Expertise
and insight
for the future

Soyhan Yazgan

Developing a Virtual Reality Application for Realistic Architectural Visualizations

Metropolia University of Applied Sciences

Bachelor of Engineering

Information Technology

Bachelor's Thesis

26 June 2020

Author	Soyhan Yazgan
Title	Developing a Virtual Reality Application for Realistic Architectural Visualizations
Number of Pages	30 pages
Date	26 June 2020
Degree	Bachelor of Engineering
Degree Programme	Information Technology
Professional Major	Software Engineering
Instructors	Janne Salonen, Supervisor

The aim of this thesis is to understand how Virtual Reality works, what are its weaknesses and how to improve on them. It also tries to make use of Virtual Reality for architects.

In this thesis, problems and limitations of Virtual Reality are discussed, how Virtual Reality can be used for architecture, how realistic graphics can be made and how Unreal Engine 4, a game development engine, can be used as a creation tool.

As a result, an application was made to be used with both a Virtual Reality kit and with regular PC with keyboard and mouse. This application is a virtual tour of an apartment where furniture can be moved around, and their materials can be changed in real-time.

Keywords	virtual reality, architecture, visualization, render

Contents

List of Abbreviations

2D	2-Dimensional
3D	3-Dimensional
AR	Augmented Reality
BIM	Building Information Modelling
CAD	Computer-Aided Design
CGI	Computer Generated Imagery
CPU	Central Processing Unit
FPS	Frames per Second
GPU	Graphics Processing Unity
HMD	Head Mounted Display
MR, XR	Mixed-Reality
PBR	Physically Based Rendering
PC	Personal Computer
RT	Ray Tracing
SSR	Screen Space Reflections
UE4	Unreal Engine 4
UI	User Interface
UMG	Unreal Motion Graphics
VR	Virtual Reality

1 Introduction

Architecture has existed for thousands of years. Ever since humans started living in caves, and using stones, wood, bricks and any other material usable for making a shelter, the need of architecture has existed. The need will continue to exist forever. Architecture is like a living being; it changes, and it evolves. Architecture is made by architects. Thus, the need for architects will exist as well. An architect designs and creates a structure. He uses and designs space, and he does this 3 dimensionally. Eventually when his design is finished, either to demonstrate it, showcase it, explain it or build it, he will need to have a presentation technique. This can be done by drawing plans on a paper, building a 3-dimensional (3D) models or use computers to render the finished look. In 21st century, there is a new tool that can be used to present a design, and this tool is virtual reality (VR) (concept seen in figure 1). [1]

Figure 1. Concept of Virtual Reality used for architectural presentations.

Architecture may have existed for centuries. Computers however, are the invention of 20th century. Like many events that changed and defined ages in human history, in the future we will look back and mark invention of computer as the event that started The Digital Age. People of today do not need to leave their house to do regular life events such as; studying, working, socializing, being entertained and they can even order food and groceries delivered to their home. [2]

One can argue the digitalization and not leaving home is making people unsocial and introvert. But in 2020 when this thesis is being written, an epidemic is happening all around the world which is caused by The Coronavirus (COVID-19 or SARS Cov-2), a contagious flu. The way to fight this epidemic has been staying home as much as possible and working/studying from home. In a world where overpopulation is increasingly becoming a problem, also is creating other problems such as traffic, working and studying from home helps with time and saving travel expenses. In this kind of world digitalization is definitely needed. Through a computer with internet, a person can reach all the information he needs within seconds. He can also travel the world without spending any money or travelling without any cars or planes. When VR is added to the picture, it is possible to be in the same room with other people anywhere in the world and interact (example in image 1). It is very possible for future to have schools in a completely virtual environment. Coronavirus epidemic clearly proved that the future will be lived in the virtual world and VR development is going to take big part of it. [3]

Image 1. A virtual reality online meeting scenario

In this thesis, the concept of virtual reality, its uses and limitations and current state of hyper-realistic computer-generated imagery (CGI) will be discussed and steps for developing a VR program for architects to use as a presentation technique will be explained.

