both handheld controllers brought in from the Oculus Utilities package. The maximize on play button was toggled off to view the controllers working within the scene. To finalize teleportation, SteamVR's teleportation prefab was placed in the scene and the steamVR player replaced the camerarig in the scene. The floor surface was finally duplicated to allow for teleportation. It was transformed slightly by being elevated. Lastly, a component was added to add colliders to the floor for teleportation destinations.

Step 11 [1 hr duration]: Build Settings

Using SteamVR for Unity and adjusting the player settings to have the Oculus SDK chosen allowed for the build settings to be set for the scene. After testing the scene using the play button, ambient occlusion and final gar were activated. Next, the scene was built for the PC and played using SteamVR for PC and an Oculus Rift headset.

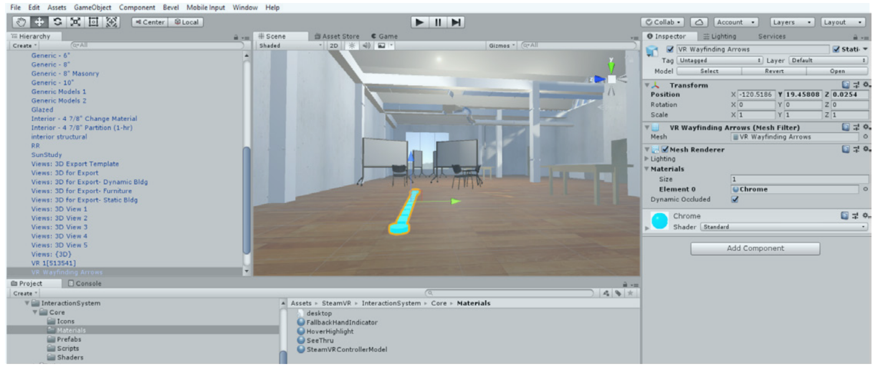


figure 6.10 Creation of Marker Arrows to Guide Movement in Unity

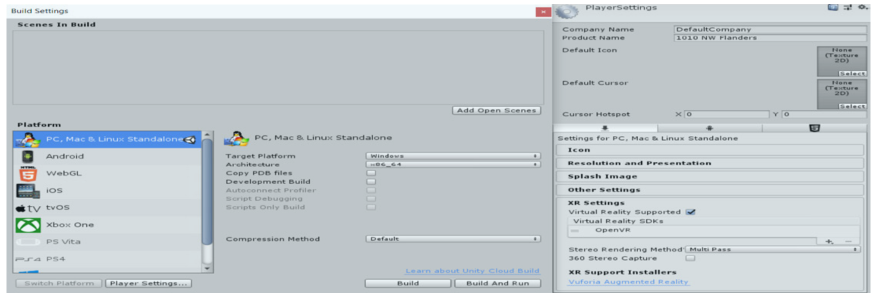


figure 6.11 Unity Build Settings

figure 6.12 Scene Navigation in VR

Part B: SketchUp to Unity

Step 2 [5 hr duration]: Export Preparation in SketchUp

The objective of this stage was to prepare the file for all the assets to be soon imported into Unity. It was discovered that only the pro version of SketchUp can generate the fbx output format needed for import to Unity, which might be a concern to ardent users of SketchUp Make. Additionally, from the research conducted toward this process, it was determined that the first step was to purge and delete unused or redundant objects. Anything that was hidden, a dimension, text or a 2D object was removed. Interior objects specific to rooms were relocated onto a new file so that their models can be brought in separately into Unity as needed. Any data that was not visible on screen and wouldn't be perceived as the visualization was walked through was removed. Through checking the model's statistics, the face count was reviewed, and unused faces were purged. After the model was verified to have no components, materials or groups that were not being used, the units and scaling was revised as meters are Unity's unit. Next, all objects within SketchUp were given descriptive names (to facilitate asset management within Unity).

This task included formatting all components and groups in the model. After, every face was examined from the styles function to determine whether the surface normals of the project were facing their appropriate direction. As every surface in the project has the potential of being assigned two potential materials (one for the front and one for the back), each material had to be corrected to be applied on the appropritate front face and process the surface normal. Lastly, the geometry required a deliberate origin point with a relationship to the coordinate axes that would make sense for Unity. SketchUp and Unity possess different orientations of geometry with respect to the coordinate grid. In SketchUp, the Z-axis is the vertical direction whereas in Unity, the Y-axis represents the vertical direction. Hence, the entire geometry was rotated with respect to the origin point and the model was exported into the FBX format with a unit set to meters.

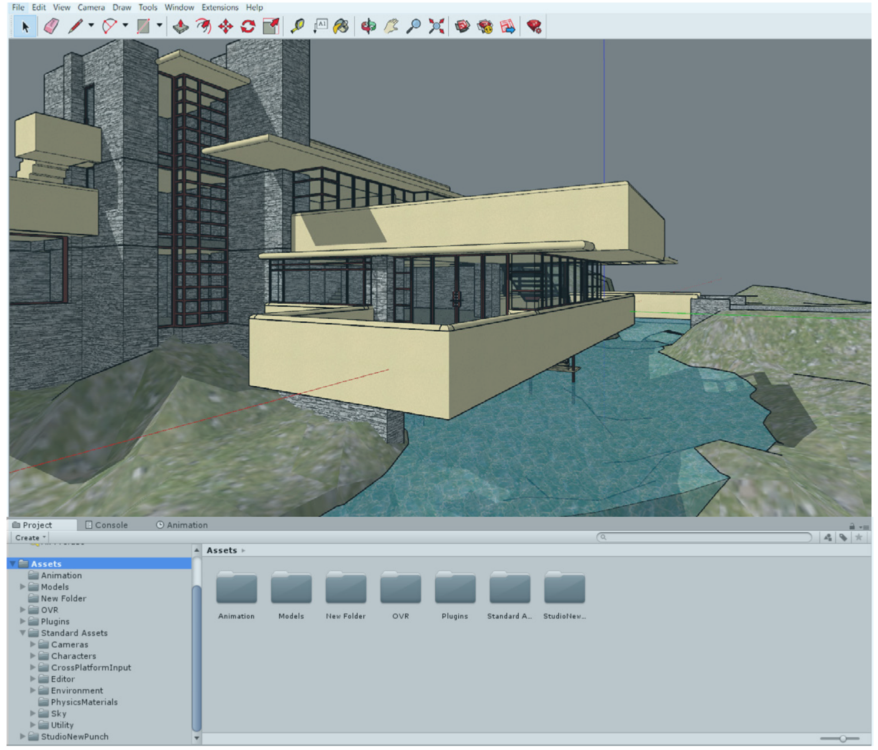


figure 6.13 SketchUp Pro Editing Mesh Editing, Export into Assets

Step 3 [2 hr duration]: Unity3D Project Creation

This step comprised of creating the project in Unity by using the 3D template from the given options. New project assets were imported into Unity as the prepared fbx file.

From the project tab, folders were created to organize the project by creating labels for all included objetcs in the final animation (such as lights, models, characters etc.). When importing the model, the texture map folder was also brought into Unity's assets. After evaluating the organization of assets, and ensuring objects are properly stored within subsequent folders, texture maps were assigned to their respective materials.

Step 4 [4 hr duration]: Scene Navigation

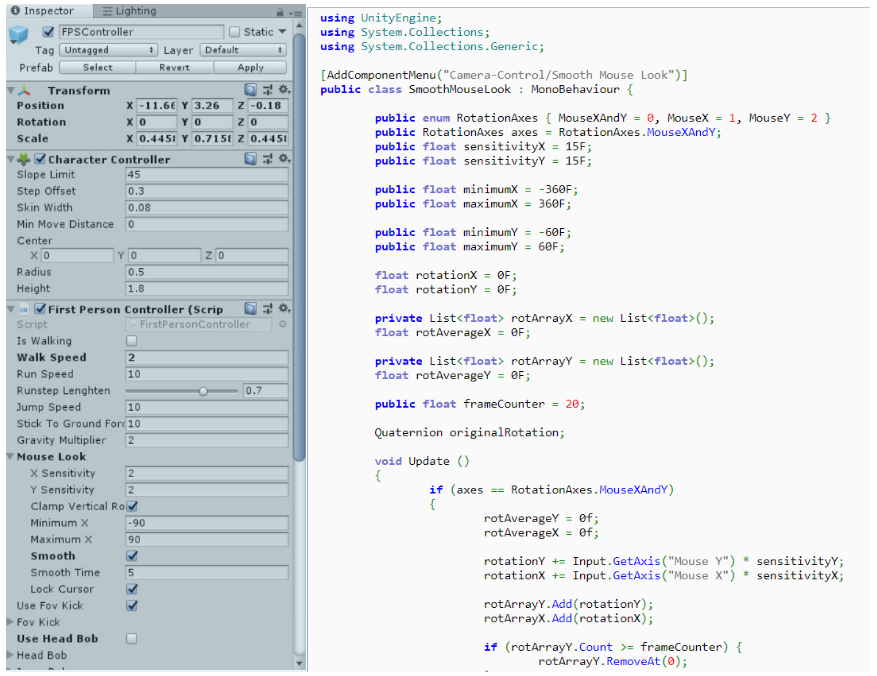
As the main model had been imported, the scene was ready to be animated. This process began with assigning a rigid body first-person character to the project, so the nested animation could be viewed while changes were made to the larger project as a whole. From the assets menu, the first person controller character package was imported. The assets package of the first person controller nested the "FirstPersonCharacter" prefab file which was dragged into the hierarchy to be activated in the scene. The sphere with the character's rigid body were positioned such that the prefab was above the model's floor surface. Colliders were generated for this mesh prefab and applied in order for objects and characters within the scene to have physical boundaries of movement. From the game tab, the scene was played in order to test the first person controller's walking speed, remove head bob, decelerate the walking speed gradually and enable an Audio Source. The audio source was then set to the narrational audio recording created to help a user navigate the scene.

Step 5 [4 hr duration]: First Person Controller Adjustments

As is, the rigid body didn't move as uniformly within the environment as expected. Therefore, a search for a smoothing camera view control was conducted. The "smoothmouselook" camera script was found and edited for the first person controller in MonoDevelop (Unity's C# scripting interface).[1] To ease the motion of going up stairs for the first person controller, an attempt was made to replace the stair collider with a sloped collider. Next, under the character controller's inspector, the slope limit was changed to 90 and the step offset was changed to 0.4 after various online searches through Google. Lastly, an invisible barrier was required to limit the character's motion from wandering into unwanted spaces in the simulation. Hence, a cube resembling a wide rectangular prism was created to set boundaries for the frist person controller's travel. A mesh collider property was associated to the cube before it was duplicated and moved as necessary.

---

[1] asteins, "SmoothMouseLook," accessed October 10, 2018, http://wiki.unity3d.com/index.php/SmoothMouseLook.



FPSController ☑ ☐ Static
Tag Untagged Layer Default
Prefab

Transform
Position X -11.66 Y 3.26 Z -0.18
Rotation X 0 Y 0 Z 0
Scale X 0.4451 Y 0.7158 Z 0.4451

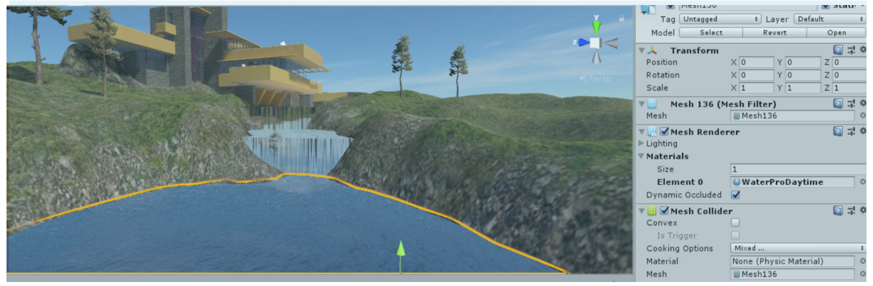
Character Controller ☑
Slope Limit 45
Step Offset 0.3
Skin Width 0.08
Min Move Distance 0
Center X 0 Y 0 Z 0
Radius 0.5
Height 1.8

First Person Controller (Scrip ☑
Script FirstPersonController
Is Walking ☐
Walk Speed 2
Run Speed 10
Runstep Lenghten 0.7
Jump Speed 10
Stick To Ground Forc 10
Gravity Multiplier 2
Mouse Look
X Sensitivity 2
Y Sensitivity 2
Clamp Vertical Ro ☑
Minimum X -90
Maximum X 90
Smooth ☑
Smooth Time 5
Lock Cursor ☑
Use Fov Kick ☑
Fov Kick
Use Head Bob ☐
Head Bob

```
using UnityEngine;
using System.Collections;
using System.Collections.Generic;

[AddComponentMenu("Camera-Control/Smooth Mouse Look")]
public class SmoothMouseLook : MonoBehaviour {

        public enum RotationAxes { MouseXAndY = 0, MouseX = 1, MouseY = 2 }
        public RotationAxes axes = RotationAxes.MouseXAndY;
        public float sensitivityX = 15F;
        public float sensitivityY = 15F;

        public float minimumX = -360F;
        public float maximumX = 360F;

        public float minimumY = -60F;
        public float maximumY = 60F;

        float rotationX = 0F;
        float rotationY = 0F;

        private List<float> rotArrayX = new List<float>();
        float rotAverageX = 0F;

        private List<float> rotArrayY = new List<float>();
        float rotAverageY = 0F;

        public float frameCounter = 20;

        Quaternion originalRotation;

        void Update ()
        {
                if (axes == RotationAxes.MouseXAndY)
                {
                        rotAverageY = 0f;
                        rotAverageX = 0f;

                        rotationY += Input.GetAxis("Mouse Y") * sensitivityY;
                        rotationX += Input.GetAxis("Mouse X") * sensitivityX;

                        rotArrayY.Add(rotationY);
                        rotArrayX.Add(rotationX);

                        if (rotArrayY.Count >= frameCounter) {
                                rotArrayY.RemoveAt(0);
```

figure 6.14 Scripting User Navigation and Walk Speed for Unity from MonoDevelop

> *Audio affixed to simulation: Welcome to a VR tour demo of the University of Waterloo School of Architecture loft space. As you slowly begin to familiarize yourself with walking through your environment, let's give you some essential historical information. Relocating from the University of Waterloo campus to the century-old Riverside Silk Mills in Cambridge Ontario, the new School of Architecture has repositioned itself as a model for the instruction of architecture, sustainable design and urban renewal. By design, the School of Architecture is a didactic model of building assembly. Since it is impossible to move through the School without passing through communal spaces, this loft continuously offers itself as a place of interaction and collaboration. Exposed connections and mechanical systems demonstrate their own utility and construction. Porous spaces frame views of things being made. A conscious decision "not to design too much" renders most surfaces raw and durable, suited to exhibition, intervention and creativity. If you take a look up and around you can notice all the piping and structural formations around you. This area of the school prides itself in being an open-concept-light filled environment. Can you feel it gleaming in?*



Tag Untagged Layer Default
Model

Transform
Position X 0 Y 0 Z 0
Rotation X 0 Y 0 Z 0
Scale X 1 Y 1 Z 1

Mesh 136 (Mesh Filter)
Mesh Mesh136

Mesh Renderer ☑
Lighting
Materials
Size 1
Element 0 WaterProDaytime
Dynamic Occluded ☑

Mesh Collider ☑
Convex ☐
Is Trigger ☐
Cooking Options Mixed ...
Material None (Physic Material)
Mesh Mesh136

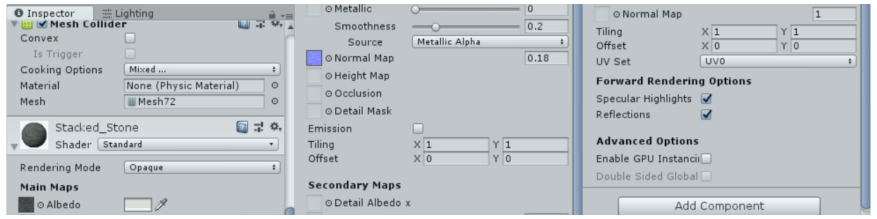
figure 6.15 Environment Texture Adjustments in Unity

Step 6 [7 hr duration]: Environment Adjustments

As the model is situated within a landscape that can be accessed by the first person controller, the sunlight, shadows, sky and terrain required modification. First, to mimic south-facing sunlight, a directional light was created and renamed. For the sun to build soft shadows, the edge's blur settings were set down to 75% and anti-aliasing was quadrupled from it's original setting level. To mimic the natural environment and sky, a custom skybox was downloaded from the Unity Asset store. The use of a skybox created a reduced intensity of lighting within the scene, and hence, the lighting intensity multiplier for the skybox was doubled. Lastly, to edit terrain within Unity, custom terrain needed creation. Unity's built-in toolbars contain the ability to paint textures, grass, trees and associate texture maps to them. Terrain was created as a GameObject and adjusted in width, length and height. Using brushes, the terrain was then pushed or pulled to sculpt it. Once terrain was smoothed over, it was raised and lowered to achieve the desired height. Grass and trees were imported as standard environmental assets and once converted into prefabs, they were painted on by brush size. The shift key was often used to decrease density. To make water in the environment more convincing, an asset titled "WaterProDaytime" was retrieved and the component "UnityStandardAssets.Water" was added to it.

Step 7 [3 hr duration]: Texture Adjustments

This step dealt entirely with the optics of each material. Objects were determined to be smooth, transparent or rough (and to which degree) and thus, properties were enhanced within each material's texture. Glass materials had transparency factors increased and stonework was made rough with a diffused specular highlight. Albedo colours for materials were tuned to achieve saturation levels required and normal maps were added to materials to increase photo-realism. With rougher textures, albedo colours were made brighter. From each image's editor, bump maps were created. The texture maps were accessed from each albedo channel and thus Photoshop's histogram was used to tone the image and enhance the bump file. The contrast in each bump file was heightened and loaded back into Unity's inspector from being linked through the asset folders.



Mesh Collider ☑
Convex ☐
Is Trigger ☐
Cooking Options Mixed ...
Material None (Physic Material)
Mesh Mesh72

Stacked_Stone
Shader Standard
Rendering Mode Opaque
Main Maps
Albedo
Metallic 0
Smoothness 0.2
Source Metallic Alpha
Normal Map 0.18
Height Map
Occlusion
Detail Mask
Emission ☐
Tiling X 1 Y 1
Offset X 0 Y 0
Secondary Maps
Detail Albedo x
Normal Map 1
Tiling X 1 Y 1
Offset X 0 Y 0
UV Set UV0
Forward Rendering Options
Specular Highlights ☑
Reflections ☑
Advanced Options
Enable GPU Instancii ☐
Double Sided Global ☐

figure 6.16 Object Texture Changes in Unity

Step 8 [9 hr duration]: Animating Objects

To create doors that were triggered open by the first person controller, numerous steps had to be taken and multiple attempts were made before achieving a final working door animation. Firstly, a new scene was created and a ground plane object added. Next, the plane was reset to the origin point with respect to the coordinate axes in Unity. From SketchUp, a door file itself was retrieved from the model and saved into the Prefab folder as an fbx file. The door file was brought into the Unity scene was saved separately from the original Unity file. From the GameObject dropdown menu, an empty child was created and renamed as the first person controller. Steps 4 and 5 were recreated and the first person controller family was relocated into models to create a prefab on disk. This new controller prefab was then saved as a scene and the main camera was disabled. From the door hierarchy, the mesh from the frame and glazing were selected and box colliders were added as components to both those objects. As the collider was activated, it meant the object was solid, when in fact this would have to have been toggled on and off during the animation. Through research, it was determined that a “trigger mechanism” was required to be implemented to allow for the animation to function. Additional box colliders were added to animate the handles of each side of the door and act as the trigger. When toggled on, they are applied surrounding the respective object at it’s origin point. Next, a new transition was created and the state of items were changed under the flowchart. After the animation was seen to play fully, a transition had to be created from the animation to the exit point. Without an exit point, the animation would play only once without opening again. To allow the animation to run smoothly, the controller has to arrive, go through the animation and re-trigger the event to allow the door to open/close once more.

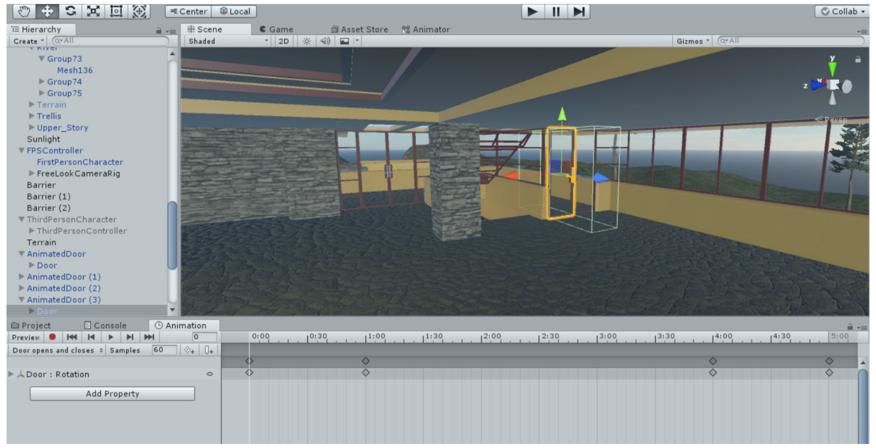
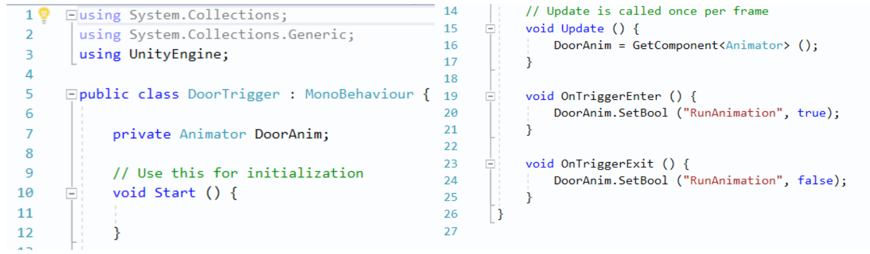


figure 6.17 Organizing the Door Animation Timeline in Unity



```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class DoorTrigger : MonoBehaviour {

    private Animator DoorAnim;

    // Use this for initialization
    void Start () {

    }

    // Update is called once per frame
    void Update () {
        DoorAnim = GetComponent<Animator> ();
    }

    void OnTriggerEnter () {
        DoorAnim.SetBool ("RunAnimation", true);
    }

    void OnTriggerExit () {
        DoorAnim.SetBool ("RunAnimation", false);
    }
}
```

figure 6.18 Scripting Door Animation in MonoDevelop

Step 9 [5 hr duration]: Scripting Animation

Animating within a game engine requires a complex system of events broken into a timeline. The process of building an animation includes the objects itself, the controller, and a script to connect the controller with the trigger. Interactive objects require the creation of environments nested within the larger environment of the VR simulation. These nested environments act as events on loops that are triggered on and off by the first person controller. Hence, from the animation dropdown, a new script was created and renamed as door trigger. The script was opened in MonoDevelop, and the following was typed into the system: “GetComponent <Animator> ();”. Thus, an object named “DoorAnim” was created. The DoorAnim was equated “void” to develop an “OnTriggerEnter” function with no parameters. This function is developed to place what triggers the event into the brackets. After DoorAnim is typed again, “SetBool” is typed in with an open parenthesis so that the autocomplete feature of MonoDevelop responds with the necessary physical response for first person controller upon activating the trigger. The string name typed in was “RunAnimation, (true);”. This value indicates a confirmation of the running animation such that after the first person controller enters the environment, the animation will respond. When the first person controller left the box collider while playing the animation, the code was copied to the clipboard and pasted into the OnTriggerExit and made false. This was done to ensure the animation would stop as the controller triggered the exit. Lastly, in the animation window, the timeframe of the animation was set to 5 seconds and thus the door was set to stay open for 3 seconds. Finally, this animated door was made into a prefab to be applied to the rest of the model. The mesh objects for doors were turned off in the scene and the newly animated door prefab was dragged into the scene.

Step 10 [6 hr duration]: Lighting Adjustments

To create final lighting adjustments, point lights were accessed from the GameObject dropdown and created to fill interior spaces.

Skybox Material: WispySkyboxMat
Sun Source: None (Light)

Environment Lighting
Source: Skybox
Intensity Multip: 1.14
Ambient Mode: Baked

Environment Reflections
Source: Skybox
Resolution: 128
Compression: Auto
Intensity Multip: 1
Bounces: 1

Realtime Global Il ☑

Baked Global Illur ☑
Lighting Mode: Shadowmask

Mixed lights provide realtime direct lighting. Indirect lighting gets baked into lightmaps and light probes. Shadowmasks and light probes occlusion get generated for baked shadows. The Shadowmask Mode used at run time can be set in the Quality Settings panel.

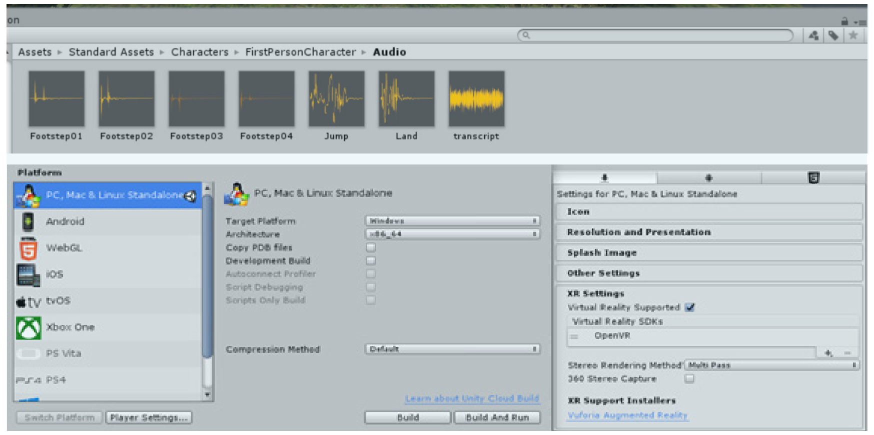
Lightmapper: Enlighten
Indirect Resolutio: 2 texels per unit
Lightmap Resoluti: 40 texels per unit
Lightmap Padding: 2 texels
Lightmap Size: 1024
Compress Lightm. ☑

figure 6.19 Lighting Adjustments in Unity

The size was decreased and soft shadows were set before it was moved into the center of each room. Spotlights were created and added to lighting fixtures that required it. Lighting needed to be baked into texture maps. Lightmap UV generation was turned on for the model and thus illumination was present in every room. After testing the scene using the play button, ambient occlusion and final gar were activated. Lastly, shadows from the inspector window were adjusted for objects to change where light contributes shading. As shadows are also computationally intensive, allowing fewer lights to contribute to shadows allow for faster light bakes and a faster frame rate.