2 Background Information

2.1 Virtual Reality

2.1.1 What is VR

Virtual Reality Society defines virtual reality as an emulation to trick senses in order to reach a near-reality experience. The concept virtual reality might sound new to some but tricking our mind to be somewhere else when we are not, is not a new concept. Seeing, in other words, using eyes for visual sense is the strongest sense of existence in a space. If a person is in a room; the easiest way for him to know it, is by simply looking around. Hearing, smelling and touching are other senses that can enhance this experience, but they are secondary senses. [4]

2.1.2 Stereoscopic View

One of the important ways the human eye works is the perception of depth, achievable by two different eyes seeing slightly different perspective and brain interpreting this as the sense of depth. When a painter paints, or a photographer takes a photo, these images are in 2-dimension (2D) and will look flat and depthless to our eyes. Even when perspective is used, our eyes and brain are clever enough to know the difference. However, if 2 different photos from slightly moved angle is taken and displayed to each corresponding eye, the brain is tricked, and concept of depth is better achieved (Illustrated in figure 2). [5] This method has been in use for over a hundred years. As an example, a stereoscopic photography card of a view of Boston, taken from 2 different angles used with a stereoscopic viewing device was already made in 1860. [6]

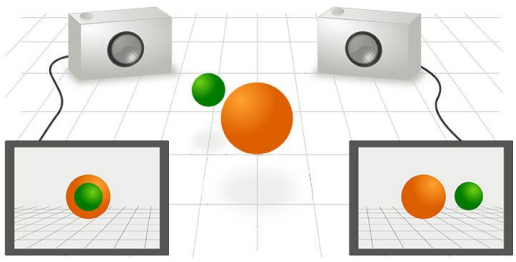

Figure 2. Illustration of stereoscopic view

2.1.3 HTC Vive

Today, with improved technology and fast computers, 3D movies and VR games exist. HTC and Valve companies together developed a consumer level VR device called HTC Vive. It consists of a head mounted display (HMD), two controllers, two base stations, the powered receiver box and cables (as seen in figure 3).

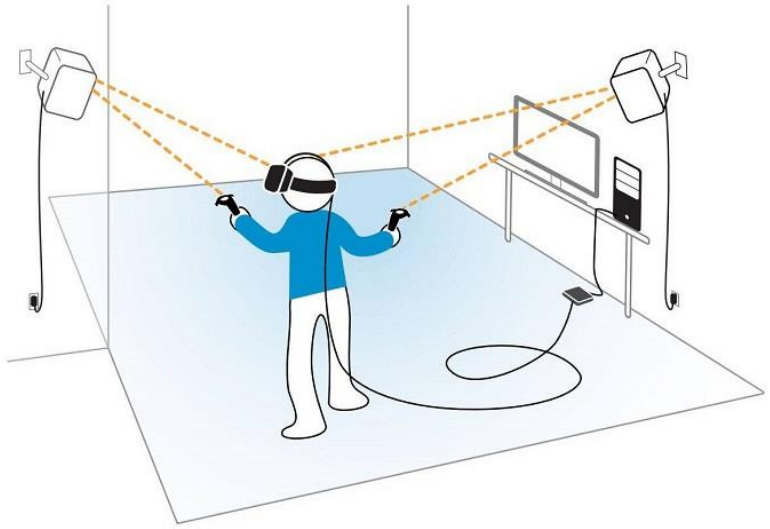

Figure 3. HTC Vive room setup

HMD is a headset with two lenses for stereoscopic view, a stereo headphone connection cable and head strap. This device then connects to the receiver box which connects to a computer. At its current stage, HMD is heavy, cables are long and tangling to the user while in motion and most importantly; head straps, within several minutes of use, create uncomfortable heat for the user's head and face. These are currently the main reasons for users to not want to wear the headset for long sessions. But future versions will battle these difficulties.