Step 11 [1 hr Duration]: Build Settings

After the scene was tested in Unity and all animations were deemed to work with keyboard input, the simulation was exported for the PC build with the openVR SDK turned on. SteamVR was used to run the scene for viewing on an Oculus Rift Headset.



PC, Mac & Linux Standalone
Target Platform: Windows
Architecture: x86_64
Copy PDB files ☐
Development Build ☐
Autoconnect Profiler ☐
Script Debugging ☐
Scripts Only Build ☐
Compression Method: Default

Settings for PC, Mac & Linux Standalone
XR Settings
Virtual Reality Supported ☑
Virtual Reality SDKs
OpenVR
Stereo Rendering Method: Multi Pass
360 Stereo Capture ☐

figure 6.20 Narrational Audio Check and Build Settings in Unity

Table 5 Efficiency Documentation

	TASK	DATE	START TIME	BREAK TIME	FINISH TIME	TOTAL DURATION
-	1. Research and Tutorials (Part A)	2018-05-14	8:00 am	2.5 hours	8:30 pm	10 hours
	1. Research and Tutorials (Part B)	2018-09-10	8:00 am	4.5 hours	10:30 pm	10 hours
A	2. Export Preparation in Revit	2018-05-24	2:30 pm	3 hours	7:30 pm	2 hours
	3. 3ds Max Import	2018-05-25	3:30 pm	0 hours	4:30 pm	1 hour
	4. 3ds Max Fixes	2018-05-27	7:00 pm	0 hours	8:00 pm	1 hour
	5. Unity3D Project Creation	2018-05-30	11:00 am	4 hours	5:00 pm	3 hours
	6. Object Placement	2018-06-04	12:00 pm	1.5 hours	5:30 pm	4 hours
	7. Materials and Prefabs Adjustments	2018-06-05	2:00 pm	2 hours	8:00 pm	4 hours
		2018-06-15	9:00 am	3 hours	3:00 pm	3 hours
	8. Lighting Adjustments	2018-06-22	6:00 pm	2 hours	12:00 am	4 hours
		2018-06-24	2:00 pm	1.5 hours	7:30 pm	4 hours
	9. Scene Interaction	2018-07-12	12:30 pm	2.5 hours	5:00 pm	2 hours
		2018-07-15	4:00 pm	0 hours	8:00 pm	4 hours
	10. Creating Teleportation	2018-07-17	12:00 pm	0.5 hours	3:30 pm	3 hours
		2018-07-30	10:00 am	0 hours	11:00 am	I hour
	11. Build Settings	2018-08-01	9:00 am	0 hours	10:00 am	1 hour
B	2. Export Preparation in SketchUp	2018-09-15	12:00 pm	2.5 hours	7:30 pm	5 hours
	3. Unity 3D Project Creation	2018-09-16	7:45 pm	0 hours	8:25 pm	1 hour
		2018-09-17	9:30 pm	0 hours	10:00 pm	0.5 hour
		2018-09-18	11:00 pm	0 hours	11:30 pm	0.5 hour
	4. Scene Navigation	2018-09-19	12:15 pm	2 hours	4:15 pm	2 hours
		2018-09-20	10:00 am	1 hour	1:00 pm	2 hours
	5. First Person Controller Adjustments	2019-09-23	9:00 pm	2 hours	3:00 pm	4 hours
	6. Environment Adjustments	2019-09-24	10:30 pm	2 hours	7:30 pm	7 hours
	7. Texture Adjustments	2019-09-25	12:30 pm	3 hours	6:30 pm	3 hours
	8. Animating Objects	2019-09-26	9:00 am	4 hours	4:00 pm	3 hours
		2018-09-29	10:00 am	1 hour	3:00 pm	4 hours
		2018-10-01	10:00 am	2 hours	2:00 pm	2 hours
	9. Scripting Animation	2018-10-02	11:00 am	2.5 hours	4:30 pm	3 hours
		2018-10-03	10:30 am	3 hours	3:30 pm	2 hours
	10. Lighting Adjustments	2018-10-03	8:00 am	1 hour	12:00 pm	3 hours
		2018-10-04	1:00 pm	I.5 hour	5:30 pm	3 hours
	11. Build Settings	2018-10-05	7:00 am	0 hours	8:00 am	1 hour

Results

From this experiment, narrational walkthrough scenes were created to be administered on a PC and demonstrated on an Oculus VR headset. In the previous chapter, a conventional mobile headset was used and yielded a low visual quality for the environment and a complicated export process. Within this experiment, a high quality tethered headset VR experience was used to evaluate the application. This experiment resulted in an extensive duration of two and a half weeks worth of documented work. The evaluation of the practicality overall in the creation of this VR experience will derive from weighing efficiency (speed & duration of scene creation) against efficacy (in creating a compelling walkthrough animation with presentation-friendly capabilities).

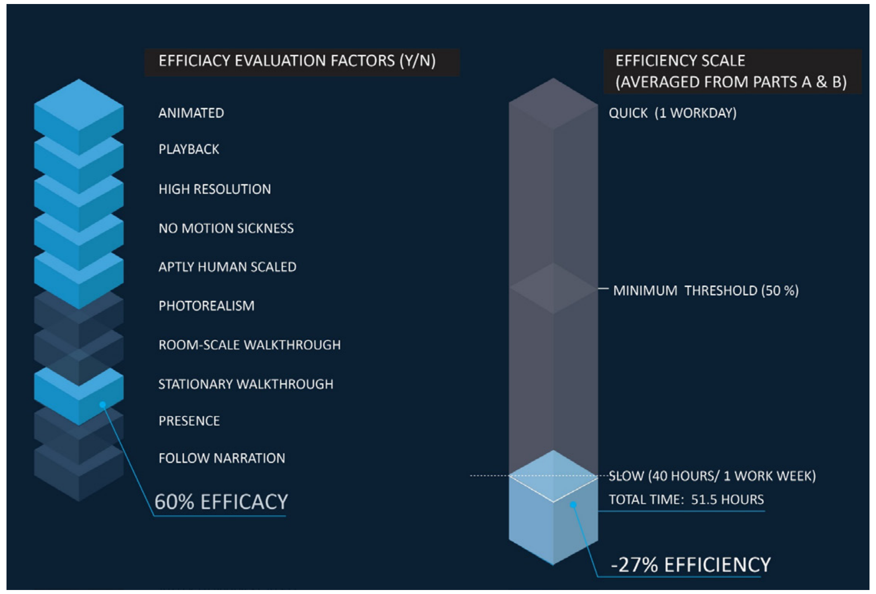


figure 6.21 Animated VR Scene: Practicality

Evaluation of Process

EFFICIENCY:

Evaluating the efficiency of this process highlighted the concerns of a modern fast-paced architectural firm. As the experiment took two weeks to complete with an existing model to adequately configure and animate within a game engine, the time frame was heavily compromised. The experiment yielded 47 hours worth of work for Part A and 56 hours worth of work for Part B. This amounts to a regular work week (approximately 50 hours) spent on either animated walkthrough scene. This timeframe accounts for the work of one person. However, in an architectural firm if this task is split up between multiple people (the logistics of which require further research), it might become a more realistic task. Ideally, in the architectural workplace a week would be forgone to complete a walk-through, yet, if the model posed changes within that specified timeframe, several steps within the process of either scene would need recreation, regardless of files being directly connected into the VR simulation. The majority of time spent within this task went into troubleshooting scene creation elements, scripting animations for doors, stairs, and teleportation and manipulating the visual quality of the environment. To evaluate efficacy, the experiment resulted in both walkthrough animations with narrational audio that did not achieve the expected photo-realistic simulation.

The simulation also revealed door animation glitches and teleportation glitches, illustrating the need for more time spent as required to perfect these components. This highlights the encounter with higher level programming and complexity to build even the simplest of interactive elements. In the context of an architectural design practice, these concepts might not be fully grasped within the timeframe they require to build a visualization. As these coding and scripting principles to understand the chain of events that trigger an animation are quintessential to functionality, the animation process is unforgiving on time spent. Thus, troubleshooting and adjusting lighting/visual quality accounted for the greatest amount of time within this process. The learning curve was steep, and the expansive amount of time spent on research and tutorials was necessary both in the beginning of the process and during troubleshooting as well. Not to mention, research and tutorials also involved sifting through mass amounts of superfluous information to determine what might be usefully applied. This indicates information isn't catered to a specific solution at all times, resulting in a trial and error process. This lengthy learning curve can also be attributed to the fact that as software develops, tutorials become outdated and require recreation, resulting in greater time spent to learn from multiple sources.

As scene creation came with the knowledge of Unity navigational basics (explored in the previous chapter), if this animation was attempted firsthand, additional time would have to be factored in to perfect previously learned elements. The creation of these two animation scenes also resulted in numerous error stages, which presented "road-block" type situations to be dealt with immediately before proceeding. Unfortunately, this compromised the ability to multitask in Unity and forced the notion of conquering tasks in a step-by-step method. With a complex workflow dependant on immediate troubleshooting, it was incredibly easy to experience problems and remain stuck without a solution. Throughout this process, missing a small step from a tutorial or forgetting to switch a simple parameter were examples of mistakes which required attention-to-detail to ultimately ensure success. Therefore, research, tutorials, troubleshooting the animation and improving visual quality were major tasks decreasing the overall efficiency of this process to the documented two weeks worth of time.

EFFICACY:

With both the scenes created, most effort was placed into achieving a photo-realistic visual quality. Unfortunately, the visual quality was lacking photo-realistic aesthetic in both scenes regardless of time spent. Thus, the focus within the environment was shifted into providing a good user interface with intuitive interaction.

This is primarily due to the fact that in VR, a less photo-realistic environment can still be enjoyable provided a good user interface and intuitive interaction are ensured. This experiment teaches a user than VR is much more complex than plug-and-play when it comes to creating animated applications from scratch. Mainly, this is because making a user subconsciously present in a virtual world carries a grave amount of fragility. An experience of a virtual environment with activated presence is fragile. This is due to the human mind having undergone centuries of evolution to perceive the world naturally. Tricking the brain into believing virtual environment to have the effect of natural is a harrowing task. In this way, using awkward transitions in a virtual walkthrough make an experience especially poor.

The uses for an envisioned walkthrough exist first during the design process (to work through design flaws) and second, after the design has been completed (to obtain client-buy in for a compelling and realistic design). Presenting a walkthrough environment isn't valuable unless it can teach a user about a project or stun them with the visual quality of the creation.

As an architectural visualization tool, an animation also must build on the industry's animation photorealism baseline to be practical for the final stages of design visualization. Compromising visual quality within the extensive time frame used questions the value of animation creation altogether. It stunts the Unity game-engine platform into being used mostly within the design phase. This would consequentially deem the process ineffective even more so as the process would lengthen as design changes are made. The iterative process of design would make this even more ineffective unless models were linked and didn't carry drastic changes. Not every software is compatible to allow itself to be linked into Unity and even then, linked models would only reduce the amount of changes needed to be made within Unity.

Similarly, narration also presented issues during both scenes as the timing had to occur perfectly within the experience to guide the user effectively. If there are sounds within the environment as well, those sounds wouldn't be able to play along with the narration track. Narration would also only pertain to a user witnessing the project for the first time and might not be useful for visualizations created during the design process. Narration also only works if the user triggers movements to follow the narrative track correctly. With a room scale VR scene, this problem might not arise as frequently as with a large-scale project. Overall, the loss of presence, photorealism and the ineffectiveness of narration were important efficacy limitations encountered in this experiment.

Limitations & Current Issues

Creating the application and running the application present their own limitations. Creating the application presented issues that can mainly arise with architectural software compatibility. The architectural practice typically uses a multitude of software to model and present designs[1]. Ensuring the compatibility of Revit, 3ds Max and SketchUp with the Unity game engine was an important part of determining interoperability. In Part A of this experiment, using Revit without 3ds Max would have resulted in a loss of materials, textures and lighting information housed within the Revit model originally. This loss in translation indicates that Revit isn't directly compatible with Unity. In Part B of this experiment, SketchUp Pro is the only version capable of exporting an acceptable file format for Unity. This indicates that SketchUp Make isn't compatible with Unity. Thus, not every architectural modeling software could be compatible with game engines as they may present the same issues as Revit or SketchUp with interoperability. Not to mention, each modeling software might present a unique workflow due to these interoperability issues. Revit and SketchUp required varying levels of adjustments made to ensure translation. For example, even though SketchUp Pro was exported directly for Unity, the file required greater editing prior to game engine import than with the Revit file.

This interoperability limitation can also be understood at a higher level by understanding the common workflows of practice in the architectural field. In the industry, projects are currently most commonly drafted in AutoCAD or BIM before being modeled in Revit, Rhino, ArchiCAD, SketchUp, MicroStation, 3ds Max or Maya.[2] Although industry standards for common software usage exists, the constant development of software ensures a wide range of programs used in practice. Where this impacts the architectural industry's usage of game engines is that seamless translation from 3D modeling software is highly variable. Every program's needs in export preparation and UVW mapping can differ, making each workflow process unfortunately atypical. Though files can create a direct link with the Unity Game Engine, this is not unanimously available to all architectural 3D modeling software. The process of editing within a 3D model file has the potential to warrant recreating major steps in the game engine file. Therefore, the workflow is variable and unstandardized, limiting those with interoperability issues.

---

[1] Romullo Baratto, "Trends in Architectural Representation: Understanding the Techniques," ArchDaily, accessed October 23, 2018, https://www.archdaily.com/867060/trends-in-architectural-representation-understanding-the-techniques.

[2] Arch2O, "8 Architectural Design Softwares that Every Architect Should Learn," accessed October 20, 2018, https://www.arch2o.com/architectural-design-software/.

Viewing the application presented issues regarding the digital and physical limitations of synchronization. Due to the fact that timing is of utmost importance when coordinating an animation, it was essential for narrations to line up with movements in the scene. A user had to receive an adequate amount of time to witness each component of the visualization before receiving a future prompt. If any part of the animation glitches or the user was unable to follow through, the entire visualization experience can be an adverse experience with poor user interaction. To accompany this technological difficulty with synchronicity, it can be seen that physically this factor is limiting as well. Within the architectural industry, the demonstration of visualization projects is most commonly done as a seated or stationary experience. Many firms might not have the physical space required to demonstrate room-scale experiences to clients unencumbered by physical objects. This makes teleportation a necessitous factor to experiencing a VR space in the architectural practice. In the creation of both parts of this experiment (where teleportation was administered in one part and keyboard input used in the other) it quickly became evident that the physical space used to administer the VR experience could be confining for a user. As teleportation was built in the first part, the stationary experience proved more pleasant than the latter part where keyboard input had the potential to disrupt synchronicity heavily. The challenges of viewing an application freely at room-scale presents the limitations of synchronizing a virtual experience to a physical one. The primary reason VR works is by allowing the utilization of body movements when experiencing a simulation. To move the body as experiencing a real event in the world makes VR different from simply using a computer's mouse and keyboard. This jarring limitation of viewing the application without synchronicity is therefore a significant concern.

Technical Conclusions

1. The walkthrough animation process consists of numerous trial and error stages. Not to mention, a solution that was apt for another user isn't the solution that will work for every user. Each problem is complex a solution takes numerous tests to find out. Due to this, troubleshooting makes the entire process tedious and increases the amount of times a user is "stuck".

2. Workflow of architectural modeling software into Unity varies and presents compatibility issues. Revit could only be successfully imported into Unity with an intermediary (3ds Max) and SketchUp required adequate of preparation of the file before it was viable to export for Unity (only the PRO version is viable). Thus, different architectural modeling platforms can present their own issues of incompatibility with Unity. A more universally sound process or a built-in plugin to ensure adequate preparation of the file would ensure any file's compatibility and ease of workflow into Unity.

3. Coding/scripting along with the principles of physics are integral features required to be understood when animating a scene. Copying and pasting code is also ineffective as it generates a plethora of warnings due to likely transposition errors. Hence, the need to learn the basics of coding with C# in MonoDevelop is instrumental for animating with Unity. Without learning the basics of C#, a user is unable to create any of the basic animation capabilities Unity provides.

4. Unity needs to develop the ability to link files in from any modeling software to be viable in the iterative design process. Even though Part A used the process of linking a file into Unity, as changes were made to the layout of that file, almost every step taken had changes or edits to be completed within the Unity file. Part B involved SketchUp which doesn't link the file directly or support textures, attributes, styles, shadow settings or layers. If changes were made to the SketchUp file, the process of creating the environment in Unity would have to be recreated.

5. A room-scale VR experience of a walkthrough animation is extremely complex to create effectively. Though this project achieved the creation of a seated VR experience (with keyboard input), a room-scale experience is the quintessential method of experiencing VR. Timing, teleportation and door animation all play a drastic role in ensuring the success of a room-scale VR journey. Both animations unfortunately were only able to be successful seated experiences.

Conceptual Conclusions & Revisiting Research Q2

An architectural walkthrough in VR represents a better way to experience and visualize an unbuilt design when done correctly. In VR, a walkthrough animation can activate presence, scale and provide a greater understanding of an unbuilt space with supplementary narration. Creating content that feeds a real-time rendered animation is a compelling way to depict visualizations. Yet, if a walkthrough animation for VR is built ineffectively, it can provide no greater value than any pre-rendered visualization demonstration viewed on a computer screen. A successful animated walkthrough is thus only achieved when done correctly. This unfortunately, is not a time-sensitive task. Efficiency within this process can only increase as a user becomes more comfortable with the game engine software and their workflow process from the original architectural modeling software. A compelling narrational animated walkthrough is therefore quite complex to effectively achieve. The learning process is an integral and ongoing feature of content creation with Unity3D. Both factors of evaluation, efficiency and efficacy resulted poorly from evaluating this experiment. Prior to conducting the experiment, the timeframe range set for this experiment was 1 day (deemed short) till 1 week (deemed long). With the result of a 50.5 hr average duration (per each part), a narrative walkthrough animation took the equivalent of an average work week (with overtime) to complete. As the lengthy duration is indicative of the extensive necessary effort, this experiment can be concluded as complex. In addition, the efficacy determining factors resulted in a negative response for numerous factors including photorealism, presence and room-scale with the generous timeframe used for this experiment. Therefore, the experiment's evaluation factors indicate the complexity of creating narrative walkthrough demonstrations with

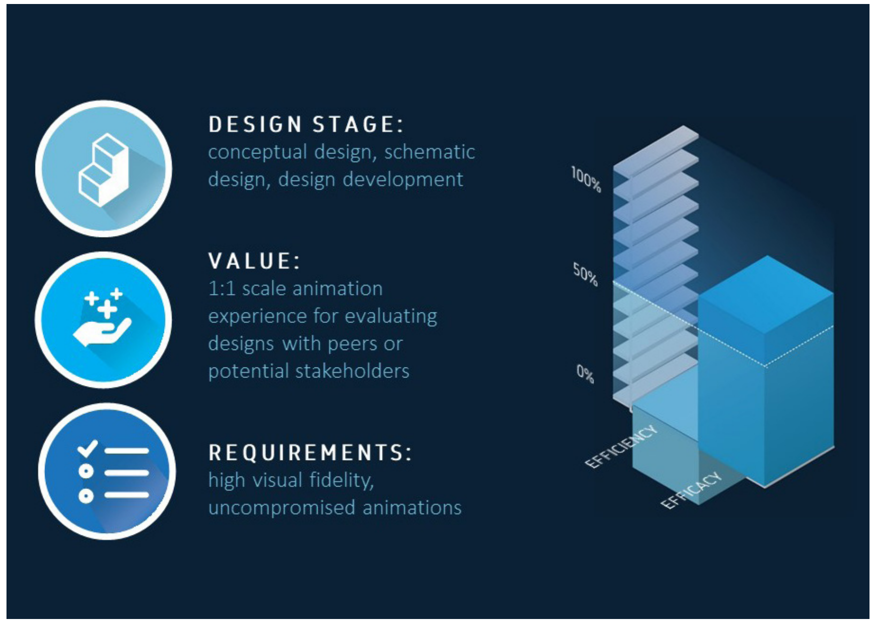


figure 6.22 Animated VR Design: Applications

Chapter 7: VR Scene Option Creation

Research Q3: Can options be easily implemented into VR applications created in Unreal Engine 4?

the generous timeframe provided and substandard efficacy achieved.

This chapter's experimentation focuses on evaluating the ease of creating a scene with options in Unreal Engine 4. Architects rely heavily on the cyclical iterative process of design to work through ideas within a project.[1] To thoroughly envision an architectural design, a project undergoes various technical iterations and changes are made constantly until the design is finalized. A primordial aspect to evaluate a design is the ability to iterate or make changes within a project in VR. Thus, this chapter seeks to verify the difficulty level of building an environment that involves the iterative change of formal and environmental variables, similar to iterations made within the architectural design process. This chapter's experimentation attempts the creation of a virtual environment with levels and teleportation built into the game as it is played. These modifiers of interactivity allow a user to remain still while replacing their virtual environment at the push of a button. With this feature, they can evaluate a design by clicking buttons on their controller to change between multiple scenes they are presented.[2]

To achieve this task, this experiment attempts to create multiple environments housed within the same project and give a user the ability to move between them. By creating an interaction for the transition between the two separate designs presented to a user, they can be offered two options to travel between. This allows them to approach each version of the project from the exact same standpoint, to form a comparative analysis for themselves. Presenting two separate environments in this manner represents the creation of iterations used in the conceptual design phase of architectural projects. Iterations in practice have most commonly occurred as modifications made to drawings or 3D models. The ability to visualize iterations for a project in VR represents an impactful way virtual reality can aid in the conceptual design process. Presenting a client or a peer with options allows for the configuration of different spatial relationships. Hence, evaluating the ease of iterating in VR represents an opportunity to enhance the design process.

---

[1] Martin W. Smit, "The Cyclical Iterative Design Process and Design Environment: learning from experience" (Dissertation, Delft University of Technology, 2008), accessed January 18, 2019, https://www.scribd.com/document/16665247/Cyclical-Iterative-Design-Process-learning-from-experience.

[2] Matt Jackson, "Game Modifiers: Create big value for little cost to your game," accessed February 1, 2019, https://deepfriedgamer.com/blog/game-modifiers.

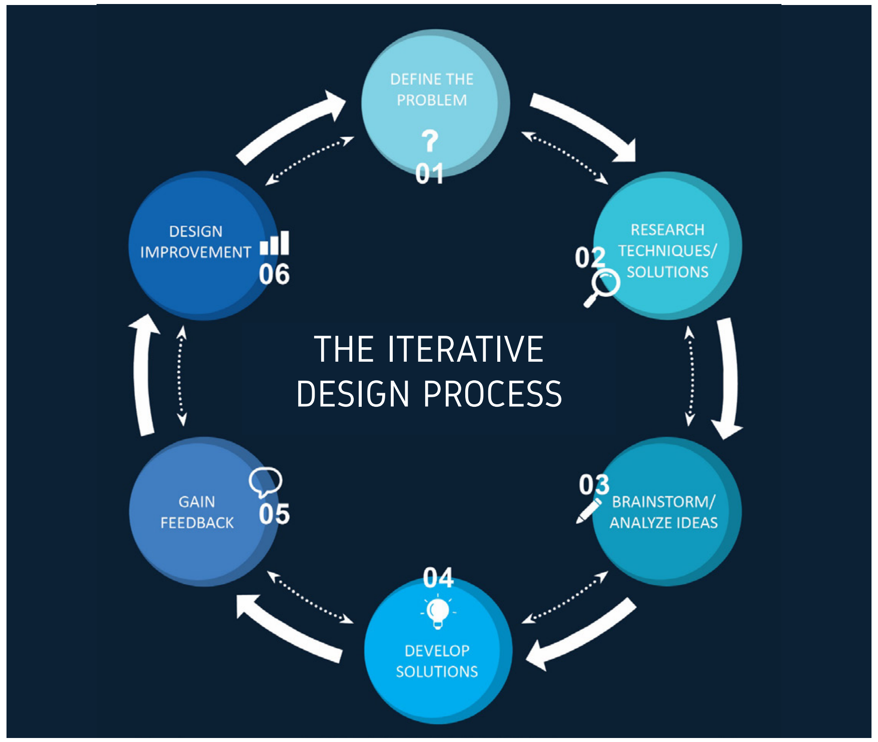


figure 7.01 The Iterative Design Process

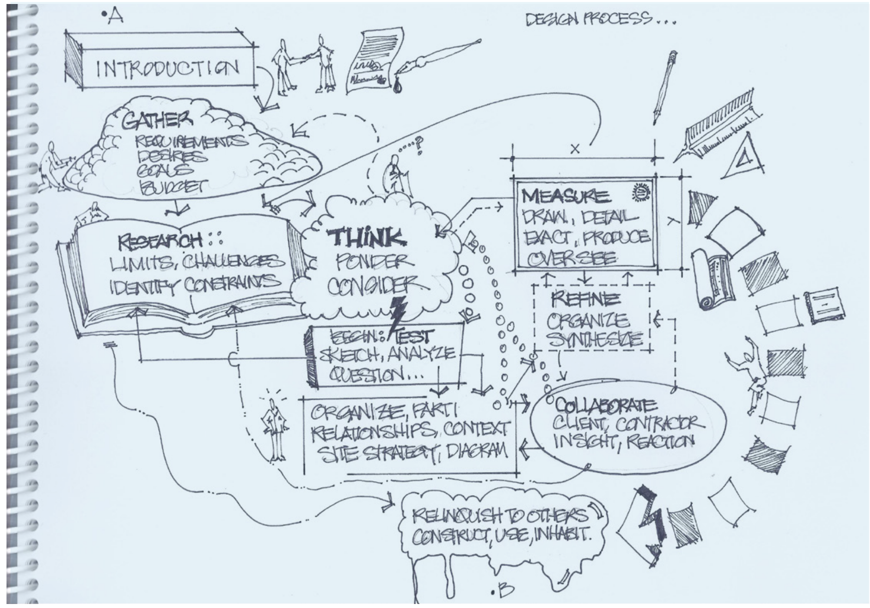


figure 7.02 Architectural Design Process

Providing a user the ability to choose between options inflicts change within a scene and ultimately carries the potential to enhance interactivity. As discovered in the previous experiment of Chapter 6, changes cannot be translated from an original architectural model seamlessly, as expected, in Unity. This is attributed to the principle that Unity simply imports models and doesn't always offer the ability to link a model file into a VR project. Changing an original model file therefore requires the new model to be imported. This entails changes once more to be made in Unity to reflect the alterations made for the model file. Depending on the degree of modifications made to the model, material assignments, lighting changes and animation changes would need to be made to reflect that. Whereas, if a model is linked into the scene, the scene would need a significantly smaller amount of changes to be made. It was discovered that to build options in the form of two architectural designs, they would have to be self-contained within a game engine project as separate imports and to use animations to trigger navigating between them. Making changes to an architectural model while working in Unity proved to be challenging and tedious due to the difficulty of animating interactivity. Therefore, this chapter seeks to evaluate creating iterative options with Unreal Engine 4, a more suitable game engine to facilitating physical and visual interactivity.

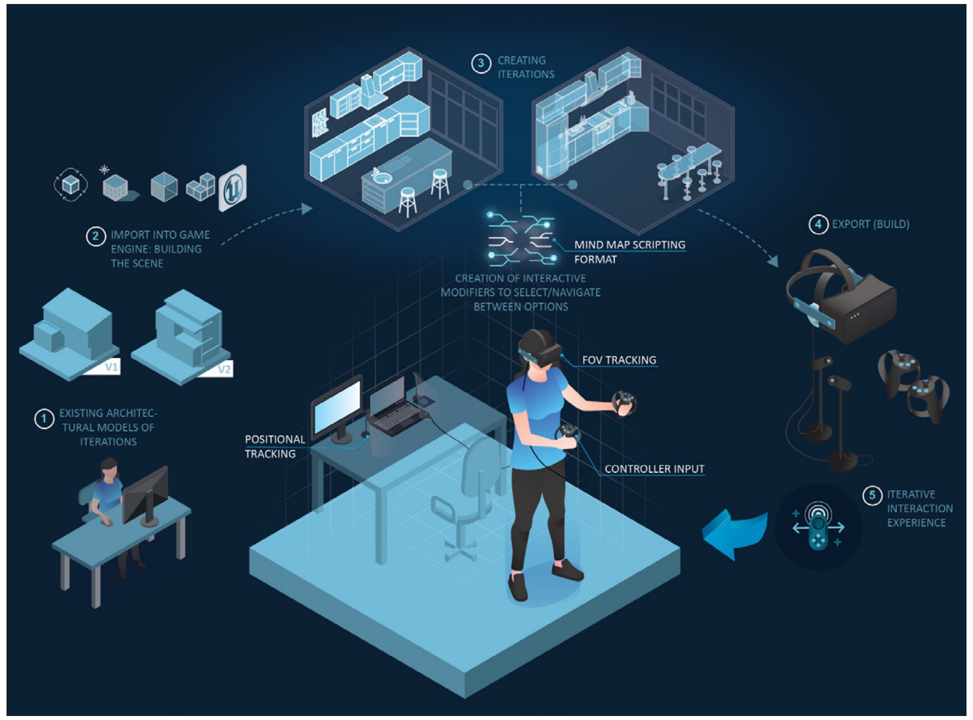


figure 7.03 Interactive Scene with Options Creation Process

Notably, this process is also quintessential in determining the differences between Unity and Unreal. Though Unreal and Unity are similar in that they are both game engines, Unreal uses a different script and contains a completely different interface (see Chapter 4). From exploring Unreal in this chapter, a differentiation can be made within the process of working with the two different game engines and the difficulty/ease that comes with such. This chapter will not only to examine interactive option creation in a room-scale demonstration but also comprehensively understand game engines within the iterative design process of architecture.

Definitions[1]

1. Interactive Modifiers: player-facing choices that can be made during the course of running an application or game.[2] Providing a user with a modifying option set changes the virtual experience by adding to the number of outcomes along with increasing game interactivity.

2. Projects: the compilation of all code and content used to create a game or application. However, projects in Unreal Engine 4 (UE4) specifically can be created, developed and maintained alongside one another with the use of the Engine and Editor, to use between easily switch between files.

This factor creates the ability to work between multiple Unreal projects simultaneously and create tests projects alongside an overall project.

3. Objects: the baseline building blocks in Unreal, which carry the invisible functionality required for assets in an Unreal project. Functionality is most commonly inherited by assets through objects in Unreal. UObject is the baseline class for all objects. This allows for the implementation of features and metadata to transcribe elements into the Unreal Editor. Once assigned, they contain functional attributes, which are referred to as components, which allow it to become a character, environment or special effect. Objects can be compared to the creation toolkits prominent in any modeling or drafting software, that allow for the conception of simple surfaces, lines, or solid shapes (baseline objects).

4. Classes: used by Unreal to define the parameters of actors and objects within a scene. They behave hierarchically and inherit information from parent classes. Classes allow the Unreal Engine to rank scene elements and allow information to be shared by multiple objects/actors. Classes are developed in C++ code or through using Unreal Engine's own Blueprint (Unreal's own gameplay scripting system with a node-based interface). Common classes include the PlayerController class and the AIController class.

5. Components: the functional capabilities attributed to actors in Unreal. Components cannot exist alone and must be attributed to actors. When attributed, the actor will gain the functionality specified by the component. An object can contain any number of components. Unreal has a variety of built-in components, and unique ones can be created by writing scripts that inherit commands from Blueprints and C++ scripts. Basically, components are the nuts & bolts of objects and behaviors in a game. An object acts as a container for many different components. In computational architectural design tools, components can be compared to the "Properties" toolbar assigned to any object. As examples, components allow spot lights to be emitted from an actor, rotating movements to be attributed for an actor to spin and audio to be attributed so an actor can play sounds.

6. Actors: represent any object that is placed into a level. Actors are all part of the generic Class. Actors support 3D transformations including scaling, translations and rotations. Actors are created or spawned within an Unreal project and also deleted through the use of Blueprints and C++ scripts. Actors all carry the AActor class; which exists as their own baseline. Types of actors include (but are not limited to) StaticMeshActor, CameraActor, and PlayerStartActor. Actors contain a subclass of pawns, which are in-game avatars or persona (i.e. characters in a scene). Characters in Unreal are subclasses of pawn actors, whom are intended to be used in the first person. This character subclass of pawns is used to configure collision setups, inputs for movement and code required for player-controlled movement.

7. Brushes: actors that describe invisible guidelines denoted by 3D volumes within an Unreal project. They are used to prototype scenes and block out levels while an application is being tested. Brush Volumes can have effects attached which allow them to function as triggers for actors who enter/exit them.

8. Levels: user-defined areas within a scene. They can be accessed and altered through transformations or edits made to it's containing actors. Levels are also saved separately within a project. Levels are shown within a list in a World, which handles streaming from one level to another and the creation of dynamic actors. Levels facilitate the creation of multiple models consolidated into a single project.

Parameters of Experimentation

Software: 3DsMax, Unreal Engine 4 (Game Engine)
Hardware: PC, Oculus Rift (Tethered Headset)

---

[1] Epic Games Inc., "Unreal Engine 4 Terminology: Get Started with UE4," accessed January 12, 2019, https://docs.unrealengine.com/en-us/GettingStarted/Terminology.

[2] Matt Jackson, "Game Modifiers: Create big value for little cost to your game," accessed February 1, 2019, https://deepfriedgamer.com/blog/game-modifiers.

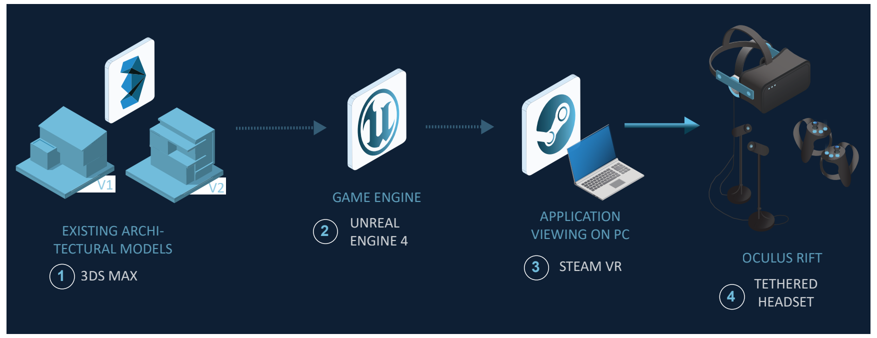


figure 7.04 Workflow: Iterations to VR

Process & Documentation of Creating a Scene in Unreal

Step 1 [6 hr duration]: Research and Tutorials[1]

As previous chapters dealt with the Unity game engine, which differs from Unreal significantly by both interface and navigation, the software (UE4) had to be learned before the experimentation process began. Online tutorials were accessed and used to learn how to navigate Unreal, deal with assets and manage a project. Once again, this process involved understanding how a project is set up, exploring controls and component functionalities. Aside from understanding the basics, tutorials in this stage were also accessed in order to engage the idea of iteration and devise a plan to create a game with options. During this stage, material learned from using another game engine was applicable in a number of areas. Thus, the learning curve was slightly reduced. Yet, as the Unreal game engine required a project to be managed through differing elements, the learning curve was still reasonably present. The majority of tutorials to understand iteration were focused on learning Unreal Engine's usage of levels, navigation, meshes and key-bindings in Blueprint.

Step 2 [1 hr duration]: Template Setup in Unreal Engine

Unreal possesses two types of templates to begin working with a project; code-based and blueprint-based. Blueprints are the equivalent of visual scripting whereas coding involves directly scripting in C++ for Unreal. Templates are designed to help with starting out in a project. As a new user, a blueprint-based approach provided a means of having a procedural setup for things without having to dive deeply into concepts specific to coding or programming. The virtual reality template was accessed to begin. The template was set to keep the maximum quality of the scene. Blueprint templates are the online resource to access objects, meshes and materials useful to construct a project. The basic Motion Controller map was accessed to automatically calibrate performance enhancements specific to that of a virtual reality project. Rendering for VR projects are intensive, as they render scenes for each eye, making the entire process extensive on common hardware. The ability to build a scene with the selected template setup allowed for interactions such as teleportation and intuitive movements to be set automatically. 3D modeling assets of the two project iteration were thus imported into this file, to build from and develop the project. This was completed to ensure that the teleportation settings and graphic enhancements for VR usage are already optimized with minimal programming before developing the environment. The environment was then optimized for virtual reality viewing, leaving challenging settings or scripting taken care of.

---

[1] Lynda.com, "Unreal for Architecture and Visualization Courses," accessed November 10, 2018, https://www.lynda.com/.

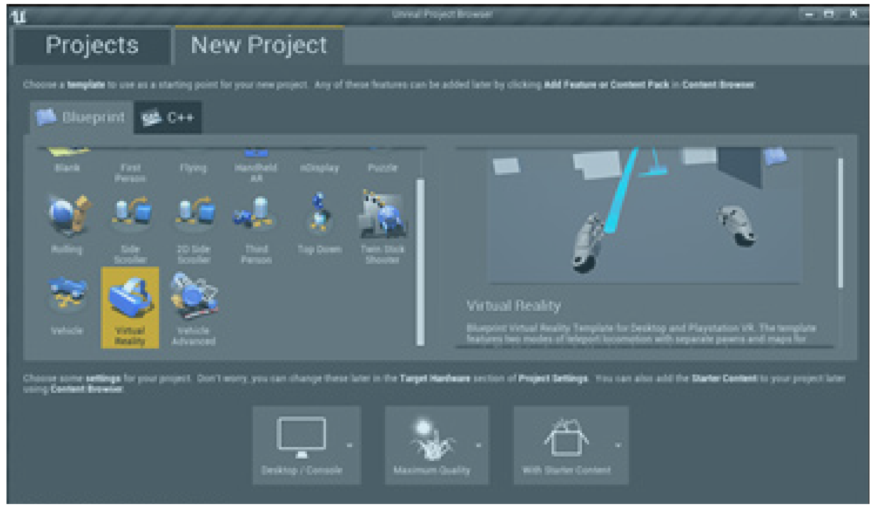


figure 7.05 Project Template Setup in Unreal

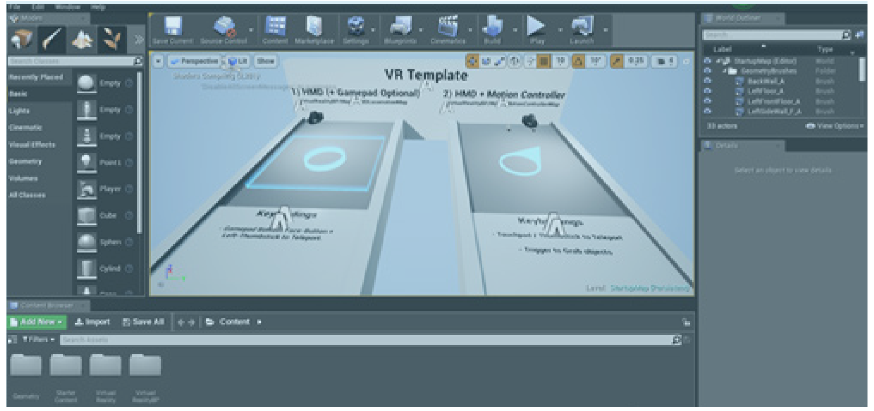


figure 7.06 Project Navigation in Unreal

Step 3 [5 hr duration]: Consolidating Assets

Building the assets for the scene required importing project and model files, materials and textures and organizing all elements into their specific folders to ensure the creation of a functioning directory. As discovered in previous experimentation, it was quintessential to organize project contents into simple subdirectory structures. Before the room project fbx files were brought into Unreal, the existing floor space was deleted. Hence, as the room model was brought in, a new built floor was envisioned to replace the floor that existed from the template. Next, All the folders in the directory were re-organized and made to include subdirectories. This was created to consolidate materials, meshes and textures into their own specific folders immediately as the project is setup. As the model is brought in, the teleportation frame within the entire project had to be altered. A modifier volume, called “Nav Mesh Bounds” was added to the environment and scaled to enclose the boundaries of the scene, to ensure the user does not teleport into an unbuilt portion of the project. The goal within this step was to keep the file management neatly organized. If unwanted assets and data going unused were still present in a project, the application would have the potential to be bogged down carrying that information. It was of best interest to thus purge all unwanted assets and data in the file as it was being organized.

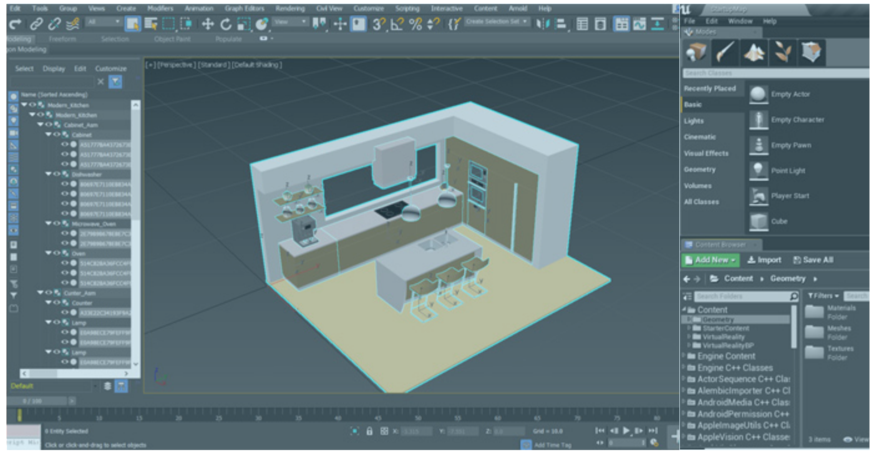


figure 7.07 Project Fixes in 3ds Max & Mesh Import into Unreal

Step 4 [3 hr duration]: Importing Meshes

For this project, two simple rooms were built to transition between in the project, to act as the options for the scene. Models imported into Unreal are were created as meshes. Before a model is output from architectural software into the fbx format for Unreal to import, a number of tasks had to be completed to ensure the file was well maintained. All surface edges had to be verified to have snapped perfectly to others, the subsequent vertices of which were ensured to be fused together. When this task isn’t completed, textures stretched across surfaces in have the potential of breaking. These seams then become increasingly noticeable during a demonstration. Next, the face normals were checked to ensure meshes were kept at the lowest polygon count as possible, without losing any mesh resolution. After, the fbx files were imported into Unreal. To verify distance and scale within the project (as the models were built in the metric setup), the measure tool was used from Unreal to calculate distances and verify the accuracy of units as the meshes were brought in.

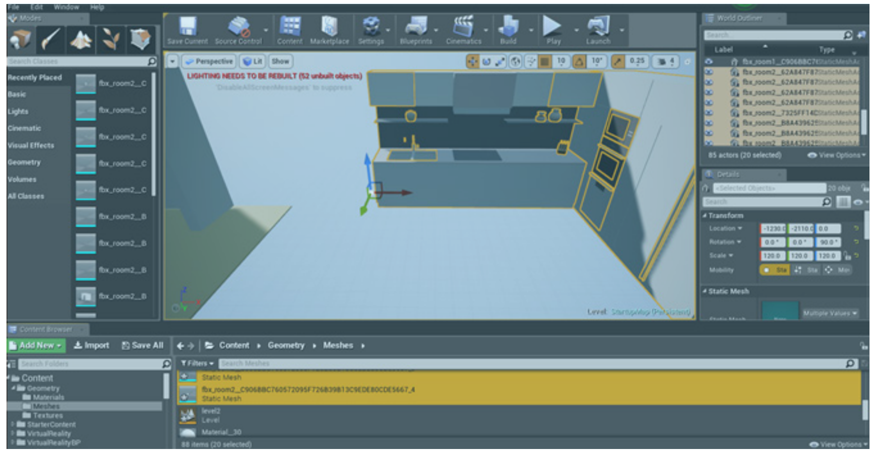
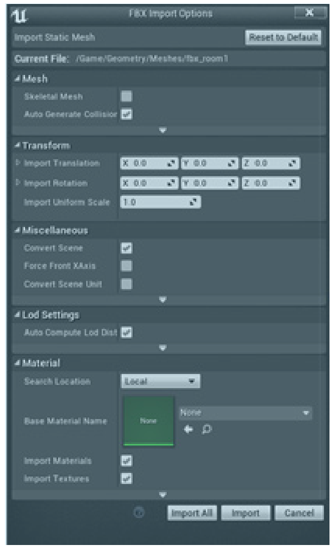


figure 7.08 Mesh Positioning and Property Edits in Unreal



FBX Import Options

Import Static Mesh

Current File: /Game/Geometry/Meshes/fbx_room1

Mesh
Skeletal Mesh ☐
Auto Generate Collision ☑

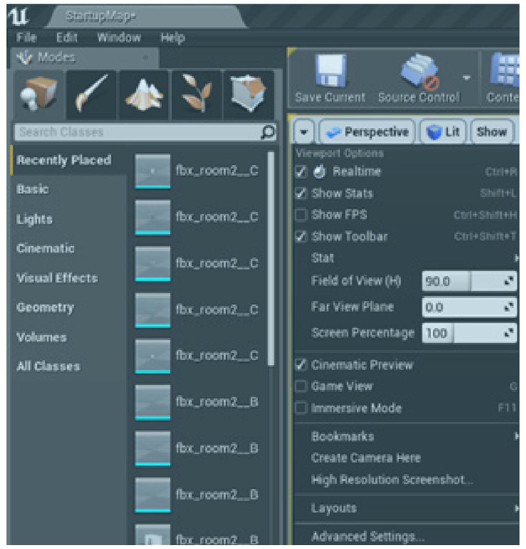
Transform
Import Translation X 0.0 Y 0.0 Z 0.0
Import Rotation X 0.0 Y 0.0 Z 0.0
Import Uniform Scale 1.0

Miscellaneous
Convert Scene ☑
Force Front XAxis ☐
Convert Scene Unit ☐

Lod Settings
Auto Compute Lod Dist ☑

Material
Search Location: Local
Base Material Name: None
Import Materials ☑
Import Textures ☑

figure 7.09 Unreal Component Edits



☑ Realtime Ctrl+R
☑ Show Stats Shift+L
☐ Show FPS Ctrl+Shift+H
☑ Show Toolbar Ctrl+Shift+T
Field of View (H) 90.0
Far View Plane 0.0
Screen Percentage 100
☑ Cinematic Preview
☐ Game View
☐ Immersive Mode F11

figure 7.10 Collision Adjustments in Unreal

Step 5 [1 hr duration]: Mesh Placement

Grouping the meshes that were brought in was an integral measure necessary to minimize the number of pieces being brought into the scene separately. Fbx is a file format that could effectively take advantage of the grouping aspect and minimize the workload within an Unreal project. To place meshes into the scene, the World origin was used. This aspect set every mesh or mesh group's pivot to begin at the origin, such that they could have been easily dragged after setting location coordinates to the origin. Underneath the transform toolbar, all the meshes placed were then ensured to be in line with one another under the transformation toolbar.

Step 6 [4 hr duration]: Scene Navigation

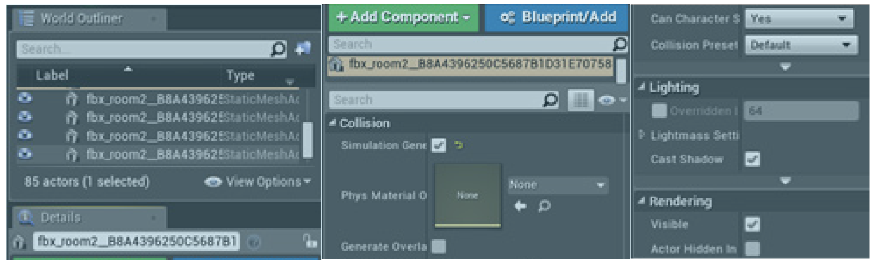
As the main model had been imported, the scene was now ready to be have animations added. So, field-of-view and world scale were accessed to increase a user;s head motion range within the scene. To edit this feature, the Unreal Editor was accessed and the VR Pawn Player in the outliner's blueprint was edited. Within the camera settings of this player, the degree aspect of the camera's settings was increased to mimic the real-life field-of-view (of a forward facing range of approximately 210 degrees). To correlate this change, the world view was also edited to account for the size and field-of-view in the space, after playing and testing the scene. Collisions were setup to ensure geometry and landing areas with objects were built with interactions. This was also necessary to ensure the user wasn't experiencing the scene's unintended areas. Colliders being set up allow for a guided experience.

Step 7 [2 hr duration]: Collision Adjustments

For many objects in the scene, Unreal allowed for collisions to be automatically generated underneath the mesh menu. As the floor was removed previously and its subsequent collider setting, a new collision was set to generate automatically for the floor in the scene. Colliders were next added for the walls and furniture within in the space. Auto-generating collisions could have been used but were understood to bog down the processing of the scene. Hence, collisions were built in and around specific areas adjacent to walls. Collisions were placed, duplicated, rotated and snapped to fill the scene wherever needed. Next, the first-person controller's capsule was edited. The capsule represents the physics and collision force around a character, making it a crucial consideration toward the setting boundaries of travel for a user. After collisions were added, the best way to verify whether the collisions were properly functioning was to test the game. Tests were conducted multiple times after new colliders were integrated into the scene. From testing the environment and walking around, the environment was ensured to have its collisions looking and behaving appropriately. After, the first person controller's collider was edited within the outliner.

Step 8 [3 hr duration]: Texture Adjustments

In reality, texture maps play a significant role in presenting photorealism. Maps help carry light across a surface and can help preserve the detail of textures. The scale of textures within the scene were then edited to mimic that of reality. As humans have the natural ability to sense scale being incorrect or having a discrepancy within the environment, testing the scale of an environment and making edits to reflect natural scale was an essential step in this process. When textures were brought into Unreal, the engine carried the capacity to compress these textures. Within the texture editor, settings such as brightness, saturation and vibrancy were simple settings to tweak. Textures were converted to materials and then applied into the scene. The material editor built into Unreal was used to preview textures, edit nodes and configure all the details and attributes associated to textures. Materials were then applied to assets by being dragged and dropped into the viewport.