Two base stations send information to HMD and between each other to create the virtual space. Vive can work with one base station but accuracy of the tracking is not the best with only one. Yet, it can also work with more than two stations to further improve the accuracy. [7, a]

Two controllers connect with base stations and HMD to give user an input option. Usual actions with these controllers are grabbing, hitting, pointing and selecting. They also have haptic feedback which helps when an object comes in contact with the controller in virtual world. This vibration feedback tries to mimic the sensation of touch. [7, b]

For the remainder of this thesis, HTC Vive will be used as the VR device.

2.1.4 Future Improvements

Vive is still in development and newer versions are to be released. [8] It has its problems and has a lot of room for improvement, such as resolution, frames per second (FPS), quality of the image, real-time response, dizziness, focus depth, weight of the HMD and heating of the straps and long cables. While technology improves, weight of the device will reduce, cables will be removed for wireless access and the comfort will get better. But some of the other limitation; such as focus depth may be difficult to solve. Human eye adjusts its focus on the distance of the object, while this focus can falsely be created by depth of field effect in a single picture, if the person looks in another object, the focus changes, and the hardware needs to have ability to detect this and redraw the depth of field accordingly. Other improvements to more realistic VR such as true 3D spatial audio, sensation of touch and smell may also one day be implemented to the system. It is not known what the abilities of VR will be in the future (example in image 2).

Image 2. Ready Player One is a movie about the possibilities of VR in year 2045. [9]

2.1.5 Uses of VR

Virtual Reality has so far been used for the following purposes:

- architecture
- gaming
- interactive movies
- medical training
- medical treatments (pain management, PTSD, anxiety...)
- work collaboration

2.2 Architectural Visualization

Architectural visualizations are representations of a planned structure for them to be easily understood. An architect or an engineer is able to read through floor plans but not everybody can. The purpose for these visualizations is to present it to everyone, to better understand the final product, to help with the design, to look for errors or for marketing purposes. [10]

2.2.1 3D Models

One of the methods to show a finished structure is building a real life, small scale 3D model. These models are usually around 1:50 to 1:200 scale and built with paper, foam, wood, laser cut or 3d printed. It is pretty common to see pure white models for its simplicity and clarity. Building these models require good hand skills since it can become very messy. There are companies that specialize in building them for professional purposes. Even though they are a classic solution for visualization, they are difficult to make, hard to fix and most importantly they are static. Today, there are more dynamic methods to display a building. [11]

2.2.2 Still 3D Renders

Usually made by a 3D software such as 3ds Max, 3D renders are the most popular ways to present a building. This type of presentation used to be done with paper and pen by drawing the finished structure. Today, computers are used. They can be hyper realistic (as seen in figure 4) or stylized depending on the purpose. With the increasing speed of the computers and increasing availability of the tools, they are becoming more cost efficient everyday. However, they are still static images. While preparing a scene might take hours to days, rendering the final image also takes from minutes to hours, for a single frame. It is usual to have several renders of a building to show different angles and details, yet once the render is finished, it is not possible to make dramatic changes on it, such as removing a piece of furniture, changing the colour of a wall or changing the daytime.

Figure 4. A hyper-realistic interior 3D render (render time: 20 minutes)

2.2.3 3D Animations

Walkthrough animations are yet another way to present a building. While a still render is static, an animation can have movement. It is possible to have changes in an animation, show timeline, movement, progress or change sun angle. While these are a step better than still images, they come with extremely high amount of render times and building the

scene effort. It can take a month to make a 3D animation. It is costly. If there is a mistake on the animation, it takes long time to re-render the animation. Moreover, whatever is baked in the animation, even though in motion, once baked, is not anymore dynamic. It is not possible to move the camera on the fly or change the scene however liked.

2.2.4 Real-Time Rendering

With the increasing speed of central processing units (CPU) and graphics processing units (GPU), it is possible to make real-time renderings. These are another step forward than animations. Today, gaming engines such as Epic Games Unreal Engine 4 (UE4) allow hyper-realistic graphics in real-time. Companies started using these engines for architectural renderings instead of game programming. The advantages of using a game engine for rendering are:

- speed
- real-time response
- ability to change anything on the fly
- ability to interact
- using programming to add functionality to objects

Abilities of Unreal Engine 4 will be discussed further in upcoming chapters.