85 actors (1 selected)

fbx_room2__B8A4396250C5687B1

fbx_room2__B8A4396250C5687B1D31E70758

Simulation Gene ☑

Phys Material O None

Generate Overla ☐

Can Character S Yes

Collision Preset Default

Overridden I ☐ 64

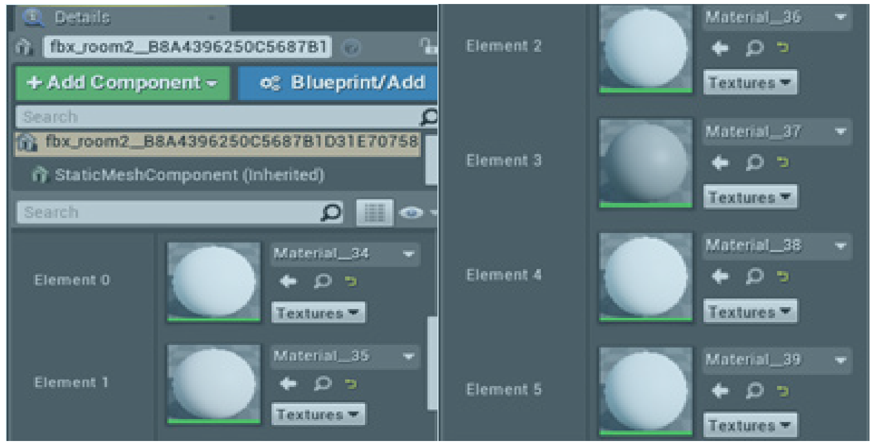
Lightmass Setti

Cast Shadow ☑

Visible ☑

Actor Hidden In ☐

figure 7.11 Texture Adjustments in Unreal



fbx_room2__B8A4396250C5687B1

fbx_room2__B8A4396250C5687B1D31E70758

Element 0 Material__34

Element 1 Material__35

Element 2 Material__36

Element 3 Material__37

Element 4 Material__38

Element 5 Material__39

figure 7.12 Texture Property Editing in Unreal

Step 9 [5 hr duration]: Scene Interaction & Option Creation

Scene interactions were discovered to be reasonably complex features to activate. However, this process was far less complex to rationalize than that of previous investigations. The process included the use of parametric chains of command. The controller was connected to an interaction by the use of a corresponding blueprint chain to formulate events. The majority of interactions are created with the use of inputs as they allow for functions to be executed and interactions to be programmed without directly typing script (using a more comprehensive method of intuitive interaction). Blueprints act to trigger functions in an environment and can be reused and applied to multiple actors (objects) within a project. To activate creating interactions within the project, first the project settings were accessed. Under the general settings, the input sector allows a user to create settings from manual input. The bindings sector allows key-bindings to be set within the simulation. Input can be made from keyboard settings, controllers or mobile settings made. Bindings were discovered to allow for specific action mapping such as level changes. These action maps set up in project settings were then used to access blueprints for the first person character. From the input section within the blueprint manager, the action was searched and associated to the first person character. The controller input was set to trigger a level switch for the scene. This would allow for the ability to transition between two levels within the game. Finally, the first person controller's walking speed was edited to function at a slower walk speed.

Action and Axis Mappings provide a mechanism to conveniently map keys and axes to input behaviors by inserting a layer of indirection between the input behavior and the keys that invoke it. Action Mappings are for key presses and releases, while Axis Mappings allow for inputs that have a continuous range.

Action Mappings
TeleportRight
GrabLeft
GrabRight
TeleportLeft
HMDTeleport
NewLevel
MotionController (L) Thumbstick Shift ☐ Ctrl ☐ Alt ☐ Cmd ☐
Axis Mappings
TeleportDirectionUp
Gamepad Left Thumbstick Y-Axis Scale 1.0
TeleportDirectionRight
MotionControllerThumbLeft_Y
MotionControllerThumbLeft_X
MotionControllerThumbRight_Y
MotionControllerThumbRight_X

figure 7.13 Interaction Modification in Unreal

figure 7.14 Blueprint Editing in Unreal

Step 10 [3 hr duration]: Lighting Adjustments

Lighting in Unreal is essentially driven by two factors; real-time lighting in the scene and render settings. To enhance the project's photorealism, the generation of lightmap UV's was turned on under the static mesh settings. Next, ambient occlusion was activated and assigned to materials. Simple light sources were then added into the project and their intensities were edited to suit preferences toward the scene. Creating an overall sunlight system setup was the next step within the lighting process. Spherical refraction was added to manipulate the shape of light diffusion from light-emitting sources. This process had to be executed carefully as an increase in the amount of lights ensures an increase in the overall build time when light is baked into the scene. Lightmaps were built as a composite over top of textures to gain an overall effect of baked lighting within the scene using lightmass settings. Next, the "build lighting only" setting was activated to preview the final lighting of the scene. Last adjustments for this step included making post process volume setting adjustments and adding ambient occlusion. With these steps, the overall quality of render settings for the project were updated. Another test of the scene was completed to ensure for accurate lighting, shadows and an overall aesthetically photorealistic animation. The goal within this step was to ensure the output is a simulation as close as possible to the rendering standards of photorealism commonly expected of architectural visualizations.

Step 11 [1 hr duration]: Build Settings

Under packaging settings, the project was built using package project settings. In this final step, the build map was defined and the output standards set. After the VR simulation was exported for PC, SteamVR was used to run the scene for viewing on an Oculus Rift Headset.

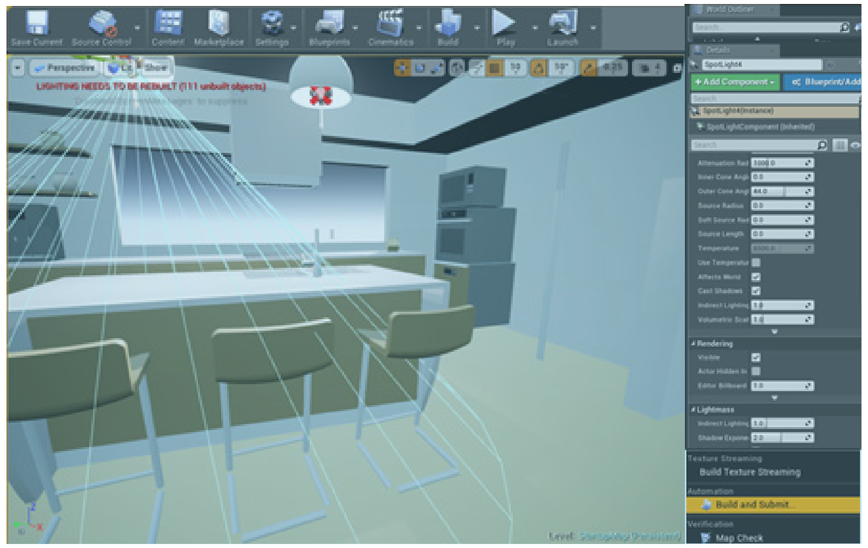


figure 7.15 Lighting Build and Project Export in Unreal

Results

Through this experimentation, a simulation in Unreal Engine 4 was created with a controller trigger to change between project levels. This experiment's simulation was created to be administered on a PC and demonstrated on an Oculus Rift VR headset. A high quality tethered headset VR experience was used to evaluate the application. Similar to previous experimentation chapters, the evaluation of the practicality overall in the creation of this VR experience will derive from weighing efficiency (speed & duration of application creation) against efficacy (in creating a compelling VR experience with options/ iterations included for a user).

Table 6 Efficiency Documentation

TASK	DATE	START TIME	BREAK TIME	FINISH TIME	TOTAL DURATION
1. Research and Tutorials	2018-10-20	10:00 am	2 hours	6:00 pm	6 hours
2. Template Setup in Unreal Engine 4	2018-11-10	3:30 pm	0 hours	4:30 pm	1 hour
3. Consolidating Assets	2018-11-11	2:00 pm	0 hours	7:00 pm	5 hours
4. Importing Meshes	2018-11-18	11:30 am	2.5 hours	5:00 pm	3 hours
5. Mesh Placement	2018-11-25	1:15 pm	4 hours	6:15 pm	1 hour
6. Scene Navigation	2018-12-07	12:00 pm	1.5 hours	5:30 pm	4 hours
7. Collision Adjustments	2018-12-15	5:00 pm	2 hours	9:00 pm	2 hours
8. Texture Adjustments	2018-12-16	12:45 pm	0.5 hours	4:15 pm	3 hours
9. Scene Interaction & Option Creation	2018-12-17	9:00 am	4 hours	6:00 pm	5 hours
10. Lighting Adjustments	2018-12-18	12:30 pm	2.5 hours	6:00 pm	3 hours
11. Build Settings	2018-12-20	7:30 pm	0 hours	8:30 pm	1 hour

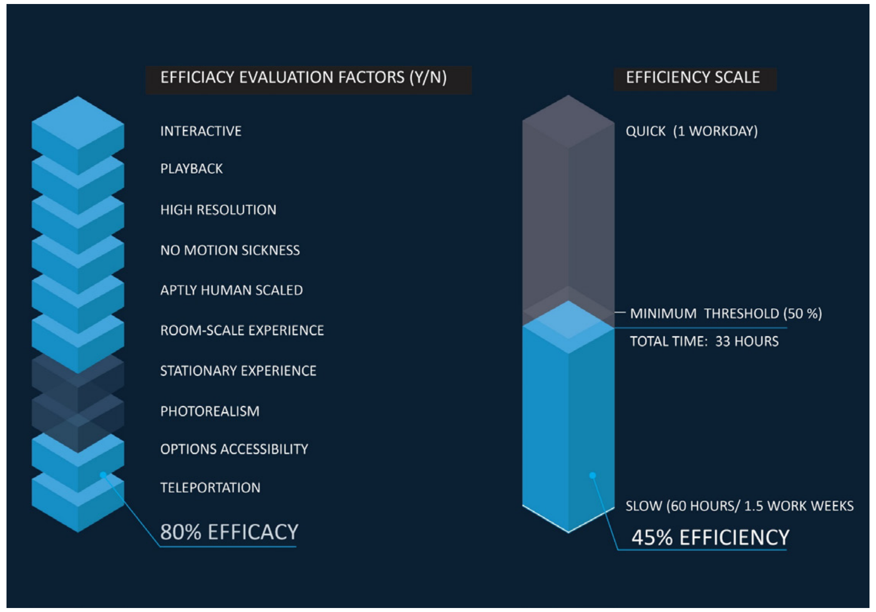


figure 7.16 Iterative VR Scene: Practicality

Evaluation of Process

EFFICIENCY:

From the initial evaluation factors defined, the simulation was created within a reasonably low time frame of 33 hours. Assessing the result from the perspective of a common fast-paced architectural environment, a designer might recognize that this time frame is suitable for the conception of a functional VR scene with iterations. This is because the project was built in less time than the average work week of 40 hours. Configuring interactivity while working with new software (as proven by previous experimentation) is a harrowing task. This experiment's positive result yield asserts the ability for an increasingly intuitive learning process associated with UE4. In an architectural firm, working and completing a project in VR within a week's timeframe would diminish the fear of changes being made to the project's concepts during an exceedingly drawn out timeframe. Though this experiment yielded a successful duration, the majority of time spent in project creation were attributed to initial research, consolidation of assets and creating interactivity. Troubleshooting working with blueprints in the trial and error process of interactivity played an integral role in extending the learning curve. The simulation required many attempts to make interactive features run smoothly. A lot of time went into making edits and repeatedly testing the scene intended functionality was achieved. Troubleshooting also added to the timeframe due to the fact that blueprints carry a considerable learning curve (as opposed to the scripting approach taken in earlier investigations). An expansive amount of time was spent on research and tutorials, which was necessary both in the inital stages of the process and during troubleshooting. As indicated in previous experimentation, this factor represents information not being catered to a specific solution at all times and the inherent trial and error process in effect while searching for an adequate solution. Also, it can be noted that this duration might not apply to every scenario, as extensive game engine research was conducted in general (though past experiments) before this experiment was attempted.

Having a knowledge base of another game engine, simplifies the learning process considerably. Therefore, additional time would technically need to be accounted for as this experimentation is not a first-hand attempt at game engines in general but simply Unreal Engine 4, specifically. The efficiency of this process was also aided with the blueprint-style of scripting available to UE4, rather than the strictly code-based methods used beforehand. The design of the Unreal game engine thus facilitates a greater, more intuitive-based approach to programming, rather than one that is heavily scripting-based. With the creation of well designed pre-enhanced templates, the early stages of content creation were considerably eased. From the architect's perspective, where dynamic movements as cause and effect scenarios are second nature (triggered by nodes rather than written lines of code), this feature is exceptional in easing a user's way in to content creation. With the process heavily dependant on trial and error, with a minimal amount of "road blocks", creating a VR demonstration in UE4 proved itself to be reasonably efficient. Therefore, research, tutorials, troubleshooting the animation and improving visual quality were the major tasks decreasing the overall efficiency of this process. Yet, the intuitive based-approach of the software and previous game engine experimentation accounted for a worthwhile increase in said efficiency.

EFFICACY:

The efficacy of this experiment resulted in an exceedingly effective output in comparison to previous experimentation with Unity. Most effort within this experimentation was placed within creating a working interaction. As visual quality was originally not the focus of this experiment, the lighting settings could have used further adjustment. In hindsight, it was learned that more time could have been allotted to increasing the photorealism of the animation. By this standard, the focus within the experiment was geared heavily into providing a good user interface with intuitive interaction. This resulted in an enjoyable experience, yet one that could be enhanced with photorealistic visuals. Again, this indicates as in

Chapter 6, that in VR, an experience that has an un-photorealistic environment, but a good user interface and intuitive interaction, can still be an enjoyable experience. This experiment teaches a user that VR can provide an entirely new understanding and appreciation of a design. Jumping into VR during the concept and layout process allows the designer to approve an overall "proof of concept". Likewise, the ability to test a project in context and iteratively express ideas can be effective in working through choices.

The efficacy of this experiment was enhanced through Unreal's integration of the blueprint-approach to programming functions. Interactions are normally a difficult task to creators without a background in programming. Compared to writing lines of code, working with attaching nodes in Unreal's Editor allows for professionals to design interactions based on principles of physics. If an effective interaction can be created without the need for writing code, architectural practitioners can easily be able to design VR projects without a steep learning curve. In this way, iterative VR simulations have the ability of being created with a greater degree of efficacy with the loss of a steep scripting learning curve. Iterations in virtual reality carry the ability to thus work through design flaws and create options for users if programmed effectively. With an increased amount of time spent on achieving a high-resolution photorealistic baseline (and waiting the due time for the game engine to process that real-time rendering information), projects also carry the possibility of being used to obtain client buy-in.

Although, to increase the efficacy to achieve a presentation-worthy simulation with iterations, a greater amount of time would be required to hone in on photorealistic lighting effects and texture mapping. Overall, the only ineffective aspect of the visualization was with regards to photorealism. All other factors including interaction, options, teleportation, resolution and room scale were built effectively within the denoted time frame.

Limitations & Current Issues

The creation of VR scenes in UE4 carry a greater focus on working through designing intuitively. Though this process benefits architects and designers when it comes to programming intuitively, other factors can rely heavily on "preference". While choices in designing are beneficial, working with parameters in a game engine that deal with preference increases the trial and error process while working with the software. This can be beneficial but from this experience was also a tedious and repetitive task. For example, collision settings and field of view were all factors that were modified preferentially, which might not make the settings the most aptly configured from a programming standpoint. As a designer will likely be guessing on these preferential factors, there is a greater chance of typical settings being modified incorrectly. A prominent example of this occurs with factors such as depth of field, which can make a viewer uncomfortable in their environment, and in worst cases claustrophobic or motion sick. Disorienting camera effects are often added to increase post-processing visual quality. Thus, if future updates to the software include a database of templates focused towards architectural visualization was implemented, the harrowing trial and error process linked to user-specific preferences could be reduced for standard VR tasks. Similarly, starter content in UE4 can bloat a project's directory heavily, indicating a need and use for more specifically catered starter content.

Technical Conclusions

1. Scale is of paramount importance. Only if a scene is built adequately to scale can all its contained elements be functional. Creating a scene in VR that mimics the real world is the primary goal and a considerable amount of trial and error is therefore necessary. The scale factor has to be perfectly adjusted, such that the objects and textures can drive the VR experience effectively.

2. A great deal of consideration is required to go into understanding the impact of visuals on a user. The smaller the field of view, the greater level of discomfort a user will feel. Inversely, the larger scale factor of field of view will result in less user discomfort. Similarly, visual effects and camera-based or screen-spaced visual effects must be avoided at all costs. Also, lighting needs to be less powerful in environments. For a user jumping into a scene, a bright environment can be unappealing and harmful.

3. Asset management, creation and use must be factored into the design process. For an increased performance in a visualization, efficient assets carry the potential to speed up or bog down a project heavily simply based on asset organization. UE4 puts an enormous strain on hardware and this requires alleviation as much as possible by a content creator. For a playable, interactive resolution with high quality texture maps, this becomes all the more crucial.

4. Visual effects for virtual reality differ from visual effects used for regular desktop visualizations in UE4. In fact, many effects don't work and slow down the engine's rendering speed because of their screen-space. These visual effects work excellently for playable video or image content. This proves that the output has to be designed for from the get-go of working with an Unreal project.

5. Errors within a project are easily transparent to a user. A user in a VR scene is able to wander around and look at all angles of an object. Because they are encapsulated in an environment and their focus is directed to the screen in front of them, it is unavoidable to recognize mistakes. What a user can see on their screen really is what the user is getting in VR. With virtual reality, everything needs to be prepared properly with scale and any errors with the mesh or textures are going to be readily apparent to a viewer.

Conceptual Conclusions & Revisiting Research Q3

Creating options in virtual reality represents the opportunity for an enhanced design process with the ability to showcase iterations. It represents a better way to experience and visualize an unbuilt design due to the ability to test out a design-in context. By allowing a user to remain in a fixed position and push buttons on their controller to switch between environments, they are presented varying options in their field-of-view. This feature provides an unprecedented means for objective comparison. In an interactive VR simulation with options, designers have the potential of testing fabric, material swatches, space configurations and lighting scenarios within a user's field of view. To be able to activate presence, scale and effectively communicate options for a project, this experiment is representative of VR's capabilities as a powerful integration tool in the design process. At the same time, if an interaction is built ineffectively, the simulation will be no more effective than a visualization demonstrated on screen. However, as this investigation was constructed within a week's worth of documented hours with reasonable effectiveness, it carries the potential for any architectural designer to get started with ease. As such, efficiency in this process will increase as a user becomes more familiar with UE4.

A compelling virtual reality scene with options is therefore achievable within a week's timeframe. Both factors of efficacy and efficiency resulted positively from this experiment, indicating such. The learning curve, however, is an integral and continually ongoing feature of content creation with UE4. Prior to conducting the experiment, the timeframe range set for this experiment was 1 week (deemed short) until 2 weeks (deemed long). This parameter was lengthened from previous experimentation to account for the learning of a new software and account for the increased interactivity requirements with programming for this project. With the result of a 33 hr duration, an iterative VR project with options took the equivalent of less than an average work week to complete. In addition, the efficacy determining factors resulted in a positive response for numerous factors including interaction, presence and room-scale with the generous timeframe used for this experiment. As the duration is indicative of the achievable necessary effort, this experiment can be concluded with a favourable measure of practicality in the architectural design context.

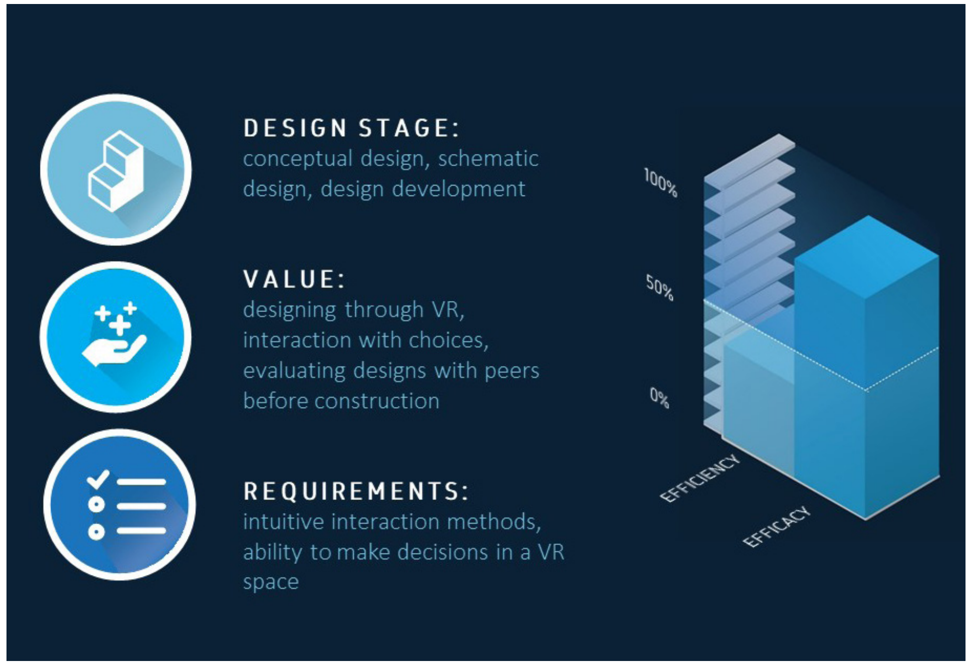


figure 7.17 Iterative VR Design: Applications

figure 7.18 Intuitive Interaction with Oculus Rift Controllers

Chapter 8: Evaluating Software Plugins and Push-Button VR

Research Q4: Are software plugins and push-button VR experiences more practical than game engines for the architectural field?

The objective of this experiment is to determine the practicality of VR using existing architectural software with the supplementation of push-button plugins. Architects, designers, and their clients have access to use tools like Enscape, Autodesk Live, InsiteVR and Tiltbrush to experience their designs in VR with little to no experience or knowledge of the technology. Many of these digital tools or integrations claim to be a one-click, or push button VR scene creation experience, which could transform the design process. The value of VR within push-button technologies will be evident in the speed of decision making, iterative applicability and timeframe efficiency. These tools are often marketed with the promise of enabling anyone to create visualizations without the necessity of learning any additional complex software or programming. Though VR technology is quickly evolving in terms of software options available, evaluating what these options currently offer can be beneficial in understanding the strengths and weaknesses of the current software climate. As it would also be impossible to evaluate every single program or plugin feature, only a few promising options were selected as they represented the most popular choices.

Evaluation of Process

Revit// AUTODESK LIVE:

Anyone using Revit can directly use Live through the direct integration. By clicking on start, the visualization is created directly from the built model by copying the 3D geometry, rendering it, and setting viewpoints where the camera is placed.[1] Flipping through views in the 3D model, the application can be exported as a PC or iPad application with the download of the Autodesk viewer from Apple Appstore/Autodesk Appstore to viewing "on the go".[2] With this software, it can be seen that efficiency is high, but efficacy is questionable, as the plugin does not provide a finalized walkthrough that a user can be guided by (as was attempted in Chapter 6). A user has to click to walk through it themselves as an application is made. The application simply facilitates "flipping through VR views" and navigation on the platform itself.

---

[1] Autodesk Inc., "Revit Live: Create Immersive Architectural Visualizations," accessed March 10, 2019, https://www.autodesk.com/products/revit-live/overview.

[2] Ibid.

A user (when in VR mode) can rotate themselves, teleport within their environment, measure distances and return to viewports. Navigation options thus only accommodate stationary experiences representing limitations within the scene. In terms of output and rendering quality, light is baked accurately within the environment, allowing for materials to look compelling. Yet, the premise of sole navigation and flipping through rendered scenes created (which didn't transfer camera views held in the original Revit file) made the experience less promising in terms of efficacy. As mentioned in the "Tactics of the Trade" section in Chapter 4, VR is too much of a democracy and requires settings to enable a designer to define what their user is meant to focus on within a virtual space. This visualization technique is therefore extremely efficient but might not be effective in conveying design intent. Additionally, the VR experience can't be created for every consumer VR device, making accessibility and interoperability a detrimental factor to the platform's efficacy. Autodesk Live can be applied within the design process effectively to create quick demonstrations for designers to "step into" their project and evaluate their designs in terms of materiality, textures and lighting.


1 Open a model in **Revit** and select a 3D view

2 In the **View** or the **Add-in** tab, click **Go Live** and your model uploads to the cloud

3 When finished, click **Open** and your model opens in **Live**

figure 8.01 Autodesk Live Home Screen and Project Views

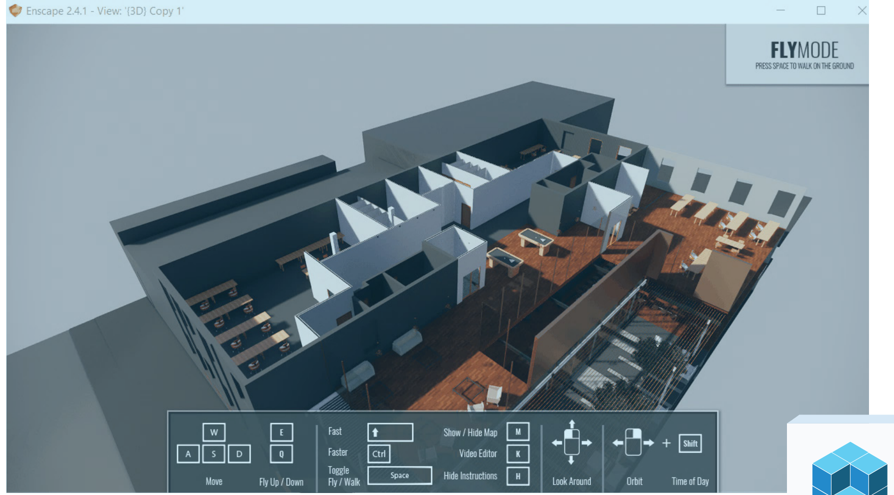


figure 8.02 Enscape FlyMode

Revit, SketchUp, Rhino// ENSCAPE:

Enscape is a Revit plugin that claims to be a push-button live project experience, with ease of jumping into a project without perspectives set up to easily experience a project in VR.[1] Similar to Live, it is a "fast-interactive" renderer which takes work created in one of the three architectural mediums and immediately outputs a visualization experience from the work already completed in the initial software program. According to the developers, "There is no need to carefully adjust your perspective before rendering, just start Enscape and fly to the spots you like. It's fully dynamic". [2] Enscape is a key add-in for Revit and Rhino workflow through to VR, as claimed by clientele. As a real-time rendering plugin, it lets a user orbit around a model, walk the ground surface (if topography has not been built in Revit) or teleport to specific areas of a project while wearing a VR headset. The VR mode contains two main ways of interacting with a model; flymode (stationary) or walkmode (room-scale). Out of these interaction options, the "walkmode" is rather disorienting as it creates a ghost-like flow of movement through the space, making the experience of the environment rather unnatural. Alternatively, the application is successfully viewable by teleporting with controllers from place-to-place. The speed of creating these interactions makes Enscape a promising application for the architectural process.

---

[1] Enscape, "Enscape: Real Time Rendering and Virtual Reality: Live Link from BIM to Virtual Walkthrough," accessed November 9, 2018, https://enscape3d.com/.

[2] Brandon Pike, "Game Engines for BIM- The Next Generation of Project Visualization," accessed November 9, 2018, www.midwest-u.com/wp-content/uploads/…/Pike_BIM-and-Gaming-Engines.pptx.