2.2.5 Virtual Tour

When real-time rendering is combined with VR, the result is a virtual tour. This method is the most immersive, most interactive method for architectural visualizations. The user can experience the environment in stereoscopic 3D, look around in 360 degrees, move around in 3D, interact with object using the controllers, change the scene however he likes, hear the ambient voices (if any implemented) and feel how it will feel when the structure is finished. This method also allows designers to experience their design while making them. A program can be made for designer to use during the designing process; such as making walls, adding furniture all within the VR in real-time. [12]

2.3 Design Steps

Designing a building is made out of 5 stages:

- pre-design
- schematic design
- design development
- construction drawings
- construction

In the case of this thesis, a medium sized sample apartment will be designed and modelled. The design development phase will be the main step discussed. Last two phases will be skipped because this project will not come to life but will be only a sample project to show abilities of a VR application. [13, a]

2.3.1 Pre-Design

Pre-design phase is about analysis, budgeting, scheduling and targeting. In this case, design is simple, budgeting is zero, scheduling is about one week and since it is not a real-life scenario, it will be imagined as an apartment floor on a 20-floor building. [13, b]

2.3.2 Schematic Design

This phase includes planning of the floor and spaces, sketching and creating the main layout. In the case of a medium size apartment, the floor plan will contain a bedroom, a kitchenette with living room, a bathroom, a toilet and a sauna. The drawings will be made in Autodesk Revit and then exported to Unreal Engine 4 for programming. [13, c]

2.3.3 Design Development

In this phase visualizations are created. In the case of this thesis programming for VR functionality will be created. Also, materials will be assigned, furniture will be imported, and final export of the program will happen. [13, d]

2.4 Autodesk Revit

One of the most commonly used computer-aided-design (CAD) program by engineers and architects is AutoCAD. This program allows precise drawings in vector. Even though it is easy to learn and draw lines and arcs on AutoCAD, detailing and rearranging the drawings take time. An example to this is drawing window or roof details. Once the floor plan is drawn in 2D, it is possible to rotate the plan into 3D and draw vertical lines. Line by line drawing all these details take long time. Once drawing is done and if a wall needs to be changed in size, a lot of effort is necessary to redraw all the lines. While AutoCAD used to be a favourite tool, another tool was developed for better use for architects.

Revit is a program, currently owned and developed by Autodesk, is a building information modelling (BIM) tool. It extends the functionality of AutoCAD. In AutoCAD lines are drawn. In Revit, building objects such as walls, windows, doors and roofs are defined and created. Revit objects exist 3-dimensionally. When a wall is defined, program asks for length, height, width and also materials, layers and a variety of extra information. Windows exists within walls. When a wall is modified, windows move with it. Roofs are generated automatically but can be modified manually. Revit is designed for architects by architects and usually is intelligent enough to understand what the user is trying to do. As a BIM tool, it can also work with timed events such as building, containing information, wearing and demolition. It also contains performance tools for cost calculating. These are the reasons why Revit is more preferred over AutoCAD today. [14]

As explained above, Revit works on 3D. A floor plan can be drawn s and can be turned into a 3D model within minutes. As much intensive detail can be put into a Revit project, for this thesis, only the walls, windows, doors, floors and ceilings will be created within Revit. As stated in design development section, the floor plan for this project contains a bedroom, a kitchenette with living room, a bathroom, a WC and a sauna (as seen in figure 5).