The application achieves maximum levels of efficiency as projects are developed through software integration. Although efficiency is extremely high, Enscape (alike Autodesk Live) might be hindered with regards to efficacy. Due to an un-customizable template of interaction and a sole navigation platform, the software might not meet the needs of a diverse range of projects. As every architectural project is different, and clients might not be able to perceive design principles through sole navigation, Enscape requires greater customizability in features to facilitate design immersion. It would be of interest to the platform to develop spatial audio, content reorientation features, animation toggles and object interactivity to communicate design intent effectively. This way, the program would be able to use all the immersive advantages of VR. In it's current state, the software is effective simply in creating an appealing environment with lighting and design styles a user can experience in VR. Different types of explorations are required to meet the needs of unique projects, depending on the scale, location, size and focus of the design. Though Enscape provides a high-quality resolution rendering standard, navigation is left to the user itself and baseline assets (from Enscape) not within the original model need to be added to the project to embellish the environment and achieve the marketed results of the application. Enscape also maintains an active live-link to the model, easing the process of iteration pre- and post- VR viewing. Enscape can apply to the architectural process as a means for developing a demonstration of a project that any design professional can immediately step into. The high visual appeal of the platform makes it a compelling medium to use to easily showcase designs. However, it might not be able to communicate design intent effectively. In this manner, Enscape might be primarily used in conveying lighting, visual appeal, materiality and general navigation of space effectively.

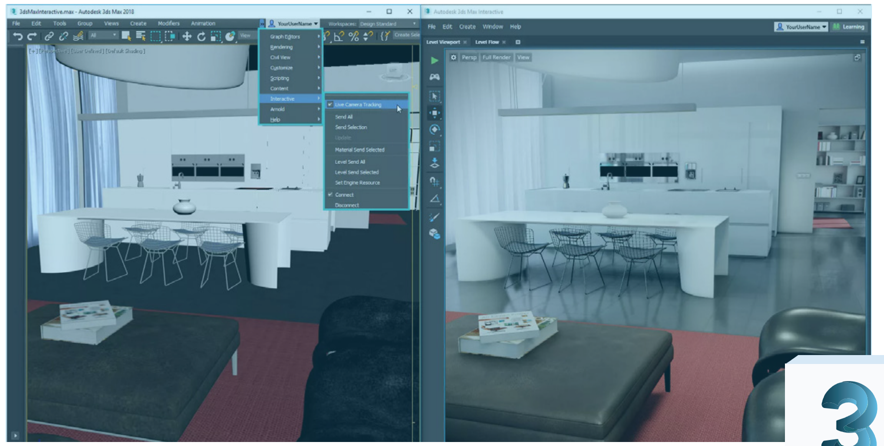


figure 8.03 Autodesk Live Home Screen and Project Views

3ds Max// INTERACTIVE:

Interactive acts as it's own virtual reality engine. Its efficiency was reasonably lower than its predecessor Live but between the gradient of push-button VR and game engines, Interactive lies in the middle. It takes a considerable amount of time to navigate but, similar to Unreal, has developed packages with necessary tools and scripts to aid non-developers to create mobile, PC and room-scale experiences.[1] Though the efficacy of experiences can increase by using Interactive, the steep learning curve is still prevalent. Where Interactive succeeds is in being a medium that improves the workflow process by accommodating iteration. 3ds Max data is linked directly into the engine to allow for an easier way to make changes within a project. 3ds Max Interactive is still not a perfect medium but might present a method of diving into developing with a programming mindset in the architectural field. Software interoperability becomes easily mediated with the use of Interactive, as model files of all formats can be input into the program. As 3ds Max is often an intermediary between game engines and architectural software, Interactive is a program that might present the best option for choosing between working with push-button VR and game engines. Although, it does fall closer to the game engine side of that spectrum, requiring the same principles to be mastered as when working with Unity and Unreal. Interactive more so facilitates content authoring. Additionally, as Max provides integrated 3D tools that surpass the capabilities of BIM tools, it provides a greater degree of customizability and can provide greater results than push-button plug-ins.

AutoCAD, Revit, SketchUp// INSITE VR:

InsiteVR is a platform to create an immersive screen-share of a project. InsiteVR is marketed as a product able to create meetings for design professionals to review models in VR.[2] The features of the platform include designating a lead presenter, using built-in voice integration, collaborative markups, synchronizing models to the cloud and enabling scaling and volume controls. The platform is designed to be suited to meetings to review projects, discuss design issues, direct attention to concepts and annotate or markup a project in real time. The efficiency of using InsiteVR is reasonably high as models can be reviewed in a variety of formats and exports are created with ease. However, some detrimental features noticed in investigation are the high price tag and slow speeds of loading and uploading projects.

---

[1] Autodesk Inc., "3ds Max Interactive," accessed March 10, 2019, https://area.autodesk.com/tutorials/series/getting-started-in-3ds-max-interactive/.

[2] Inc. vrban, "Insite VR: VR for Architecture, Engineering and Construction," accessed March 10, 2019, https://www.insitevr.com/.

In particular, renderings are not generated to perfection with InsiteVR; walls and materials glitched in certain project sectors and were represented incorrectly. The scale of objects sometimes also cut off, representing incorrect dimensions within a project. Though this medium couldn't be tested for efficacy in context of collaboration, the software did prove to effectively create a VR experience output of a design with the ability to markup and navigate through a project. InsiteVR is ineffective in providing a finalized visualization directed towards obtaining client buy-in for a project. Despite the low resolution, glitches and un-photorealistic quality generated by the platform, it can showcase projects successfully for the purposes or collaboration through design. As the AEC practice requires collaboration at every stage of the design process, the efficacy and efficiency of the medium in providing a tool that users can "jump into" and "converse within" represents promise to the profession. InsiteVR might have a special place in the design process to facilitate the markup and presentation of designs while architects coordinate a project collaboratively.

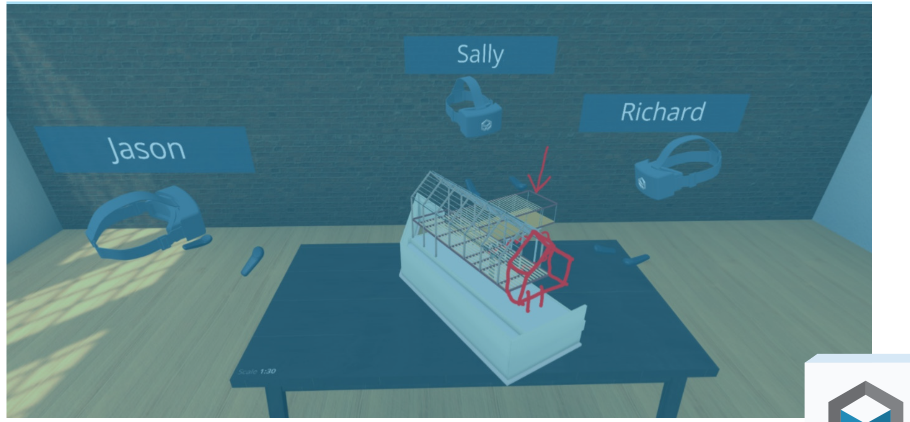


figure 8.04 Insite VR Project Collaboration

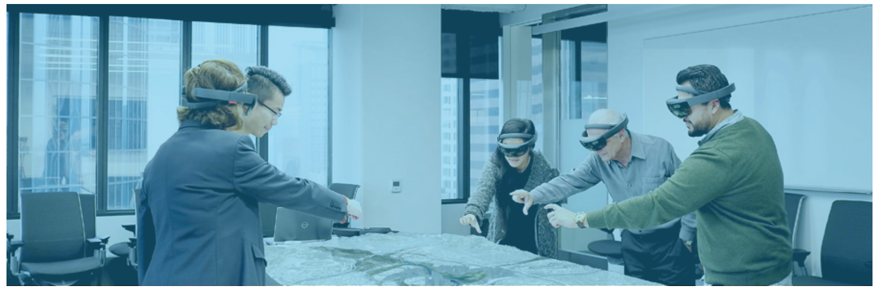


figure 8.05 Design Team Making Decisions with Microsoft HoloLens

Google VR// GOOGLE EARTH, TILTBRUSH, BLOCKS:

Google has released a multitude of software options which have the potential to revolutionize the architectural design process by adding the immersion of VR.[1] Google Earth is commonly used in the design practice to scope out sites of an architectural project and understand the landscape surrounding any specific location. With the implementation of virtual reality for Google Earth, a site can be explored with ease at a real-life scale. The platform allows a user to walk around, fly or browse common search locations in virtual reality. Similarly, the developers released a virtual drafting software and modeling software. Tiltbrush and Blocks allow users to digitally draft and sculpt forms in virtual reality. Though these manipulations might not be scaled adequately or be able to develop full architectural models/drafts from, they might have their own place to explore 3D space and provide new perspectives during the early design stages. The efficiency of using the multitude of Google VR software is relatively fast and simple. Although the graphics are less polished than originally expected, manipulations in VR using Google's tools are successfully engaging methods for conceptualizing projects.

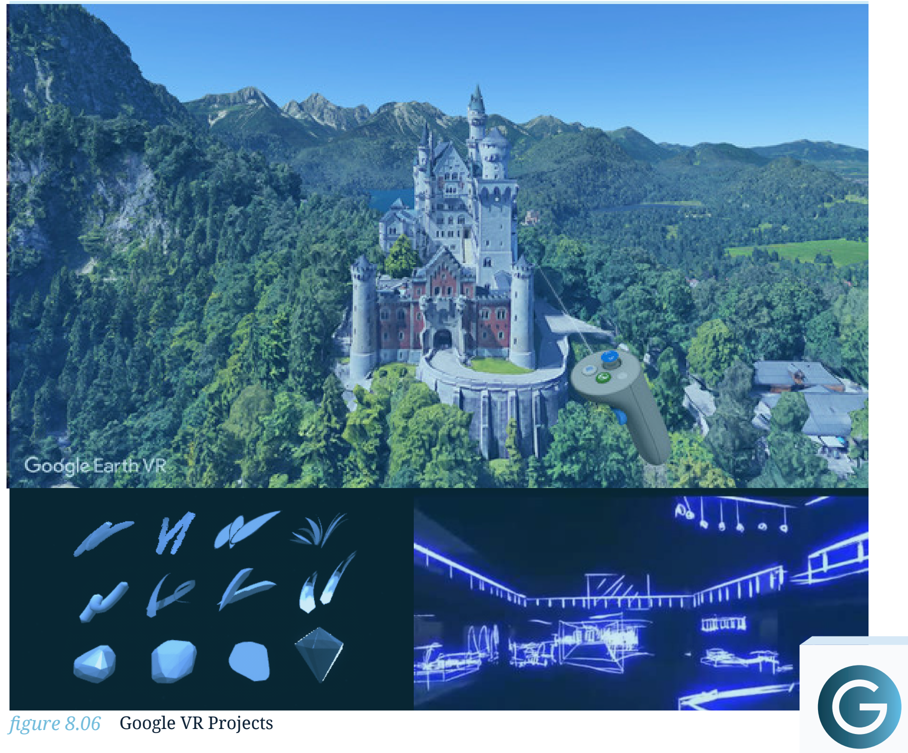


figure 8.06 Google VR Projects

---

[1] Google, "Google VR: Virtual Reality for Everyone," accessed March 10, 2019, https://vr.google.com/.

Limitations & Current Issues

Although a multitude of software platforms exist that can create a VR scene at the click of a button, the problem exists that these software application subscriptions are often expensive (ex. Enscape and InsiteVR), limiting their value for most designers, when they can explore the creation of virtual reality projects with game engines themselves. These VR experiences also are not customizable to meet the needs of diverse ranges of projects. Every architectural project is different, and clients or design peers also require different explorations in their visualization, depending on the scale, location, size and focus of their project. Push- button 3D Plugins enable anyone to create visualizations without the necessity of learning any additional software, though these experiences are often not customizable. In searching for the most viable option for taking an existing 3D model into a 4D scene, allowing for customization and iteration within the design process, Unity and Unreal Engine, or 3ds Max Interactive might be the best options. Unity and Unreal are game development platforms, which offer full control of an environment, its lighting and materials. As seen in the experimentation of the previous chapters 5, 6 and 7, they often have limitations resulting in their impracticality. They take time and effort to execute effectively. With the exploration of 3ds Max Interactive in this chapter, Interactive might be a better option for most architects than game engines, to offer the same level of customizability, yet ease into the programming features of design. It offers multi-platform compatibility, which is primordial for the average architectural designer who uses a range of software to meet their needs. Also, it maintains direct project integration through 3ds Max and requires a learning curve. Yet, the exploration of customizability and ease of availability Interactive apt for working with a diverse range of architectural projects. The use of other programs mentioned above, however, are not null. It can be seen that even though these applications have limitations, they still provide uniquely valuable applications for the design process. Each software can be seen to have its own place within the design process to complete specific tasks efficiently and effectively.

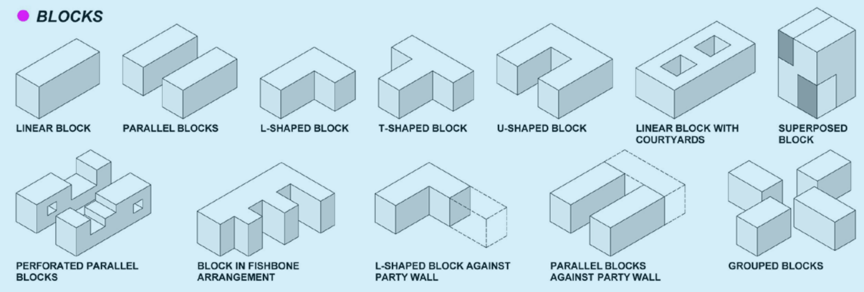


figure 8.07 Unique Design Typologies Requiring Unique Experiences

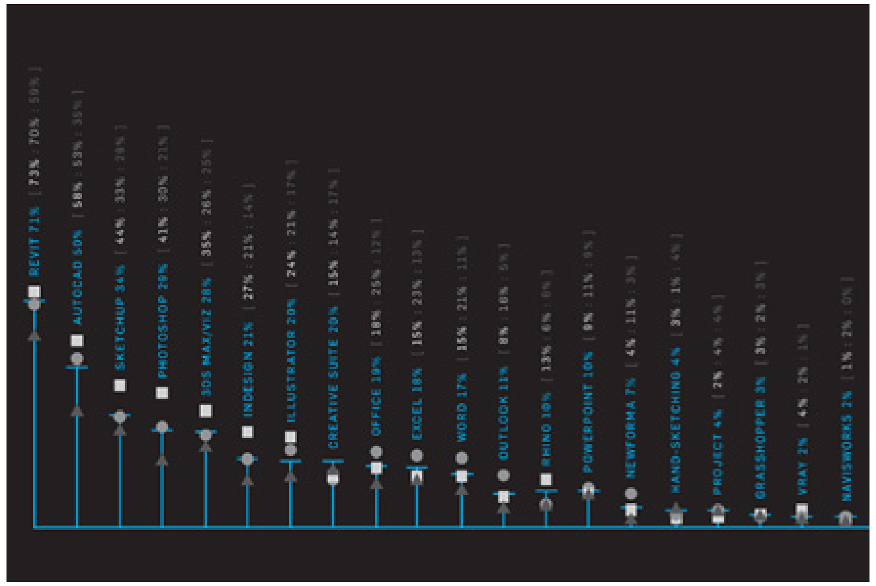
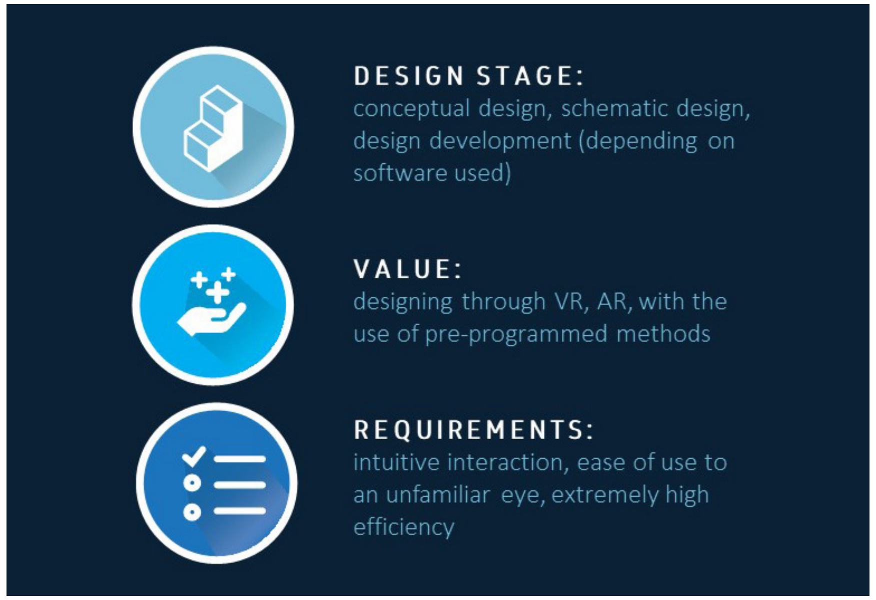


figure 8.08 Software Requirements for Working at the Top Architecture Firms

Conclusions

The evaluation of different programs resulted in high efficiency and relatively high efficacy, deeming most programs quite practical within the architectural realm. As noted previously, however, practicality does not equate directly to applicability. The software explored in this chapter is fast and easy to use, but not at every part of the design process. Architects can choose from differing VR applications to implement unique experiences at each design stage. Designers are able to pay for the output they require but this also negates learning through practice with VR, and can hinder the ability to customize an experience. Further consideration is required to gain an understanding of what programs are enhanced by VR and what programs might better suit other mediums of representation. Designers are urged to ask themselves whether something specifically needs to be created in VR and what value it provides the design process. Whether VR is explored through sketching, pushing/pulling forms, within a detail construction view, or a flythrough camera, the architectural field now possesses an added layer of immersion within a design project. Although these programs are restricted to specific output formats, templates, graphic levels and functionality, they can each be seen to have their own place within the architectural design process. No one software can achieve everything. Instead each software might be able to provide a practical and unique perspective by exploring space in VR.



DESIGN STAGE:
conceptual design, schematic design, design development (depending on software used)

VALUE:
designing through VR, AR, with the use of pre-programmed methods

REQUIREMENTS:
intuitive interaction, ease of use to an unfamiliar eye, extremely high efficiency

figure 8.09 Push-Button VR Design: Applications

Chapter 9: Discussion

Evaluating Previous Assumptions

Before the experimentation process began, it was predicted that the experiment would be able to confirm the practicality and ease of use of VR within the design field. Because VR has achieved successful implementation toward educational and medical disciplines, VR creators and media outlets make grandiose claims regarding the technology's potential. From these claims, it might be easy to assume that VR indeed does have widespread applicability and practicality. Yet, as noted in Chapter 2, the distinction between applicability and practicality lies within testing the ease of use of the medium specifically within the architectural discipline. VR is applicable toward a diverse range of visualization practices, as shown in experimentation, but, it may be practical only for a small subset of projects. VR experiences are much more difficult to create than they seem. VR has been proven to distinctly signal that of real life experience, and so it's integration into the design process comes hand-in-hand with complexity.

Each experiment conducted in this thesis as a novice programmer in the architectural design profession revealed the inherent difficulty of building experiences. For architects, designing an experience of space can be recognized as the basis of the profession. However, each representation medium for architects (i.e. drafting, model making, rendering) commonly focuses on a few specific features of a design, leaving the rest of the information to be extrapolated by a viewer's mind. With virtual reality, less of the experience of space is constructed by the human mind and instead directly showcased before one's eyes, which can be extremely difficult to create effectively and efficiently. This principle of difficulty in constructing a full experience for a user can be recognized from experimentation involving a steep, unexpected learning curve and major limitations of hardware and software configurations coming across in each investigation. Although these challenges are not deterrents to using the technology -as worthwhile tasks are often associated with difficulty- it can be understood that this revelation may discourage those with limited resources to exhaust their effort and time exploring game engines. For those that do, they might see that constant repetition of a difficult task might allow for active learning to take place, indicating a long term value of investing the time and effort to generate VR experiences.[1]

---

[1] Russell A. Dewey, "Psychology: An Introduction: Motor Activity: Learning Curve," accessed March 10, 2019, https://www.psywww.com/intropsych/ch07-cognition/motor-activity.html#learningcurv.

Evident in research, increasing the number of attempts at the process (as done in each experiment) resulted in an eventually greater performance measure and understanding of the technology behind VR. The process of experimentation, although exhaustive of effort and time, was able to provide a deeper understanding of VR's unique contributions to the architectural design practice.

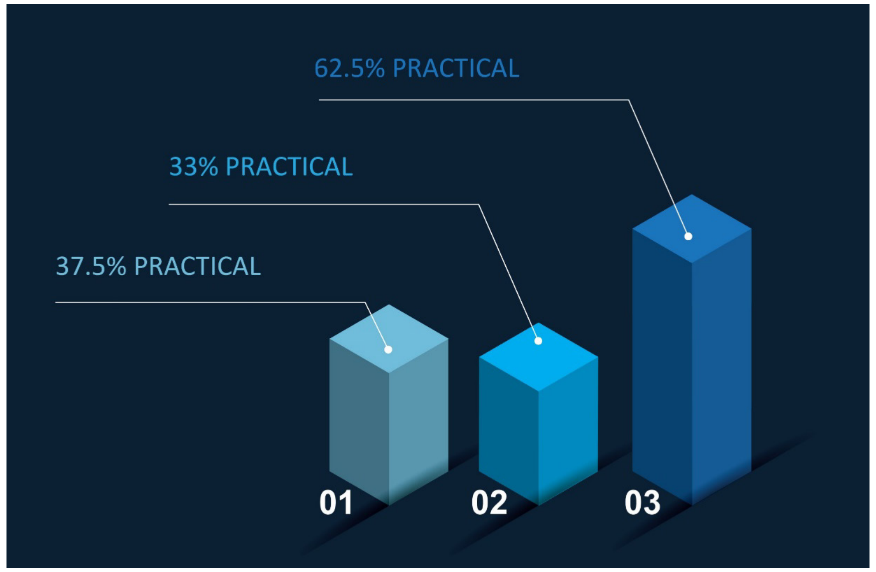


figure 9.01 Game Engine Experimentation: Practicality Comparison

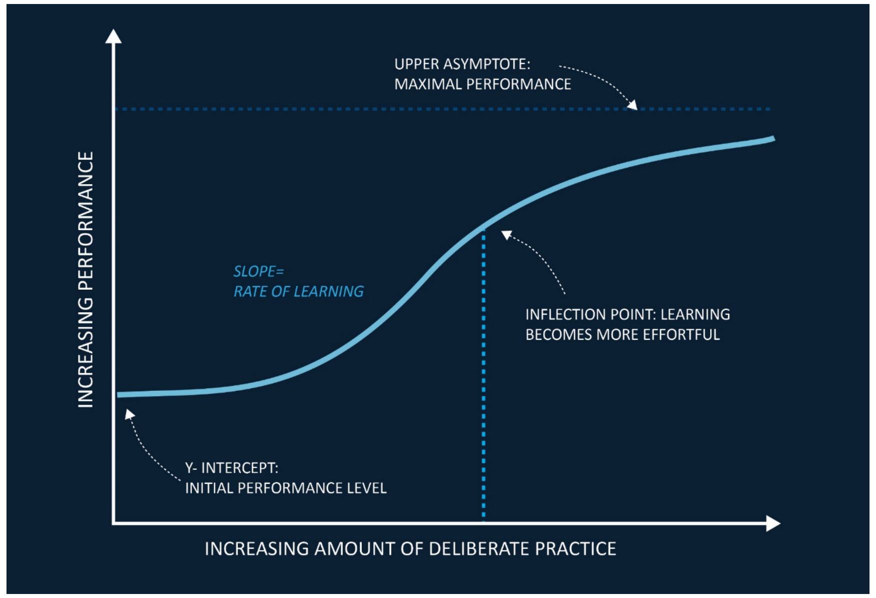


figure 9.02 Learning Curve

Efficiency & Efficacy

Tom Shannon, a visualization specialist who wrote the book, Unreal Engine 4 for Design Visualization, claims that developing with game engines requires the recipe of the right hardware, software and the right person.[1] This perfect combination opens up the possibilities for engaging 3D architectural models in different perspectives throughout the design process. Shannon also asserts that the greatest challenge for visualization artists and architectural designers is in acquiring the skillsets associated with developing attractive & interactive representations of projects. Even though VR hardware and software options are readily accessible to practitioners, developing the skills to utilize them effectively takes persistent effort and patience. The premise Shannon presents is reflected within the results of experimentation. Only if an implementation of virtual reality manages to get the combination of hardware, software and sensory synchronicity just right does it achieve that perfect sense of presence and scale, activating a user within their environment. As the results have shown, activating those features is a difficult task that requires a considerable amount of practice and time. Quantitatively, it took a combined total of 154 hours of work to design with game engines. As well, the three game engine investigations produced an average efficacy score of 6.7/10 or a success rating of 67%. Both these results indicate that even though a considerable amount of time was taken into the process of learning to design a project in VR with game engines, an even greater amount of time spent may have been necessary for more favourable results in efficacy factors. Although the efficiency is considerably high when working with software plugins and push-button VR, the visual quality and customizability were negatively affected, as mentioned in Chapter 8. Ideally, all experimentation cumulatively would have required a short amount of time to achieve a much more effective visualization result (i.e. the best-case scenario of optimal efficiency and efficacy initially set out). With the long duration and the efficacy rate that was significantly lower than expected, the research points to the flaws of the VR methodology. For future consideration, this represents the need for a more favourable workflow process to successfully translate an architectural project into a VR experience. Whether this involves improvements in hardware, software, or both; further research is required to determine a better workflow geared towards the architectural practice. Conclusively, this experimentation represents that design representation in VR is highly impractical toward the architectural discipline. However, even with this resultant impracticality, architects may recognize the long term value of investing time and effort to understand VR's niche place in design.

---

[1] Tom Shannon, Unreal Engine 4 for design visualization: Developing stunning interactive visualizations, animations, and renderings (Boston: Addison-Wesley, 2018).

Technical Conclusions

STRENGTHS:

The greatest strength of VR is that it engages designing within it. From the perspective of an architectural designer, this could mean that using game engines might activate a new strain of thinking through a project. Although this requires more research, it was noticeable during experimentation that the workflow contains similarities to the architectural process. The preparation of data assets, scene organization, application of materials and study of lighting adjustments occurs similarly to the way architects design using other visualization platforms. The primary means by which the workflow is similar to the architectural process lies within designing intentionally. From the very initial stages, the output is designed with significant consideration placed into understanding the visual impact on a user. For example, in each investigation, thought was put into scale, camera paths and views so that the VR experience could be driven effectively. Care was also taken into ensuring that a user jumping into scene was not placed into a jarring or uncomfortable environment. Considerable thought was placed into each design component of the virtual simulation because any user input represented design feedback. In this manner, VR is an honest form of representation, in the sense that “what a user sees is what they will get.” Further research is required, however, to reveal whether what a user understands from a VR space is exactly what they will understand from the physical manifestation of that space. This research would aid in understanding the distinction between VR, physical reality and phenomenology, given the promising communication factor of representing designs in VR.

WEAKNESSES:

The largest account of impracticality came troubleshooting difficulties in the workflow. It was noticed during experimentation that many steps were geared towards a trial-and-error process. Finding solutions to troubleshoot common issues experienced was problematic as information was not tailored to a specific solution at all times. Even though learning through tutorials was valuable, information quickly became outdated as software updates were made. If the solution to one problem was found if another error presented itself shortly, it would represent another road block in the design process. These road blocks accumulated to eventually represent limitations in hardware and software option configuration. In chapters 5 and 6 especially, the Internet had to be scoured to retrieve solutions to each unique problem, and this often took exorbitant amounts of time to configure at each occasion. Errors that presented themselves during simulation creation had to be resolved immediately because being encapsulated in an environment with directed focus to a screen placed in front of the eyes does not

allow room for any flaws. Glitches in a virtual space become easily identifiable to a user as they enter a simulation. If a user is also not able to activate interactive functionality successfully (by moving around, picking up virtual objects, teleporting, or following narration), the entire virtual experience can result negatively. Another weakness of designing in VR was the innate need for coding/scripting. It was necessary to learn the basics of code in multiple formats (as per the game engine); there was no manner of using game engines without coding or scripting, unless to create an environment without any interaction in any form.

HARDWARE LIMITATIONS:

The hardware requirements of running VR are quite challenging as a powerful computer processer and generous hard drive space are necessary (as mentioned in Chapter 4). Although it is useful to have the possibility of playback on a wide range of devices, this increases the complexity of project-to-device translation. This also hinders accessibility of the medium as the process of output is not always seamless (as seen with mobile headsets). Similarly, when an application is finished being designed, if it needs to be run on different hardware than its original output, the entire application will require modification, as the method of tracking and technical specifications of displays changes. All interactions would need to be modified as well to accommodate the final result. Finally, high-end systems can present the problems of requiring technical experience, on part of the designer, to be managed correctly. As an example, if the VR system has only a few trackers, and an avatar smaller than the user is required, it can be extremely challenging to extrapolate limb positioning to a virtual body.

SOFTWARE LIMITATIONS:

As soon as interactivity comes into play, a whole new level of difficulty presents itself. Interactivity is a digital conversation between a character and an object. Both characters and objects require input to be told what to do and output to be told how and when to do it. In this way interactions are built as programmed reactions. Programming is a requirement that most traditional visualization techniques do not warrant learning. Thus, most firms or studios in the architectural practice would not have experience with it. Unfortunately, it was discovered that even the simplest interactive visualizations require a level of scripting to facilitate logic into the behavior of the interface. Aside from programming, it can also be seen that the workflow of architectural modeling software into game engines presents compatibility issues. Specifically, these problems lie within the loss of information. For example, as seen in Chapter 6, Revit requires the use of an intermediary to translate information. This is indicative of the need for a seamless workflow involving file linkage and a more universally-applicable process for the preparation of a model file.

figure 9.03 Design Thinking in Virtual Reality from Game Engine Experimentation

Conceptual Conclusions

Discussing each experiment's results, application, overall practicality & design process integration:

1. STEREOSCOPIC RENDERS (CH5)

RESULT

Presenting a VR environment as a stereoscopic render might not be a viable option for most scenarios. The efficacy and efficiency of this process were both significantly low. A process that requires less time and yields a more visually appealing result is necessary for this method to be deemed practical. Stereoscopic renders are built with limited/no interaction and only the promise of visual immersion. The limited interaction immediately makes stereoscopic renders only applicable towards the final stages of design development in a project. This method can thus only be used to obtain client buy-in towards an expensive, photorealistic and fully-resolved design. If alternative methods of stereoscopic scene creation can achieve this with a fair resolution quality and the activation of embodiment, such tools may be more viable options to consider in the future. In this investigation, as a stereoscopic scene was designed from a game engine and administered on an accessible mobile headset, it went through multiple stages of image resolution down-sampling. This resulted in an unappealing end result, unable to match the high-resolution photorealistic baseline standard upheld for the architectural industry. This process was therefore extensively time-consuming and ineffective at photorealism to justify practicality. Currently, the method behind creating stereoscopic renders with game engines requires further research to mitigate these undesirable results and exhaustion of time.

THINKING PROCESS

Designing through perception:

While creating a stereoscopic render through the Unity game engine was impractical, the process did trigger a unique response of thinking through a design in virtual reality. The experience of a render in VR was distinctly different from that of other media experiences, which often only capture fragments of what the human brain can detect. The experience, though un-photorealistic, was able to place a someone into the rendered environment. When humans even move their heads left or right to experience an environment, their optic flow alters their field of view, their vestibular system sends signals of movement to the brain, and proprioceptive cues from the skin and muscles respond to their space.[1] Scientifically, this represents that people have temporal and visceral reactions to space, beit virtual or physical. This is important because it also denotes the fact that people have defined thresholds of physical space. Subconsciously, humans understand the way they occupy an environment. A personal perception of space is a contributing factor to understanding an environment's scale, lighting and interior/exterior conditions. The ability to step inside a render is useful within an architectural design process because it engages a direct perception of these factors. When looking at a render on screen, concepts pertaining to distance, dimensionality and spatial relationships are understood from afar. On the other hand, with VR, a viewer isn't separated from the render by a screen. This a user to think about their spatial relationship with their environment whilst in a virtual space. Designers might be able to see the merit in designing spaces by first viewing them in Virtual Reality as it has the potential to evoke a greater sense of perception and engage visual stimuli toward the environment in question.

Comparing a stereoscopic experience of a rendering to that from a screen, designers can recognize that renders are often used to misrepresent the features of a space. To enhance a client's perception of the render, Photoshop effects, filters, and wide-angle lenses are often used to give a false account of the environment and increase appeal. Though this feature is necessary toward marketing a project, it might place unrealistic expectations on the envisioned design in reality. Masking unappealing features and using unrealistic angles to an advantage are tactics that are not as easily implemented in stereoscopic rendering. Users are also less able to be given a false sense of space, making the room for bias and error much smaller. This also carries a direct impact towards features such as materiality and lighting, which might be able to engage visual stimuli to convey the texture of surfaces with greater clarity in VR. Users may be more inclined notice specific fixtures, furnishings or finishes in a stereoscopic rendering. While experiencing a render in stereoscopy, it can be argued that "what a user sees is what they will get." Even as the result of the investigation yielded a low-resolution rendering, the stereoscopic rendering experience was still able to convey texture, materiality and lighting conditions with an added level of depth perception, compared to looking at the rendered image on screen. Architects might therefore be inclined to think critically by using stereoscopic renders for their project, so that they are able to perceive every aspect of a render and notice features of a design which may have otherwise gone unnoticed, veiled by the barrier of a computer screen.

---

[1] Paul Mealy, Virtual & augmented reality for dummies, 1st edition (Indianapolis IN: John Wiley and Sons, 2018).

2. NARRATED WALKTHROUGHS (CH6)

RESULT

The premise of this experiment focused on designing the animations for a viewer in the creation of a VR walkthrough. This was facilitated with the use of teleportation, object movements, spatialized audio and the craft of timelines. One example of the animations created is in how sounds within the scene were designed to be spatialized (i.e. activated in volume depending on a user's movement towards and away from a sound source). To ensure the success of this virtual journey, a great deal of effort and time was placed into the coordination of a user's actions and their virtual environment's response. It was quickly realized while conducting this experiment that even miniscule technicalities can offset a user's ability to relax, listen, focus and appreciate the design that surrounds them. The overall efficacy of the experiment therefore hinged on the success of animation functionality within the VR experience. Unfortunately, this also resulted in a greater amount of time spent to troubleshoot the animations of the experience. The experiment also focused on achieving polished graphics alongside resolved animations. Although the environment achieved was not photorealistic, the strong narration and guidance through the virtual space provided the basis for a rather enjoyable experience. In this manner, it was understood that the goal with making effective VR walkthroughs in practice relies on the design of a coherent world, rather than a completely photorealistic one. Even beautiful environments can be unenjoyable without well-programmed animations of user interface elements. In hindsight, this lesson would have prompted less time to be spent on enhancing visual appeal of the walkthroughs. Yet, due to the exhaustive time and effort spent on animations to achieve successful walkthroughs, the experiment was indicative of impracticality.

THINKING PROCESS

Designing through animation:

The main concept understood from the process of using a game engine to create a walkthrough animation was how animations themselves are formed. The knowledge of this concept brought forth a greater engagement in designing virtual spaces through an understanding of the relevant properties of physics. Animations are formed with the use of objects, a controller and a script that connects the controller and the object with the use of a trigger function. These animations are each a part of their own game environment, nested within the overall game. These nested environments act on an endless timeline loop, which are triggered on and off by a user or first-person controller in a scene. These principles represent an understanding of space based on physical relationships.

Environments can be seen to act as cause-and-effect functions of force. A designer might therefore be able to understand physical and structural principles behind dynamic environments by creating animations in VR. In everyday life, architects often do not need to consider the weight of objects in their direct environment or the forces at play between these objects. These effects cause humans to regularly experience hitting a surface, either by tripping over a ledge or knocking over an object from a table. Subconsciously, there is no need to think about forces occurring real world when designing, because these forces occur naturally. However, when designing a space in virtual reality, force and the principles of physicality become important considerations. This engagement of actively thinking through spatial relationships by the principles of physics for animations can bring a greater understanding of spatial proximity.

Architects might therefore be able to think about a design thoroughly by animating the forces at play for a structure in real life. This ergonomic aspect of design carries applications toward the construction of environments that need to determine spatial affordances. Whether this comes from adjusting counter dimensions in VR, choosing edges to chamfer or organizing the layout of physical objects within a space, VR carries a deep relationship with proxemics (the amount of space people feel necessary to place between themselves and objects/others). Animation in VR might also be used to deliberately conceptualize movement through a design, by conducting passage studies and walkthroughs, as attempted in experimentation. Though the functionality wasn't activated effectively and movement through the space was difficult to coordinate, the choice of actions became much more deliberate when designing with VR. The placement of doors, windows and stairs to tactfully compare layouts in VR might therefore be unique worthwhile investigations through animation. Thinking through a design by animation may also impart the knowledge of spaces that feel too small against spaces that feel too grand. It might be used to offer an understanding of the dynamics of how one space connects to another within the grand scheme of a design. Similarly, a crowding study could be animated in VR to determine the comfortable amount of traffic for a user to be able to move through a design. These unique explorations might offer a conceptualization of ideas that can manifest themselves by animating through VR. Because still VR is very much a physical experience of space, a user's bodily dimensions directly impact their understanding of space affordance. VR can therefore be used to benefit the architectural practice by challenging these effects of proxemics using animation. If a design is meant to provoke a physical response to space in reality, it can be beneficial to have a VR simulation designed to encourage or test that physical response.

3. ITERATIVE INTERACTIONS (CH7)

RESULT

This experimentation focused on the creation of interacting with an environment to access iterations of a project. Although the efficacy and efficiency of this process were determinably high, the process involved an understanding of complex interactive principles. Successful interaction methods, which are well established within the gaming and user interface/experience (UI/UX) industries, such as moving around, picking things up, reading text, using menus and activating functionality, do not work simplistically in VR. It becomes important to know which interaction techniques to use within the design context. In light of this fact, a considerable amount of thought was placed into designing the interactions of this project because user input represents design feedback. It was also determined that interaction would be prioritized over visual appeal/photorealism in this experiment. Having an un-photorealistic environment in this scenario also lowered the expectation of perfection and resulted in a more consistent feeling of embodiment. Achieving embodiment allowed for a higher level of focus to be placed on facilitating response. As a user navigated through the environment, functions were activated to allow for them to switch between two interior spaces and ultimately choose between these spaces. These interactions were made by programming the functions of controllers and using directional navigation. While methods of creating interaction and required levels of programming knowledge vary between software, once interactions were built within a project, the experience offered an additional level of decision-making. This can be a powerful tool in receiving input and feedback on projects in the design field. Due to the iterative process of design, the manifestation of options through built-in modifiers can engage an entirely new method of designing-in-context. Interactions therefore become more powerful and relevant in practice, worthy of the effort and time spent on understanding their creation.

THINKING PROCESS

Designing through iteration:

Architects have used iteration to evaluate projects in almost every architectural medium. The opportunity to experience a space at full scale with VR has often been too expensive and impractical in the past. There is a certain threshold of technical expertise that comes into play to successfully recreate a space for virtual reality successfully. The feat of iteration in VR, though time-consuming and complex, far less so than the creation of a building itself. Buildings often take years to manifest and are both labor-intensive and expensive to create. An immersive perspective on iterations of a project is therefore extremely valuable to designers and stakeholders of a project. Designing through the visualization of a 4D experience is previously unprecedented. Now, this process allows designers to activate a physical experiences of space far before designs are manifested in real life. Designing through iteration is a powerful way to think through a project for architects as it presents the ability to interact with a project pre-conception. There are ways to access navigating the space and test out methods of functionality or usage of space in real time. As understood from this experimentation, any level of interaction, whether simplistic, crude or complex can activate the feeling of immersion. Immersion allows for iteration in VR to reach higher levels of thinking through a project. Architects are not generally able to comprehensively think through a project while interacting with it without VR. VR allows architects to consider functionality from a personal perspective. Their response to objects within their environment as they interact with a space might predict how people who eventually inhabit the space might interact with the space. The nature of VR to envelop a user within a design provides a strong foundation for predicting and testing interaction. In this way, architects can engage thinking through iteration with virtual reality, opening up the possibilities for experiencing designs that are otherwise far to expensive, risky or dangerous to recreate in real life. From an alternative perspective, for a design that is projected to be created in real life, testing out an experience of the space pre-conception might uncover new ways to challenge or improve the design.

A temporal experience of built space is unimaginable with any other medium than VR. Physical and digital models have always left iterations to be made from an outside perspective. In an investigation of iterating in VR, it was discovered that the experience of comparing layouts of space while enclosed in that virtual space offered a unique and objective approach to design thinking. By comparing two separate versions of a space by teleporting between them in VR, the environments were able to be replaced as a user remained still. This experience is unachievable in any other design medium or in real life, provide a new avenue with which to explore and discover design solutions. This navigation between iterations can allow for a niche way to compare/contrast separate ideas in a single design project. Architects might be able to test the effect or reaction of different spaces to determine what constitutes a more comfortable or successful environment. Spaces can be configured by changing factors such as; lighting, materiality, size, layout and positioning. To a user, the experience of a dark and small room filled with cluttered objects may feel more claustrophobic whereas, a bright, airy and expansive room might feel more freeing. Properties can be tested at different levels in iteration to better understand the design principles behind successful and compelling architectural environments. Taking iteration in VR one step further, rearranging the furniture of a single space in real-time might allow designers to formulate the best configuration of that environment. Iterations of a project can also be represented in VR to showcase an experience of a renovation. Similar to the experimentation, in an example of investigating two kitchen spaces, the ability to witness and choose between real-time options for backsplashes, countertops, cabinet materials or island placement might ease the process of decision-making. A tactile engagement with the changing conditions of space carries the ability to strengthen the design practice by improving the way architects make decisions. Multiple configurations of furniture, fixtures, finishes, lighting or even massing studies can be conducted in VR to determine the version that might work best for a user. Iterating in VR might not only benefit making design decisions from an architectural perspective, but be useful in gaining quality feedback from clients as well. It can allow for a differentiation between design configurations that make sense in theory and ones that do not work in true scale, facilitating a potentially more viable comparative evaluation from both clients and design peers.

4. "MICROWAVED" DEMONSTRATIONS (CH8)

In the multitude of software applications tested, it was determined that "push-button" applications were selectively effective. While highly efficient, most programs are practical for use in the architectural field but may not be applicable to every design project. Customizability or the ability to make substantial modifications in terms of output for many of the platforms tested became challenging. This led to the question of whether game engines might be the most applicable way of constructing and designing in VR to achieve customizability. Another limiting factor noticed in this investigation is that the cost to use many of the tested platform substitutes the high expense of effort seen in previous investigations. However, each application platform is evidently highly practical. Compared to game engines, where programming and scripting can quickly become significant expenditures in design budgets of both time and resources, the majority of push-button applications were exceedingly efficient and successful with regard to their designated mission statements. The primary limitation of software integrations and push-button plugins are that they were generally not found to achieve significant design communication, but rather an awe of virtual reality itself. The uncustomizability of VR experiences with these tools leaves architects to decipher what to specifically use each platform for. On the other hand, these applications do offer a way to access small, specific and simple VR features that might matter in the design of an architectural project.

Each application provides value to a specific design feature (e.g. Tiltbrush for VR sketching, Google Earth for VR site context viewing, Enscape for designing quick VR flythroughs). These programs can therefore often provide unique and quick solutions for specific parts of the design process, allowing them to be easily integrated accordingly. In the use of these platforms, architects are able to choose from a diverse range of software options to achieve unique VR outputs from each program. Design practitioners can pay for the output required with the many software options presented in experimentation. Unfortunately, this also negates learning through practice with VR and can often hinder the ability to customize an experience. On the other hand, being unencumbered by a learning curve might allow these applications to engage designing through intuition, by allowing architects to make quick decisions within a project. Further research would be required to understand the role of design thinking in push-button applications and direct software integrations.

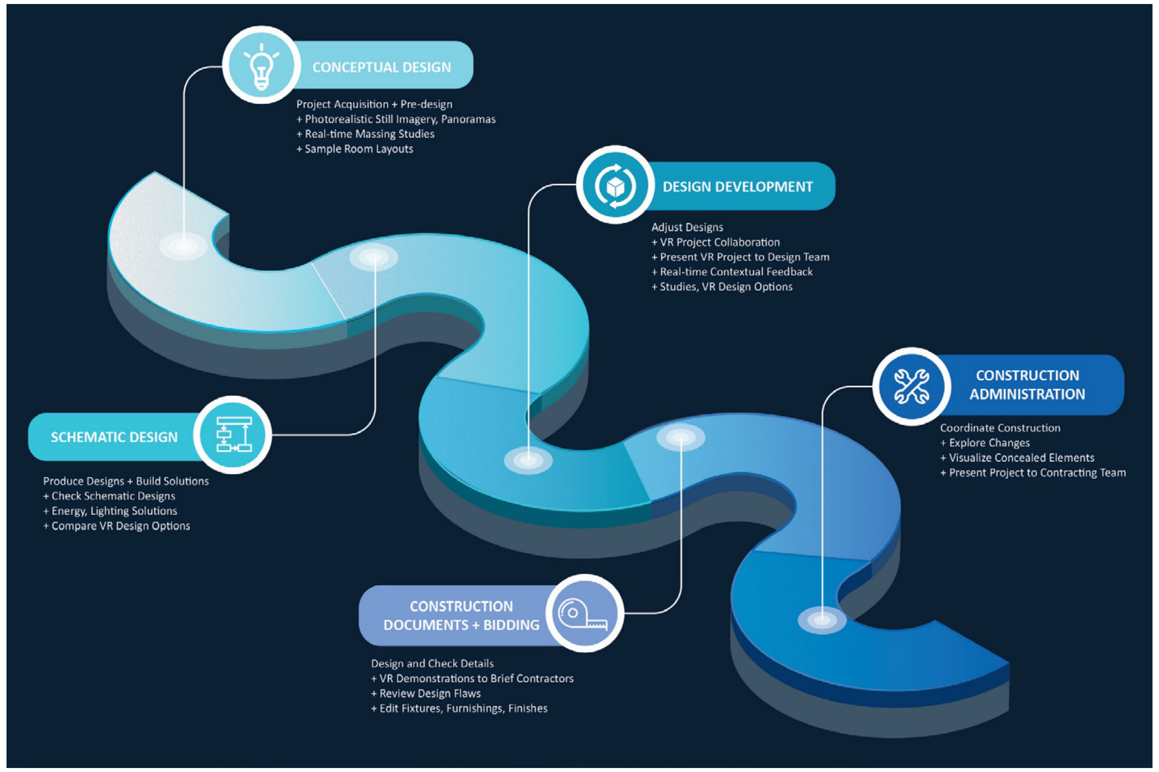


figure 9.04 Applications for VR at Each Design Stage

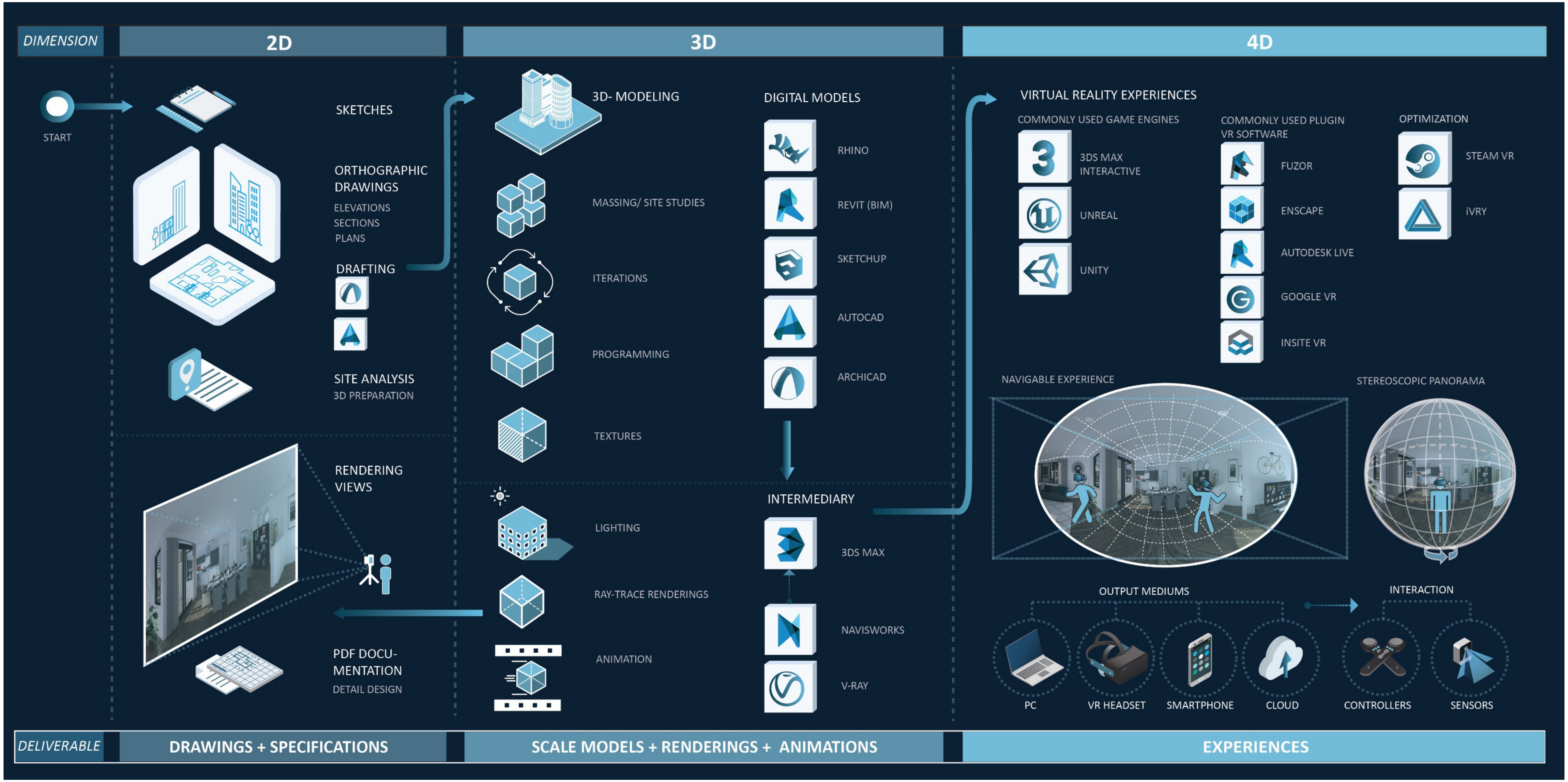


figure 9.05 Virtual Reality in the Architectural Design Process

Further Research

There are explorations already being conducted to determine the impact of VR towards the architectural design practice. German interior design manufacturer Dornbracht designed a VR installation to examine the effects of the technology in changing the way people interact with water. Their installation features the use of a bucket and hose which through an HMD, appears as a marble basin that releases a stream of small twinkling shapes. In another research study, conducted at the University of Waterloo School of Architecture (previously referenced in Chapter 2), the effects of spatial perception in VR were examined. Participants in the study conducted exercises to compare properties of VR spaces to orthographic drawings in features such as spatial volume, complexity and population. In each exercise, participants were asked to position the walls and ceilings of rooms to match a given set of dimensions. Both of these investigations showcase the fact that explorations with VR are seeking to activate design thinking. Both of investigations, although extremely insightful, might not directly represent where VR can uniquely be applied within an architectural design process. Viewing water as luminous droplets in VR might not be easily applicable to the human perception of materials in the real world. Similarly, adjusting walls in VR also might not provide value within the design context of the real world as much as manipulating more tangible factors such as counter dimensions. There is a need to objectively look at specific, isolated design features in to uncover the impact VR has towards the architectural field and pinpoint its direct placement in the design workflow. From the extensive documentation of this thesis referencing the effort and time it takes to create a successful experience, architects are urged to decide how VR can work within the design process.