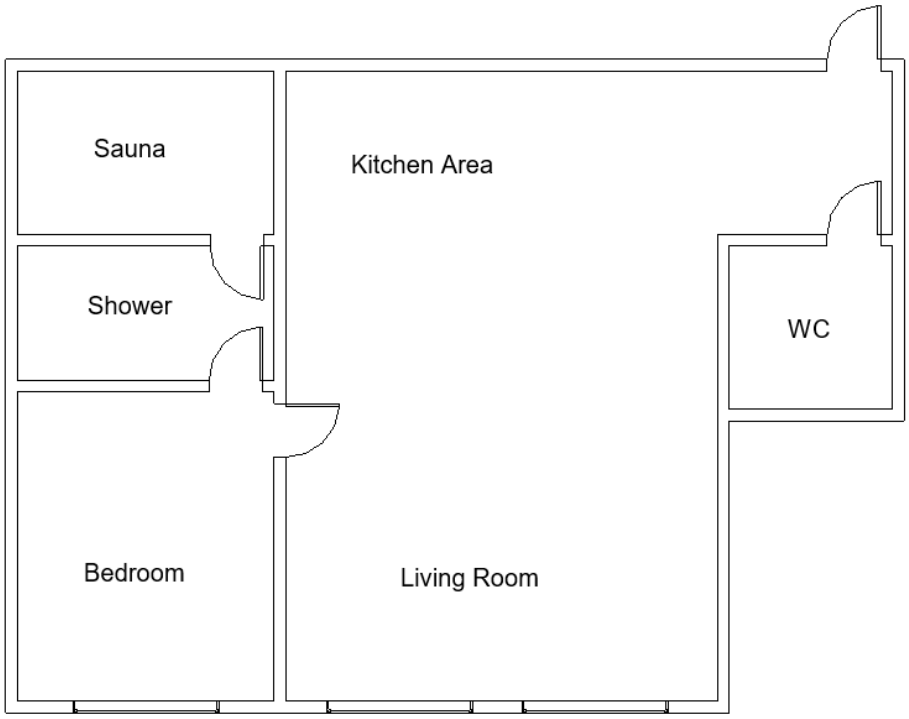

Figure 5. Revit floor plan

3 Unreal Engine 4

Unreal Engine was originally created by Tim Sweeney for the game Unreal in 1998. On 2015, UE4 was released to public for free with a royalty fee of 5% over 3000$ earning. With the free model, it gained lot of popularity within the indie industry. Small companies who cannot offer to buy expensive software could freely use UE4.

UE4 has big amount of documentation and help, YouTube tutorials and forum posts. The community around UE4 is increasing by day. After their own hugely successful game Fortnite, Epic Games started buying paid resources and set them free on their market-place for developers to use. Not only UE4 is capable of working with large variety of asset imports from other sources, but also allows on Epic marketplace content creators to sell their own assets. [15]

UE4 allows creation of content for not only personal computers (PC) but also consoles and mobile devices. The engine is used not only by game developers but also for making movies and architectural visualizations.

Here are some of the key features of Unreal Engine 4: [16]

- C++ and Phyton scripting
- Blueprints, Unreal's visual scripting language
- Datasmith importing tool
- landscape tools
- asset optimization tool
- Animation Blueprints
- Forward Rendering shader optimized for VR
- Physical Based Rendering (PBR) material editor
- real-time Ray Tracing (RT)
- advanced shaders (lit, unlit, transparent, clear-coat, subsurface scattering, skin, hair, two-sided foliage)
- virtual texturing
- post process effects
- particle, destruction, hair & fur and cloth physics
- Unreal Motion Graphics (UMG) user interface (UI) editor
- AI and navigation
- multiplatform development
- VR, augmented reality (AR) mixed-reality (XR, MR) support
- Marketplace
- source-code available

With this project, there will be two different maps available with same furniture but two different functionalities. The first version will be controlled with keyboard and mouse and will be viewed on a regular computer screen. This version will be called Regular Map. The second version will be for VR and will use VR motion controllers. This version will be called VR Map.

The main reason two versions needing to be created is performance. VR version has to have stable 90+ fps, otherwise will cause motion sickness. When current generation GPU renders two different large resolution screens, with Ray Tracing enabled, it cannot maintain 90 fps. Details of the differences of the maps will be explained in related chapters.