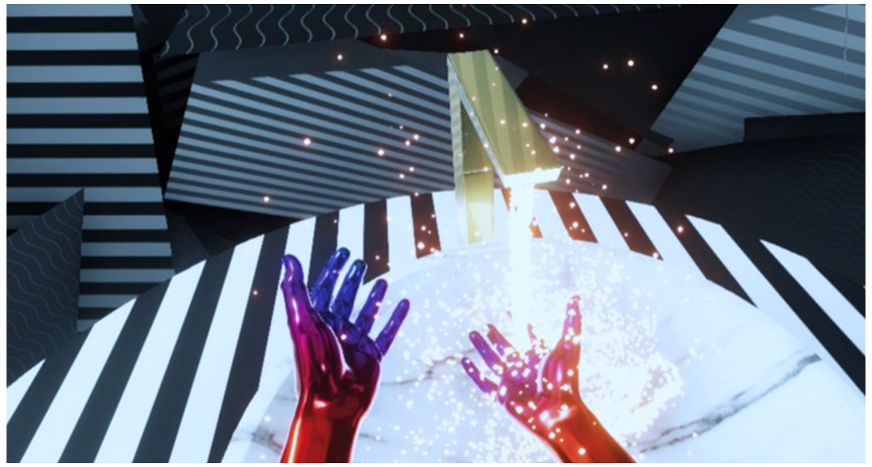


figure 9.06 Dornbracht's Milan Installation on VR and Water

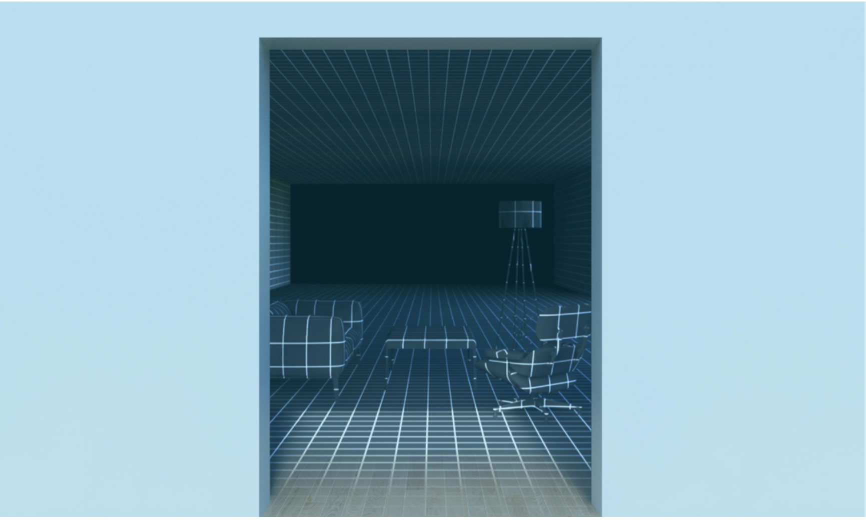


figure 9.07 Spatial Perception in Virtual Reality

This evaluation needs to be constant; designers need to be continually thinking about the medium for what it can exactly it can provide to enrich an architectural design. Deciding what makes sense to create in VR and what doesn't therefore, requires further experimentation and consideration. This thesis proposes that while VR may be too difficult to utilize as part of the final visualization stages to demonstrate hyper-realism, it carries a much greater potential toward early design stages of conceptualization. Towards this principle, hopefully this thesis is able to offer a differentiation between architectural VR design tactics and pinpoint areas of the design process where architects can implement tactful experiential design as they develop a project. This research requires further consideration, both to determine how architects can convey designs better with VR and how VR can uniquely be advantageous over other design mediums.

Future consideration could build upon the experimentation of this thesis and additionally evaluate the use of VR and an opposing architectural medium at each specific design stage for a project in at least three scales. Such an investigation might provide definitive conclusions surrounding the VR design workflow, in a comparative assessment of each scale and stage of the design process specifically. The research of this thesis also focused on understanding the workflow as a novice programmer, with an architectural education background. Exploring a diverse range of accounts of VR utilization from multiple design professionals would also be a worthwhile investigation. Research with a range of explorations and multiple individuals would provide a more conclusive account the of both VR strategies and the definitive average skill level of practitioners.

Expected technological advancements have the potential to increase the ease and accessibility of VR. Architects might be encouraged to explore these technological upgrades and might discover other niche implementations of VR in the design workflow as a result. The advancements of future HMDs might mitigate some of the effort and interoperability concerns experienced in research, warranting further investigation. These devices could allow changes in the range of motion affordable to VR users, making the technology even more enticing to architects in a discussion of spatial experience. Future headsets with greater specifications could also increase the complexity and prevalence of problems already determined in this document. Although these advancements have been released at the time of writing, further investigation into the latest technology (by re-conducting experiments in this thesis) might provide worthy insight into the challenges VR still might face. This information could draw conclusions toward what developers might need to rectify for future integration within the design practice. Alongside future HMDs, general advancements in technology might aid in the investigation of VR's place in the design practice. Supplementing VR equipment with wearable biofeedback and neurofeedback equipment (i.e. EEGs, GSR, and PPG) can better recognize the human relationship to a virtual space. These devices each measure electromagnetic brain activity, skin conductance from stimulation or monitor heartbeats. Although some devices are more accessible than others, they might provide additional information into what constitutes a stimulating virtual experience of architecture and therefore, be necessary to showcase in VR. Experimentation conducted in the future with these technological advancements can therefore increase the clarity of VR's role in architectural design.

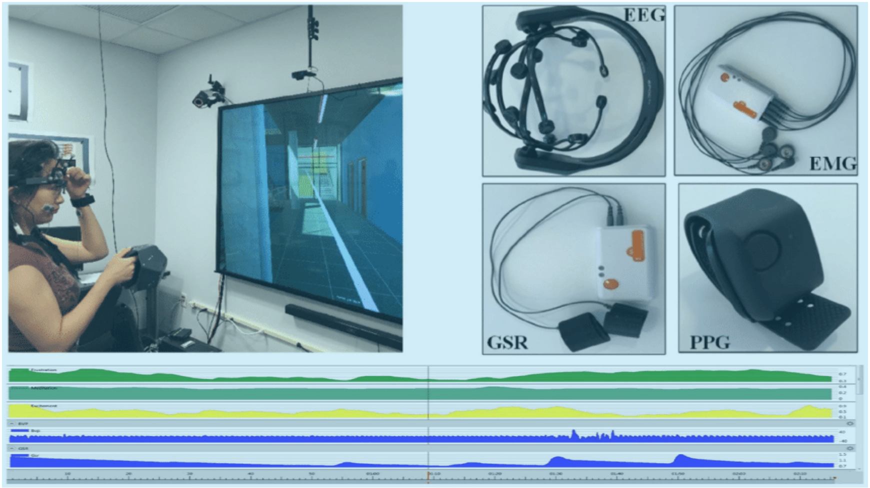


figure 9.08 Equipment to Monitor Bodily Conditions in VR Simulations

Chapter 10: Outlook

A Prospective Future

How do we experience space in new ways and how do we create successful systems that aid in the design process? With the prospects of the future "liquid" revolution of visualisation, architects are urged to participate in working with VR to understand its place in design. With each design tool in the past offering new ways of thinking about or imagining space than before, we can consider what VR provides conceptually. Hand-drafting helped us understand distance concepts and formalize relationships plan-o-metrically and sectionally. Model-making, allows us to manipulate 3D forms as puppet-masters and orbit spaces with high effectiveness. Each design medium can thus be seen to provide an added layer of information about a project. So, how can the experience of VR shape the way we design within the architectural practice? How can it allow us to conceptualize using a new strain of thinking? Even though the research within this document might not be able to derive definitive neuropsychological conclusions, it promisingly provides multiple ways to access thinking through a design project while working with Virtual Reality. This thesis originally sought out to define the practicality of the medium and engage an understanding of the role VR can play within the architectural design process.

Though the research conducted resulted in mostly unfavourable results for the practicality of the medium, experimentation asserts that VR can still provide design value. VR as an impractical technology can still be applicable to the design profession. Each experiment posed challenges with efficacy and efficiency, resulting in low evaluations of overall practicality. Some experiments were also proven less successful than others in conveying an architectural design. However, each experiment provided unique insight on the ways to think through a design project. Evidently, each experiment allowed for thinking through perception, animation and interaction, respectively while creating a VR application with a game engine. The results of experimentation are thus an activation of greater ways to think through a project with VR as medium. This proves that applicability might be more valuable than practicality with the technology. The impracticality of the medium might deem VR as successful but niche technology, with it's own unique place in the design process. It can be recognized that as the technology develops and architects become more familiar with the medium, the scope of VR usage within the design process can only expand. The applicability and practicality of the medium can therefore substantially improve as time goes on.

figure 10.01 Oculus Quest

The Next Best Thing

A tech-based focus group of investors has predicted VR to be a mainstream technology equating to the worth of sixty billion dollars within the next decade.[1] This indicates an excitement for the technological innovations of VR that are quickly developing. Critical speculation has brought forth emergent patterns for future headset options. The notion of the future "perfect headset" features a standalone, self-contained and high-resolution HMD with the untethered ability to track any environment with six degrees of freedom and marker-less inside-out tracking (without the need of sensors).[2] This is also with a wide field-of-view able to be tracked with the same six degrees of freedom.[3] Without physical limitations, greater freedom in movement can allow users to better understand their virtual environment. As these standalone headsets that offer a more frictionless experience are emerging, we can bear in mind that mass adoption of release to the consumer market requires a balance in both performance and price point. The promise of fast-emerging technology doesn't always equate to the same speed in the adoption of said technology. That is to say, a hardware innovation might release, but as the Hype Cycle predicts, products only gain traction after a considerable amount of time as people learn to use the new release.[4] With the direction VR hardware is taking in the near future, current hardware limitations could be very well diminished with the solutions expected in the horizon. It would be of interest to attempt this investigation of practicality once more as hardware changes progress to the consumer release of the "next best thing." An upgrade

---

[1] Paul Mealy, Virtual & augmented reality for dummies, 1st edition (Indianapolis IN: John Wiley and Sons, 2018).

[2] Ibid.

[3] Ibid.

[4] Gartner Inc., "Gartner Hype Cycle: Interpreting Technology Hype," accessed October 18, 2018, https://www.gartner.com/en/research/methodologies/gartner-hype-cycle.

in hardware might be able to ease practicality in the use of VR from it's current state. As drawbacks diminish, the speed of content creation might grow faster, and the efficacy of VR content creation might become even more prominent. However, it is also certainly possible that VR might never achieve the technological ubiquity or prevalence of smartphones. VR might very well exist primarily as a niche technology, while offering an incredibly diverse range of applications.

It is theorized that if enough architects can see the value in using VR, there would not be a need to convince them to build with it; it would simply happen. Even though the hardware has been released since 2016 and VR software has been releasing to date, most designers are unconvinced to build or design with VR, possibly because it is challenging and tedious. As this experiment can attest, from quantifying efficiency against efficacy, efficiency has consistently yielded low results for content creation via game engines. The onset of greater hardware configurations may very well ease content creation for the architectural field. Above hardware, it is the process of workflow that presents greater constraints than hardware limitations. It is much more important for designers to focus on becoming educated in the process of creating VR experiences with familiarity of game engines. This is because it is uncertain if there will ever be a release of the "perfect software" that resolves the difficult learning process associated with game engines while maintaining customizability. Architects are thus required to rise and face the challenges of using an unfamiliar mode of content creation to make use of the ample value it can provide. It would be interesting to witness how the results of this experimentation could change as different professionals in the architectural practice approach content creation for VR. There will always be technological innovations that supersede previous innovations, both in hardware and software for VR. Yet, what is of importance is how architects and designers navigate thinking through the medium as these changes present themselves. Architects should be empowered by VR software to design and create their own custom experiences. VR as a medium can only be as capable as the designer. Philosophers Kapp, Rothenberg and McLuhan attest that the practical uses of technology can only be an extension human action, perception, and cognition.[1] Technology is thus an extension of our human faculties. This represents the crucial fact that architects hold the power to use VR to create within the boundaries of only their design imagination. It is uncertain whether VR might spur an architect's ability to design better. However, the technology could quite possibly add a level of perception to the way they design. As seen in this experimentation, architects might also be able to improve their representation of designs, by designing and thinking through creating a VR experience.

---

[1] P. Brey, "Technology as extension of human faculties," in Metaphysics, Epistemology, and Technology, ed. Carl Mitcham (Jai, 2000).

Designing for the Senses

The future holds promise in offering high end systems that may increase the way humans interact with virtual space. One of the most immersive applications ever designed is the Verdun 1916 Time Machine which fools many senses at a time. Users would enter the simulation as wounded soldiers, and would interact with the environment by rotating their head.[1] During this interaction, users would receive sights, sounds, smells and haptic feedback from to the environment. Even though there was an extreme limitation in interaction, users in this scenario rarely experienced any lapse in presence. Some individuals of this experiment were even seen to respond to a virtual avatar by smiling at them. This case demonstrates to designers that above intuitive interaction the senses may play an even bigger role in uplifting presence. Similarly, another study conducted in Vienna, Austria attempted to determine if VR was "emotionally arousing" by creating mood-inducing procedures with virtual reality spaces.[2] Five virtual environments were created representing joy, sadness, boredom, anger and anxiety. The 120 participants were then assigned to one of the five virtual environments. The results of the experiment showed that almost every virtual environment was able to elicit the emotion intended on a user; VR was able to connect to the human senses by inducing reactions from the human vestibular system. Proprioceptive cues were activated from the skin and muscles while users were engaged in the environments. This scenario showcased another successful measure of VR conveying emotion in an engagement of the senses. Sensory perception can often be forgotten in designing VR experiences. Nevertheless, the design of spaces for all the senses in VR can affect an architect's ability to persuade others. Proven in the two laboratory experiments discussed, the human sense of perception is interconnected with emotions, which impact the way humans make decisions. With further research, an assessment could be made of how to use VR to determine moods in spaces. It would be of interest to designers to use VR to determine how people feel towards spaces and how those feelings can influence human action.

VR also presents an opportunity to go too far with the senses. A 2015 study in the Multi-Sensory Research journal represents how feelings of unreality and post VR disassociation or disorientation can be triggered by contradicting sensory input, akin to as one experiences in a VR headset. VR user Tobias van Schneider claimed, [when he takes off his VR headset] "what stays is a strange feeling of sadness and disappointment when participating in the real world, usually on

---

[1] Alexis Zerroug, Alvaro Cassinelli and Masatoshi Ishikawa, Spatial coverage vs. sensorial fidelity in VR (2019).

[2] Anna Felnhofer et al., "Is virtual reality emotionally arousing? Investigating five emotion inducing virtual park scenarios," International Journal of Human-Computer Studies 82 (2015), https://doi.org/10.1016/j.ijhcs.2015.05.004.

the same day. The sky seems less colourful and it just feels like I'm missing the 'magic' (for lack of a better word)…I often feel deeply disturbed and often end up just sitting there, staring at a wall."[1] Safe to say, this study was conducted on individuals using VR consistently for extended amounts of time, which differs heavily from the norm. What can be taken away from this is that it is important to remain wary of altering human senses with the design of VR spaces. That is not to say that architects need to fear the creation of sensory reaction in VR environments, but that the senses are fragile and necessary considerations for consciously designing with VR. The current ability to formulate tactile spaces offering engagement of all the human senses is far too complex, difficult and costly. The highest range of VR systems are admittedly the most expensive and labour-intensive. Computing the direct perspective of a user based on their direct position and engaging all their senses while doing so is an enormous feat. There is a long journey ahead for innovating VR formulate a full sensory encompassing experience of virtual space that can exist outside of a laboratory. We might not be anywhere near that goal of completely portable reality simulation for decades, but luckily VR can already be seen to impact certain senses with intuitive interaction. Even without being able to activate all the senses, VR can be recognizably beneficial to design representation.

figure 10.02 **Advanced VR Making Gains Towards Full Immersion, Comparative to Ready Player One**

---

[1] Rebecca Searles, "Virtual Reality Can Leave You With an Existential Hangover," accessed December 6, 2017.

Representation Revisited

VR enthusiasts often claim that it has the “ability to form an empathetic global social space” in the context of “revolutionizing how humans live, play and connect with one another.”[1] The justification of claims such as this might not be verifiable in the immediate or foreseeable future. However, it is easier to recognize VR in the manner scientist Jaron Lanier attests to, as “the technology of noticing experience.”[2] The commonality in both of these claims is that VR generates a translation of experiential information. The architectural practice has long sought after finding the best way to communicate experiences. VR can be seen to achieve this by utilizing what psychologists define as embodied cognition.[3] Cognition might be generally thought of as a process occurring only in the mind; whereas, in fact, other organs in the human body influence cognition as well. Muscle movements and a sensory experience affect cognition and aid in the human understanding of the surrounding environment. Other mediums commonly used to convey information often only capture fragmented experiences of what the human brain can detect (as determined by neurological studies conducted of the motor cortex). As VR environments can inspire embodied cognition to take effect, they represent a fitting medium for architects to explore the experience of space.

Virtual reality can be understood as an “experience generator” because anything architects can imagine seeing or hearing can be translated into the digital environment using VR.[4] However, this can sound terrifying and many people can be easily alarmed by the prospects of VR alongside the ever-increasing prevalence of technology. This is especially prominent when watching films such as Ready Player One, which indicate the exponential increase of virtual interaction in our daily lives.[5] The time spent online consuming content or creating it only seems to grow for most individuals.

---

[1] James J. Cummings and Jeremy N. Bailenson, “How Immersive Is Enough? A Meta-Analysis of the Effect of Immersive Technology on User Presence,” Media Psychology 19, no. 2 (2016), https://doi.org/10.1080/15213269.2015.1015740, https://vhil.stanford.edu/mm/2015/cummings-mp-how-immersive.pdf.

[2] Jaron Lanier, Dawn of the new everything: Encounters with reality and virtual reality, First edition (New York: Henry Holt and Company, 2017).

[3] Jeff Thompson, “Embodied Cognition: What It Is & Why It’s Important,” accessed March 1, 2019, https://www.psychologytoday.com/ca/blog/beyond-words/201202/embodied-cognition-what-it-is-why-its-important.

[4] James J. Cummings and Jeremy N. Bailenson, “How Immersive Is Enough? A Meta-Analysis of the Effect of Immersive Technology on User Presence,” Media Psychology 19, no. 2 (2016), https://doi.org/10.1080/15213269.2015.1015740, https://vhil.stanford.edu/mm/2015/cummings-mp-how-immersive.pdf.

[5] Ernest Cline, Ready player one, First paperback movie tie-in edition (New York: Broadway Books, 2011).

figure 10.03 VR in Design Communication Demonstrations

Yet, contradictory as it may sound, what if VR is used to deepen the appreciation carried for physical space? If architects can use the medium to communicate ideas, VR might enhance the human connection with physical space. By accessing the medium to communicate ideas of space, VR might be able to push the limits of what has been achieved in the past. Architects might be able to design spaces offering greater physical and mental engagement in reality, strengthening the human connection to the real world. With the variability in designing a custom experience, architects can approach working with VR conscientiously and decide when to take a step back from the medium. Arguably, working with the technology might inspire an appreciation for other design tactics, strengthening the interdisciplinary nature of the architectural practice. For VR to truly manifest influence within the practice, it should be uniquely woven into every stage in the design process. It is crucial for design peers and project stakeholders to experience what VR spaces communicate, as early as they are designing them. Meanwhile, it is also equally as important for designers to question if and why a space needs to be represented in VR. VR is neither good nor bad, it is simply a tool; a medium for which experiences can by communicated. Architects have the remarkable opportunity to decide how to communicate in VR and this decision can shape the prospects of the medium.

From the experimentation within this thesis, architectural designers might be able to recognize the long-term value of investing time and effort into understanding VR’s place within the architectural design process. After all, communication is the basis of architecture, and exists as the means by which the design practice contributes to the physical manifestation of buildings.

Architectural designs need to be presented in a format that can be generally understood, to allow the conversation between designers and stakeholders to become much more confident. As design practice focuses on crafting experiences of the built environment, VR can aid in communicating that craft. This can be seen in the investigative conclusions of this document, demonstrating how VR can be used to uniquely represent spaces in a multitude of visualization formats. After all, visual immersion is a factor architects should not forego, as it can bring a design project to life before a viewer's eyes, quite literally. While other mediums provide an abstracted vision of a design scheme, VR might successfully represent a project more believably. Utilizing VR uncovered the realization that designing with intent is not specific to the general architectural workflow; it applies to the creation of a virtual experience of space as well. Bearing this in mind, VR can be limiting if poorly executed, but a powerful tool if used effectively. It takes a great deal of effort and time to produce a highly detailed experience for the final stages of design. While advancements in the technology are awaited to mitigate its difficulty, in it's current state it is greatly suited to being a representational tool worth exploring for architectural design. VR can find its place as a niche medium that engages a more complex level of thinking through a project, increasing the capabilities of architects in visual communication. In the future, VR may strengthen its footing in the architectural discipline and eventually elevate itself to become a technique or necessary tool of the design trade.


figure 10.04 The Virtual Reality Worlds of Ready Player One

Bibliography

2016 IEEE Symposium on 3D User Interfaces (3DUI). IEEE, 2016 - 2016.

Abanyie SA, Botchway E. A. "The Impact of Computer Aided Architectural Design Tools on Architectural Design Education. The Case of KNUST." Journal of Architectural Engineering Technology 04, no. 02 (2015). https://doi.org/10.4172/2168-9717.1000145.

Abrash, Michael. "Welcome to the Virtual Age." Accessed October 24, 2018. https://www.oculus.com/blog/welcome-to-the-virtual-age/.

AEC Magazine. "Virtual Reality for architecture: a beginner's guide." Building Information Modelling (BIM) technology for Architecture, Engineering and Construction. Accessed October 24, 2018. https://www.aecmag.com/59-features/1166-virtual-reality-for-architecture-a-beginner-s-guide.

Arch2O. "8 Architectural Design Softwares that Every Architect Should Learn." Accessed October 20, 2018. https://www.arch2o.com/architectural-design-software/.

ArchiCGI. "Architectural Walkthrough: How to Sell Your Project Without Words." Accessed January 2, 2019. https://archicgi.com/architectural-walkthrough-for-selling-projects/.

Architizer Editors. "VR Is Now More Accessible Than Ever for Architects." Accessed March 3, 2019. https://architizer.com/blog/practice/tools/vr-yulio/.

asteins. "SmoothMouseLook." Accessed October 10, 2018. http://wiki.unity3d.com/index.php/SmoothMouseLook.

Augustin, Sally, Cindy Coleman, and Neil Frankel. Place advantage: Applied psychology for interior architecture. Hoboken, N.J.: Wiley, 2015.

Aukstakalnis, Steve. Practical augmented reality: A guide to the technologies, applications, and human factors for AR and VR. Upper Saddle River, N.J.: Pearson Education, Inc, 2017.

Autodesk Inc. "3ds Max Interactive." Accessed March 10, 2019. https://area.autodesk.com/tutorials/series/getting-started-in-3ds-max-interactive/.

Autodesk Inc. "Revit Live: Create Immersive Architectural Visualizations." Accessed March 10, 2019. https://www.autodesk.com/products/revit-live/overview.

Bailenson, Jeremy. Experience on demand: What virtual reality is, how it works, and what it can do / Jeremy Bailenson. New York, NY: W. W. Norton & Company, 2018.

Baratto, Romullo. "Trends in Architectural Representation: Understanding the Techniques." Accessed October 23, 2018. https://www.archdaily.com/867060/trends-in-architectural-representation-understanding-the-techniques.

Bernstein, Fred A. "Steven Holl Leaves His Mark: With blockbuster projects nearing completion all across America, architect Steven Holl prepares for the biggest year of his career—and many more milestones to come." Accessed December 24, 2018. https://www.architecturaldigest.com/story/steven-holl-leaves-his-mark.

Bouchlaghem, Dino, Huiping Shang, Jennifer Whyte, and Abdulkadir Ganah. "Visualisation in architecture, engineering and construction (AEC)." Automation in Construction 14, no. 3 (2005): 287–95. https://doi.org/10.1016/j.autcon.2004.08.012.

Boyum, Anna B. "Virtual Reality in Neurology." Accessed December 6, 2017. https://www.neurologytimes.com/stroke/virtual-reality-neurology.

Brett, Erica. “Architects in the Design of Virtual Reality Spaces.” UC Berkeley College of Environmental Design, 2015-2016. Accessed October 10, 2018. http://ced.berkeley.edu/downloads/thesis/arch/2016/Brett_Erica_Architects_in_the_Design_of_Virsua__Reality_thesis.pdf.

Brey, P. “Technology as extension of human faculties.” In Metaphysics, Epistemology, and Technology. Edited by Carl Mitcham. Jai, 2000.

Brown, Thackery I., Valerie A. Carr, Karen F. LaRocque, Serra E. Favila, Alan M. Gordon, Ben Bowles, Jeremy N. Bailenson, and Anthony D. Wagner. “Prospective representation of navigational goals in the human hippocampus.” Science (New York, N.Y.) 352, no. 6291 (2016): 1323–26. https://doi.org/10.1126/science.aaf0784.

ChaosGroup. “Guide to VR: Understanding and creating pre-rendered Virtual Reality content in V-Ray.” Accessed November 13, 2018. https://static.chaosgroup.com/documents/assets/000/000/023/original/Guide-to-VR_Chaos-Group-Labs.pdf?1472326790.

Cline, Ernest. Ready player one. First paperback movie tie-in edition. New York: Broadway Books, 2011.

Cobb, Sue V. G., Sarah Nichols, Amanda Ramsey, and John R. Wilson. “Virtual Reality-Induced Symptoms and Effects (VRISE).” Presence: Teleoperators and Virtual Environments 8, no. 2 (1999): 169–86. https://doi.org/10.1162/105474699566152.