3.1 Project Settings

When a new UE4 project is created, with version 4.25.1 used with this thesis, several templates will show to choose from. These have starting points for different types of projects. The “Architecture Template” and “Virtual Reality” template might be useful but for this thesis, an empty template with blueprints is selected. After the project is open, some settings need to be changed. First, Ray Tracing and Skin Cache needs to be enabled. Then Default RHI needs to be changed to DirectX 12. Datasmith and SteamVR plugins need to be enabled. Then project is restarted.

With the project having two different maps, the VR Map needs to have different lighting and reflection techniques setup. Screen Space Reflections will be used as they are second best option for reflections after Ray Tracing. Normally, a VR project would be used with Forward Shader for performance gain. But with this project, Deferred Shader will be used because Forward Shader does not support Screen Space Reflections. [17]

3.2 Importing to Unreal Engine 4

Unreal Engine 4 comes with an intelligent importing tool called Datasmith. It can read different modelling app files including Revit. When the modelling in Revit is finished, and if the Datasmith plugin for Revit is installed, the project will have Add-Ins tab where Export to Datasmith button is placed. After this, a folder with Datasmith file is created to be imported to Unreal Engine 4.

Inside UE4, if Datasmith is enabled in plug-ins, the Datasmith import button can be found on top menu. When the file created earlier is selected, it will ask what to import; such as models, lights and cameras. After the importing is done, the model as it was created in Revit will appear in UE4 project folder. None of the objects have materials at this stage. Walls, floors and windows need to have materials created for them. Their collision settings are set to complex.

One great thing about Datasmith is that, if user wants to edit the original Revit file, a reimport can be easily done with modified parts and they will fit perfectly on their coordinates on the UE4 project. [18]

3.3 Post Process Volume

To be able to change several shader settings, a Post Process Volume needs to be added. After it is added, inside its settings, the following options are set for the Regular Map: [19][20]

- Infinite Extend (Unbound): Enabled - Makes Post Process affect the whole map
- Reflections: Ray Tracing – Realistic real-time calculated reflections
- Ray Tracing Global Illumination: Final Gather – Calculates real-time shadows
- Ray tracing Global Illumination Samples Per Pixel: 24 – Higher is better but performance heavy, lower will make splotches appear on the screen.
- Translucency Type: Ray Tracing – Realistic real-time calculated transparent materials
- Bloom Method: Convolution – High quality Bloom effect on bright lights

The VR Map will have the following settings set differently than previous settings:

- Reflections: Screen Space Reflections – Quick reflections but works only with objects that are visible in the view
- Ray Tracing Global Illumination: Disabled – Global Illumination will be pre-baked
- Translucency Type: Raster – Non-ray tracing translucency
- Motion Blur: 0 – Motion Blur gives VR users motion sickness

In the future, when faster GPU's are released, VR version can also have Ray Tracing enabled. Currently, it is not suitable with today's GPU's.

3.4 Materials

When it comes to creating objects and environments, creating materials plays a crucial role. If for example there is a cube, it has 8 vertices on the corners, 12 edges and 6 faces. These are defined by coordinates and math functions. How they are displayed to the screen is defined by materials. If the faces are matte, shiny, transparent, has colors,

has bumps and deformations; they are defined in material editor. A flat plane can look like wood floor, a mirror or even ocean waves moving with wind.

In this chapter, how to make realistic looking materials with UE4 will be discussed. However, equivalent material expressions exist in other 3D software and the concept is similar when it comes to static materials; such as Diffuse, Reflection, Roughness, Opacity and Bump. On UE4, when a new material is created, a material attributes box is seen. This box has plenty of inputs to manipulate the material in the way wanted. In the simplest way, several maps can be imported and used to create good static material. It can also be used to create complex, multi functional, real-time manipulatable materials defined with functions and maths. [21]