Cummings, James J., and Jeremy N. Bailenson. “How Immersive Is Enough? A Meta-Analysis of the Effect of Immersive Technology on User Presence.” Media Psychology 19, no. 2 (2016): 272–309. https://doi.org/10.1080/15213269.2015.1015740. https://vhil.stanford.edu/mm/2015/cummings-mp-how-immersive.pdf.

Dede, Chris. “Immersive interfaces for engagement and learning.” Science (New York, N.Y.) 323, no. 5910 (2009): 66–69. https://doi.org/10.1126/science.1167311.

Delalex, Gilles. Go with the flow: Architecture, infrastructure and the everyday experience of mobility. Publication series of the University of Art and Design Helsinki A55. [Helsinki, Finland]: [Univ. of Art and Design Helsinki], 2006.

Dewey, Russell A. “Psychology: An Introduction: Motor Activity: Learning Curve.” Accessed March 10, 2019. https://www.psywww.com/intropsych/ch07-cognition/motor-activity.html#learningcurv.

Dineen, Barry. “Stereoscopic Renders in Unity3D for GearVR.” Accessed November 19, 2018. https://vrandarchitecture.com/2016/07/19/stereoscopic-renders-in-unity3d-for-gearvr/.

Dineen, Barry. “Collaborative VR Journeys: A new way to experience unbuilt architecture!” Accessed October 20, 2018. https://vrandarchitecture.com/2017/12/13/collaborative-vr-journeys-a-new-way-to-experience-unbuilt-architecture/.

Diversified Internet Holdings LLC. “The Ultimate Guide to Understanding Virtual Reality (VR) Technology.” RealityTechnologies.com. Accessed October 25, 2018. http://www.realitytechnologies.com/virtual-reality.

Easy Render. “Latest advancements in architectural rendering software.” Accessed November 8, 2018. https://www.easyrender.com/architectural-rendering/latest-advancements-in-architectural-rendering-software.

Easy Render. “What is Real Time Rendering and Why It Matters.” Accessed November 8, 2018. https://www.easyrender.com/3d-rendering/what-is-real-time-rendering-and-why-it-matters.

Enscape. “Enscape: Real Time Rendering and Virtual Reality: Live Link from BIM to Virtual Walkthrough.” Accessed November 9, 2018. https://enscape3d.com/.

Epic Games Inc. “Unreal Engine 4 Terminology: Get Started with UE4.” Accessed January 12, 2019. https://docs.unrealengine.com/en-us/GettingStarted/Terminology.

Farrelly, Lorraine. Basics Architecture 01: Representational Techniques. Basics architecture 1. Lausanne: AVA Academia, 2008.

Felnhofer, Anna, Oswald D. Kothgassner, Mareike Schmidt, Anna-Katharina Heinzle, Leon Beutl, Helmut Hlavacs, and Ilse Kryspin-Exner. “Is virtual reality emotionally arousing? Investigating five emotion inducing virtual park scenarios.” International Journal of Human-Computer Studies 82 (2015): 48–56. https://doi.org/10.1016/j.ijhcs.2015.05.004.

Fernandes, Ajoy S., and Steven K. Feiner. “Combating VR sickness through subtle dynamic field-of-view modification.” In 2016 IEEE Symposium on 3D User Interfaces (3DUI), 201–10. IEEE, 2016 - 2016.

Freeman, Daniel, Polly Haselton, Jason Freeman, Bernhard Spanlang, Sameer Kishore, Emily Albery, Megan Denne, Poppy Brown, Mel Slater, and Alecia Nickless. “Automated psychological therapy using immersive virtual reality for treatment of fear of heights: a single-blind, parallel-group, randomised controlled trial.” The Lancet Psychiatry 5, no. 8 (2018): 625–32. https://doi.org/10.1016/S2215-0366(18)30226-8.

Gartner Inc. “3 Reasons Why VR and AR Are Slow to Take Off: Here’s how technology product managers should plan for virtual reality and augmented reality adoption in the next few years.” Accessed October 23, 2018. https://www.gartner.com/smarterwithgartner/3-reasons-why-vr-and-ar-are-slow-to-take-off/.

Gartner Inc. “Gartner Hype Cycle: Interpreting Technology Hype.” Accessed October 18, 2018. https://www.gartner.com/en/research/methodologies/gartner-hype-cycle.

Google. “Google VR: Virtual Reality for Everyone.” Accessed March 10, 2019. https://vr.google.com/.

Holl, Steven, Juhani Pallasmaa, and Alberto Pérez-Gómez. Questions of perception: Phenomenology of architecture. San francisco: William stout, 2008.

Hypercell. “Hypercell Theme Park: Global Franchise Package: Wireless Full Action Up to 5000 Sq.m.” Accessed February 1, 2019. http://www.hypercell.com/.

Jackson, Matt. “Game Modifiers: Create big value for little cost to your game.” Accessed February 1, 2019. https://deepfriedgamer.com/blog/game-modifiers.

Jerald, Jason. The VR Book: Human-centered design for virtual reality. First edition. ACM Books # 8. New York, NY: Association for Computing Machinery and Morgan & Claypool Publishers, 2016.

Kielty, J. “Android v iOS market share 2018.” Accessed November 19, 2018. https://deviceatlas.com/blog/android-v-ios-market-share.

Kiteni, Konstantina, and Raphaela Groten. “The Sense of Embodiment in Virtual Reality.” Accessed March 1, 2019. http://diposit.ub.edu/dspace/bitstream/2445/53294/1/634024.pdf.

Kraus. “Theory of Phenomenology: Analyzing Substance, Application, and Influence.” Arch 630: Theory and Context. Accessed December 25, 2018. https://cte.ku.edu/sites/cte.drupal.ku.edu/files/docs/portfolios/kraus/essay2.pdf.

Lanier, Jaron. Dawn of the new everything: Encounters with reality and virtual reality. First edition. New York: Henry Holt and Company, 2017.

Laver, Kate E., Stacey George, Susie Thomas, Judith E. Deutsch, and Maria Crotty. “Virtual reality for stroke rehabilitation.” The Cochrane database of systematic reviews, no. 2 (2015): CD008349. https://doi.org/10.1002/14651858.CD008349.pub3.

LaViola, Joseph J. “A discussion of cybersickness in virtual environments.” ACM SIGCHI Bulletin 32, no. 1 (2000): 47–56. https://doi.org/10.1145/333329.333344.

Linowes, Jonathan. Unity Virtual Reality Projects. Birmingham, England: Packt Publishing, 2015.

Lombard, Matthew, and Theresa Ditton. “At the Heart of It All: The Concept of Presence.” Journal of Computer-Mediated Communication 3, no. 2 (1997): JCMC321-JCMC321. https://doi.org/10.1111/j.1083-6101.1997.tb00072.x. http://dx.doi.org/10.1111/j.1083-6101.1997.tb00072.x.

Lynda.com. “Unity for Architecture and Visualization Courses.” Accessed March 20, 2018. https://www.lynda.com/.

Lynda.com. “Unreal for Architecture and Visualization Courses.” Accessed November 10, 2018. https://www.lynda.com/.

Mealy, Paul. Virtual & augmented reality for dummies. 1st edition. Indianapolis IN: John Wiley and Sons, 2018.

Merriam-Webster. “Stereoscopy.” Accessed November 13, 2018. https://www.merriam-webster.com/dictionary/stereoscopy.

Merriam-Webster. “Virtual Reality.” Accessed October 10, 2018. https://www.merriam-webster.com/dictionary/virtual%20reality.

middleVR. “The challenges of creating a VR application.” Accessed October 23, 2018. https://www.middlevr.com/resources/the-challenges-of-creating-a-vr-application/.

Nguyen, Thong. Can Virtual Reality Change Your Mind? TEDxMinneapolis. With the assistance of TEDx. 2018. Video. Accessed March 1, 2019. https://www.youtube.com/watch?v=eFHj8OVC1_s.

Norman, Donald A. The design of everyday things. Revised and expanded edition. New York, New York: Basic Books, 2013.

Norman, Jeremy. ““Pygmalion’s Spectacles,” Probably the First Comprehensive and Specific Fictional Model for Virtual Reality (1935): HistoryofInformation.com.” Accessed October 23, 2018. http://www.historyofinformation.com/expanded.php?id=4543.

Ophir, Eyal, Clifford Nass, and Anthony D. Wagner. “Cognitive control in media multitaskers.” Proceedings of the National Academy of Sciences of the United States of America 106, no. 37 (2009): 15583–87. https://doi.org/10.1073/pnas.0903620106. http://web.uvic.ca/~dbub/Cognition_Action/Essaytopicscontrol_files/Mediamultitaskers.pdf.

Pike, Brandon. “Game Engines for BIM- The Next Generation of Project Visualization.” Accessed November 9, 2018. www.midwest-u.com/wp-content/uploads/.../Pike_BIM-and-Gaming-Engines.pptx.

Proulx, Michael J., Orlin S. Todorov, Amanda Taylor Aiken, and Alexandra A. de Sousa. “Where am I? Who am I? The Relation Between Spatial Cognition, Social Cognition and Individual Differences in the Built Environment.” Frontiers in psychology 7 (2016): 64. https://doi.org/10.3389/fpsyg.2016.00064.

RealityTechnologies.com. “Virtual Reality: The Ultimate Guide to Understanding Virtual Reality (VR) Technology.” Accessed September 5, 2018. https://www.realitytechnologies.com/virtual-reality/.

Riccio, Gary E., Eric J. Martin, and Thomas A. Stoffregen. “The role of balance dynamics in the active perception of orientation.” Journal of Experimental Psychology: Human Perception and Performance 18, no. 3 (1992): 624–44. https://doi.org/10.1037/0096-1523.18.3.624.

Rizzo, Albert A., Maria Schultheis, Kimberly A. Kerns, and Catherine Mateer. “Analysis of assets for virtual reality applications in neuropsychology.” Neuropsychological Rehabilitation 14, 1-2 (2004): 207–39. https://doi.org/10.1080/09602010343000183.

Searles, Rebecca. “Virtual Reality Can Leave You With an Existential Hangover.” Accessed December 6, 2017.

Shannon, Tom. Unreal Engine 4 for design visualization: Developing stunning interactive visualizations, animations, and renderings. Boston: Addison-Wesley, 2018.

Sherman, William, and Alan Craig. Working with virtual reality. Morgan Kuafmann, 1998.

Sherman, William R., and Alan B. Craig. Understanding virtual reality: Interface, application, and design. Second edtion. The Morgan Kaufmann series in computer graphics. Cambridge, USA: Morgan Kaufmann Publishers, 2019.

Smit, Martin Willem. “The Cyclical Iterative Design Process and Design Environment: learning from experience.” Dissertation, Delft University of Technology, 2008. Accessed January 18, 2019. https://www.scribd.com/document/16665247/Cyclical-Iterative-Design-Process-learning-from-experience.

Steinicke, Frank, and Gerd Bruder. “A self-experimentation report about long-term use of fully-immersive technology.” In Proceedings of the 2nd ACM symposium on Spatial user interaction - SUI ‘14. Edited by Andy Wilson et al., 66–69. New York, New York, USA: ACM Press, 2014.

Tang, Ronald. Step into the Void: A Study of Spatial Perception in Virtual Reality. UWSpace, 2019. http://hdl.handle.net/10012/14468.

Teslasuit. “Teslasuit: Ultimate Tech in Smart Clothing.” Accessed February 1, 2019. https://teslasuit.io/.

The Editors of Encyclopaedia Britannica, ed. Encyclopaedia Britannica: Stereoscopy. Encyclopædia Britannica, inc., 2013; Optics. Accessed November 12, 2018. https://www.britannica.com/technology/stereoscopy.

The Ghost Howls: A Blog about Virtual Reality, Startup and Stuff. “Virtual Reality is reaching a mature state according to Gartner.” Accessed October 23, 2018. https://skarredghost.com/2018/08/27/virtual-reality-is-reaching-a-mature-state-according-to-gartner/.

Thompson, Jeff. “Embodied Cognition: What It Is & Why It›s Important.” Accessed March 1, 2019. https://www.psychologytoday.com/ca/blog/beyond-words/201202/embodied-cognition-what-it-is-why-its-important.

Totzke, Michael. “The Real and Virtual Norm Li.” Accessed November 9, 2018. https://www.canadianinteriors.com/features/real-virtual-norm-li/.

Tricart, Celine. Virtual reality filmmaking: Techniques & best practices for VR filmmakers / Celine Tricart. 1st. New York: Focal Press, 2017.

Turi, Jon. “The sights and scents of the Sensorama Simulator: Time Machines.” Accessed October 23, 2018. https://www.engadget.com/2014/02/16/morton-heiligs-sensorama-simulator/.

Unity Technologies. “Stereo 360 Image and Video Capture.” Accessed March 2, 2019. https://blogs.unity3d.com/2018/01/26/stereo-360-image-and-video-capture/.

Unity Technologies. “Unity User Manual (2018.3 beta).” Accessed November 13, 2018. https://docs.unity3d.com/2018.3/Documentation/Manual/.

Virtuix. “Move Around in VR.” Accessed February 1, 2019. https://www.virtuix.com/.

vrban, Inc. “Insite VR: VR for Architecture, Engineering and Construction.” Accessed March 10, 2019. https://www.insitevr.com/.

Weidenbener, Lesley. “2018 Innovation Issue: Is virtual reality already dead? Or just getting started?,” June 1, 2018. Accessed October 24, 2018. https://www.ibj.com/articles/69026-innovation-issue-is-virtual-reality-already-dead-or-just-getting-started.

Wilson, Andy, Frank Steinicke, Evan Suma, and Wolfgang Stuerzlinger, eds. Proceedings of the 2nd ACM symposium on Spatial user interaction - SUI ‘14. New York, New York, USA: ACM Press, 2014.

X in Reality. “Markerless inside-out tracking.” Accessed October 24, 2018. https://xinreality.com/wiki/Markerless_inside-out_tracking.

Yee, Nick, and Jeremy N. Bailenson. “Walk a mile in digital shoes: The impact of embodied perspective-taking on the reduction of negative stereotyping in immersive virtual environments.” Cleveland, Ohio, Stanford University, 2006. https://vhil.stanford.edu/mm/2006/yee-digital-shoes.pdf.

Zerroug, Alexis, Alvaro Cassinelli, and Masatoshi Ishikawa. Spatial coverage vs. sensorial fidelity in VR. 2019.

Zorowitz, Jane. “It Just Got Real: Coaches like Bret Bielema and Bill Belichick are getting on the virtual-reality wave.” Accessed March 1, 2019. https://sportsworld.nbcsports.com/virtual-reality-sports-arkansas-kentucky/.

Appendices

Appendix A: VR Space Imagery

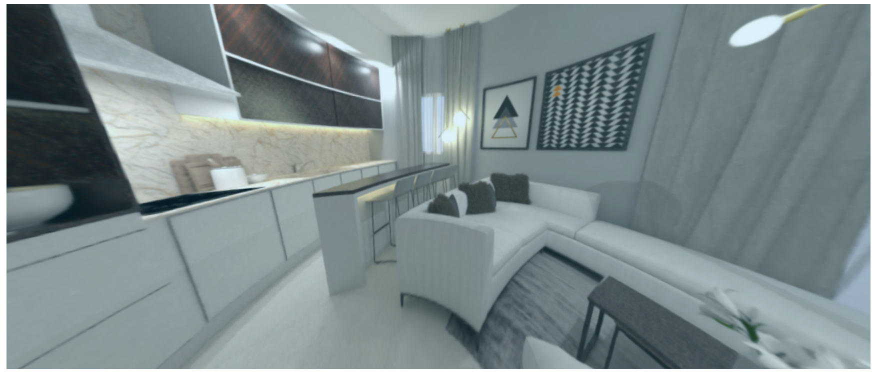


figure 11.01 Experiment 1: Living Room Space

figure 11.02 Experiment 2: Loft Space

figure 11.03 Experiment 3: Fallingwater Project

figure 11.04 Experiment 4: Kitchen Space

Appendix B: Glossary

Accelerometer: A sensory device that keeps track of the vertical direction. The device measures gravitational pull alongside keeping track of the VR headset's orientation.

Animations: The movements created through the principles of physics, stored as clips for objects/a hierarchy of objects in a scene. Animations are created as sequences of events are played and organized within a game engine's timeline. Animations use colliders to define the physical space an object takes up and define the interaction between objects with actions. Objects within a scene that activate animations to move and interact with other objects are called rigid bodies. Animations are created by rigid body components which when acting as mass with the physical principles of drag and gravity trigger kinematic objects. Animations thus work through understanding the theory of relativity which asserts that objects move relative to each other. Hence, no object within a game engine scene is ever at absolute rest or absolute motion.

Assets: Any data item that is used in a game engine's projects. Assets are organized directly as a self-contained unit and correspond to a filing structure located on a local disk. Assets can come from files outside of a game engine, such as a 3D model, an audio file, an image, or any of the other types of file that the engine supports. Assets also represent the hierarchy of project management in game engines, acting as the hub for all imports and links within the project file. Hence, they act similarly to the link manager that is commonly found in architectural computational tools.

Augmented Reality (AR): AR adds to reality, projecting information on top of what one is already seeing. The virtual addition is created through still graphics, audio or video generation onto the real world. Augmented reality differs from VR in that it adds to the real world instead of creating every part of the visual experience from scratch.

Avatar: The digital manifestation of VR users in virtual space. Avatars are commonly represented as character figures.

Baked Lighting: The first of two times a game engine processes lighting. Light Baking occurs during scene creation when an object turns static, which notifies the game engine of lighting staying in place. At runtime, the engine loads the lightmaps and applies their pre-calculated illumination back to the objects in the level, instead of having to calculate the lighting in real time every frame of the engine's main loop. Baked lighting is thus greater for performance of a scene as it can include processes that are too intricate to complete during run time.

Components: The functional capabilities attributed to objects and actors in game engines. Components are the nuts and bolts of objects and behaviors in a game. An object acts as a container for many different components. In computational architectural design tools, components can be compared to the "Properties" toolbar assigned to any object.

Controllers: Devices that control interactions within Virtual Reality. Controllers allow a user's movements to be tracked and copied by a digital avatar in the simulation, mimicking their gestures and controls from physical space into their virtual one.

Cubemaps: The efficient way to process 360 degree images. Cubemaps place six images on the inner surfaces of a virtual cube. The net of these six images is then unraveled to be placed side-by-side as a single 6:1 aspect image.
The image stitched together from these six then appears as a full 90-degree FOV image.

Degrees-of-Freedom (DoF): Movement in space is calculated as by degrees of freedom. Six is the maximal degree of motion for an object in space.

Display Screens: Through the use of lenses, screens are placed in front of eyes to allow a user to view content. Displays for VR headsets are typically LCD (Liquid Crystal Display) and are connected to a computer or commandeer that of a smartphone device.

Dynamic Lighting: The second of two times a game engine processes lighting. Dynamic Lighting is used exclusively for objects or lights that are in motion. It calculates light for the moving object in real time as the engine processes the animation of that object during runtime. All calculations for dynamic lighting are calculated when the visualization is rendering or running.

Embodied Cognition: The theory of the body playing an important role in the conscious processing of knowledge and stimuli.

Extended reality (XR): The umbrella term that encompasses the full spectrum of realities.

Field-of-View (also called Field of Vision or FOV): A component used to indicate how wide a picture is as viewed. In VR, it is used to provide users with a realistic perception of their environmental landscape. Field of view is measured based on the lateral degree of display.

Frame Rate: The measure or frequency by which a display screen shows consecutive images (or frames). In virtual reality, a minimum frame rate of approximately 60 frames per second is needed to avoid content stuttering or cause of simulation sickness. Virtual reality headsets are set to increase their frame rate, working towards dismissing latency and providing a more realistic experience.

Gesture Input: The principle behind programming interactions that are triggered within a VR simulation. Input occurs as motions are programmed to mimic real life movements. Often the most successful interactions are built from natural gestures and intuitive forms of movement.

Gyroscope: A sensor that calculates the orientation of a device. It does this to either help devices maintain a specific orientation or ensure changes in orientation function appropriately.

Haptics: The engagement of touch and tactile sensations to interact with a simulation.

Head-Mounted Display (HMD): A display device that is worn on the head or as part of a helmet, that has a small display optics in front of a singular eye (monocular) or both eyes (binocular). HMDs either replace a user's field of vision or augment features to an existing field of vision.

Image Maps: The feature that represents how the texture of an image is visible on object with settings of smoothness, specularity, shine and reflectivity.
Interactive Modifiers: The player-facing choices that can be made during the course of running an application or game. Providing a user with a modifying option set changes the virtual experience by adding to the number of outcomes along with increasing game interactivity.

Interocular Distance (IOD): The essential factor in determining a scene's scale in relation to its viewer. This factor measures the distance between the two left and right cameras that render in stereo. If the distance is too small or too large, the viewer's body is scaled disproportionately to the world around them. A VR experience can't exist with disproportions because a viewer experiences what is occurring as if through their own eyes.

Latency: The amount of time it takes for an image displayed in a user's headset to catch up to their changing position. Latency is also considered as a delay, measured in milliseconds (ms). Low latency, or very little delay, is needed to make the human brain accept the virtual environment as real. The higher the latency, a noticeable and unnatural lag may set in, consequently arousing simulation sickness for the user.

Lenses: Glass that lies between the eyes and the pixels on a display screen. The lenses of the headset are responsible for mapping the up-close display to a wide field-of-view. They focus and reshape pictures for each eye by angling imagery and mimicking how each eye takes in views of the world. This angling achieves the impression of depth and solidity, allowing the perception of a three-dimensional image.

Lightmap UVs: The textures saved on disk through a software's process of pre-calculating the lighting for objects and saving them (light baking). They control how simulated lighting hits different objects, by essentially "mapping" the way light will hit an object.

Light Baking: The process of pre-calculating the lighting for the objects in your level and saving it to textures on the local disc. These textures are called lightmaps.

Locomotion: The act of teleportation in Virtual Reality. Most often, a user points to their direction of travel and then pushes a controller button to move to that virtual location. Locomotion is used to create a comfortable form of movement in virtual space without having to consider the difficulty of traversing those large environments in reality.

Magnetometer: A sensory device that tells the VR headset which direction it is facing on the surface of the earth, acting as a compass and measuring magnetic fields.

Mixed Reality (MR): Takes the real world and integrates computer-generated content to interact with the view of the real world. It carries the ability to take fully-generated digital environments and connect them to real-world objects, making it the only technology able to combine the analog and digital realities.

Normal Maps: The feature that represents the shape and texture of a surface with respect to how light bounces off of it. This effect is further created by height maps and occlusion maps which alter the emission of light emanating from any object in any scene.

Objects/Actors: The baseline building blocks in game engines, which carry the invisible functionality required for assets in a project. Once assigned, they contain functional attributes, which are referred to as components, which allow it to become a character, environment or special effect. Objects can be compared to the creation toolkits prominent in any modeling or drafting software, that allow for the conception of simple surfaces, lines, or solid shapes.

Prefabs: A reusable asset in game engines. Game engine prefab systems allow for the creation, configuration and organization of objects, actors and their associated components, values and properties. A Prefab is a template from which to duplicate them and apply changes to all of them at once. In this manner, they act like “blocks” used in computer-aided architectural design tools.

Presence: The effect of feeling as though one is truly inside a virtual space. It is a feeling that transports a user from their physical surroundings into a virtual world. Experiencing presence in any VR scene aids in providing the basis for a positive experience within a simulation.

Processing: The instances of substantial amounts of power being used by technology. VR systems demand speeds in controlling input information (input processing), data retrieval and distribution (simulation processing) and minimizing lag time (rendering processing).

Projects: All the information created through any game engine exists as self-contained units. Projects compile all the aspects of a game such as any content or code used and subsequently act as a directory of information. In a similar fashion, architectural projects are comprised of models, elements and drawings all compiled together to form the entire design.
Proprioception: The human sense of perception featuring bodily position and movement. It allows for an awareness of equilibrium and balance, engaging the human senses that understand notions of force.

Rendering: The process of digital information changing as new movements are tracked. It is calculated through image synthesis or the automatic process of generating a photorealistic or non-photorealistic image from a 2D or 3D model (or models in what collectively are called a scene file) by means of a computer program. In this way, real-time rendering refers to animations rendered at high speeds to mimic the appearance of being generated in absolute real time. It involves three stages; application, geometry and rasterization. The end result of real-time renderings is an animation rendered by the graphics system processing image frames quickly enough to show realistic motion. Due to this, real time rendering is measured in frames per second.

Resolution/ Display Quality: The optics and visual qualities that affect how a user views the image quality and how they experience the virtual world. Images appear clearly due to features such as display resolution, optic quality, refresh rate, and field-of-view.

Room-Scale VR: Experiences created using VR devices such as the Oculus Rift and HTC Vive, which track the bodily movement of users in a defined region.

Sensors: Devices that measure user motion and direction in space. Magnometers, accelerometers and gyroscopes are the most common types present in VR devices.

Simulator Sickness: When latency and cognitive dissonance present themselves in a VR simulation, a user might have triggered feelings of sickness and discomfort. If a user’s display doesn’t match their movements, this unexpected delay triggers an uncomfortable physical response in the form of nausea.

Stationary VR: Experiences created using VR devices such as Google Daydream. Google Cardboard and Samsung VR, which use gyroscopic sensors to track only a user’s head movement.

Stereoscopy: A process of seeing an image in three dimensions using principles of binocular vision. Stereography is the means of delivering this, achieved through the creation of two images from separate cameras (slightly offset from one another) to mimic what the eyes would see at their respective angles. This act provides the experience of something that appears to be 3D, as mimicked to be from one’s own eyes.

Tracking: The vital task measure of calculating or measuring user movements in space. Tracking is mainly achieved in VR in three ways. Head tracking refers to the way in which the view in front of a user will shift as they look up, down or side-to-side. Motion tracking is the way in which users view and interact with space (e.g. hands, movements, etc.) and is facilitated with the use of controllers, joysticks and treadmills.
Eye tracking uses an infrared sensor that monitors eye movements within a headset.

Virtual Reality Audio: A feature facilitated by headphones affixed to a VR headset. It works via positional, multi-speaker audio (often called Positional Audio) to gives the illusion of a 3-dimensional or spatial sound, increasing/decreasing volume as a user gets close to objects in a simulation.