- Base Color is the first input. A 3-vector RGB node can be forwarded to Base Color and depending on the RGB parameters, the material will change color. However, it will look flat and boring. Another method is to use a texture map. PRB material sets can be downloaded from internet or from Unreal Marketplace which comes with several maps to use. A wooden floor would have the matte colored texture of the wood in diffuse channel. This can be forwarded to Base Color. With only diffuse, the material will look flat, unreflective and unrealistic. It will have the texture, but it will look like paper. To make it realistic, more expressions are needed. [22, a]

- Metallic and Specular are similar but different reflection expressions. A single vector node can be used for these. Metallic gives a mirror like reflection when set to 1.0 and Specular gives more plastic looking reflections. If a PRB material is used, a reflection channel will be available as a texture map, that defines which parts of the surface has more reflections. If reflection can be defined between 0.0 to 1.0, a reflection map with black and white texture translates into numbers between 0 to 1 on locations needed. In a wooden floor material, the lines and cracks would be darker, and planks would be lighter. [22, b]

- Roughness is connected to reflections. When Metallic is set to 1.0 and Roughness set to 0.0, the object will be a perfect mirror reflecting the light as is. However, when Roughness is set to 0.5, the reflection will get blurry and the object will look like a brushed metal, blurry and shiny. When Roughness is set to 1,

reflection will not be too visible anymore because of spread of the blur. Metallic/Specular and Roughness together make big difference when making materials look cartoonish, fake, 90s looking or hyper-realistic. Roughness also makes big difference in the performance. When making hyper-realistic looking environments, it is advisable to use reflection and roughness on every material. [22, c]

- Emissive Color make the object generate light. they can be RGB and have textures. A light bulb would have Emissive Color in it. [22, d]

- Opacity makes the object transparent. If an object has only its Opacity set close to 0, it will look faded, like a ghost. A glass on the other hand would have low Opacity and high Metallic and low Roughness values because it reflects the light. But it also would need Refraction setup. [22, e]

- Refraction is for a transparent object, to bend the light passing through it. Every transparent object, including air has a refraction value in real world physics. For example, vacuum is 1.0, water is 1.33, flint glass is 1.65 and diamond 2.42. If a glass material is made, Opacity of 0.2, Metallic of 1.0, Roughness of 0.0 and Refraction of 1.65 should be set. [22, f]

- Normal is a special material expression. (Other programs might refer to it as height map or bump map, even though they work similar, these maps would look different but can be convertible between each other.) Normal map is used to determine if the flat object would have non-flat surface, such as dents and scratches. In a wooden floor, planks and corners of the planks would be different height, and this is defined by Normal Map texture. If an object is modelled with all the details and dents on its surfaces, it would have high number of polygons. This makes the project size high, used memory high and CPU calculations heavy. But if the object is defined flat but material has bump, the object itself is not manipulated and lighting shader calculates the dents. It is faster and more practical. [22, g]

- World Displacement is similar to Normal but there is a difference on how they work. Normal does not manipulate the object, but Displacement does. It adds and changes the vertices of an object. A flat plane object with moving wave

Displacement map will look like ocean waves. A sphere object with Displacement can look like a strawberry. A flat plane with noise map as Displacement can look like grass. Displacement is performance heavy because of new polygon creation and should be used only when necessary. [22, h]

- Tessellation is used with Displacement. When Tessellation is used, the object is divided to smaller sized polygons, allowing Displacement map to use more detail. If the object looks spiky, blocky and low resolution with Displacement, Tessellation will add more polygons to make the object higher resolution and more detailed. UE4 can use an object's distance from the camera to add more Tessellation to an object when close and reduce when far, which helps with performance. [22, i]

An example of a realistic looking wooden floor can be seen in image 3 with all necessary maps attached.

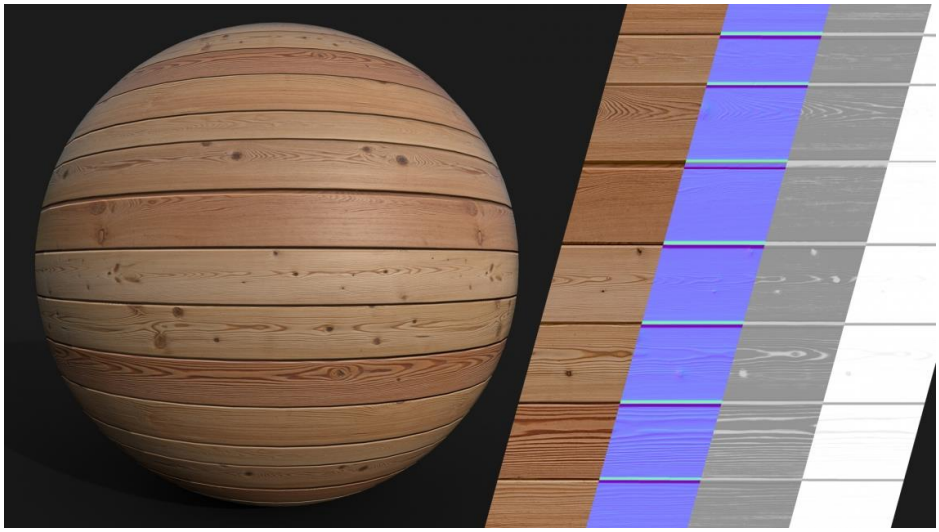

Image 3. Wooden floor material with Diffuse, Normal, Roughness and Metallic maps

3.5 Texture Optimization

A realistic project needs realistic materials with high quality and high-resolution textures. The largest texture resolution UE4 supports is 8192x8192 pixels. In theory, every object and material can have this size textures. But, when the number of materials grow high,

eventually the computers memory or graphics card's memory will run out. For this reason, optimization has to be done.

UE4 has a clever method for handling large textures. If a texture is imported and has its resolution in 2^n x 2^n size (such as 1024 x 2048), Mipmaps are created. UE4 creates smaller versions of the same texture by making its resolution half and quarter and so on. As an example, if a 1024x1024 pixel texture is imported, Mipmaps with 512x512, 256x256, 128x128, 64x64 and 32x32 are created. How many Mipmaps created depends on the setting.

After the Mipmaps are created, when the object with the material using the created texture is on the screen, how much of the screen space the object is covering is calculated. After this calculation, the maximum size of texture that fits in this screen size is selected. In short, if object comes close to camera and gets larger portion of the screen, a higher resolution texture will be used, and further object will use smaller texture. Combined with Texture Streaming, a technique to load and unload needed textures on the fly, the memory issue is solved. [23]

UE4 also recently implemented a new technique called Virtual Texturing. With Virtual Texturing, when a large texture is used for an object, but only a smaller portion of the texture is seen and used, UE4 will cut out the needed part and load only that part of the large texture to save memory. This helps especially with landscape textures. [24]

If a texture does not have its resolution in power of two, Mipmaps are not created and only the original file is used. For this reason, with every new texture is imported, they must be checked that their size is on power of two and if not, they should be scaled.

3.6 Lighting Mobility

On UE4 there are 3 settings for mobility of the light. These are; static, stationary and dynamic.

When a light is added and is set to static, the lighting needs to be built for that light to work. Pre-built static lights are very easy on the CPU. But they need to be rebuilt every

time the scene changes. During the design process, static lights can be let without building them. When the project is finished, then lighting can be built once for final version. Depending on the size of the project and quality set for shadows, this might take from minutes to hours. UE4 builds shadow maps to be applied over the original textures for showing where shadows are. Pre-built light shadows can be sharp and good quality but since they are static, the light cannot be changed. It can't be moved or turned off. However, if the light will not change during gameplay, it is advisable to set it to static. [25]

Stationary lights are similar to static lights. They need to be built. The difference is, they can be manipulated during gameplay. They can be turned off; color and brightness can be changed. However, they have a limitation. No surface on the project can have more than 4 stationary light affecting it at the same time. This means, if there is a room and there are 5 lights and their area of effect overlap, one of the lights will not work. (image 4) When using stationary lights, the placement of the lights needs to be arranged so that they don't overlap. [26]

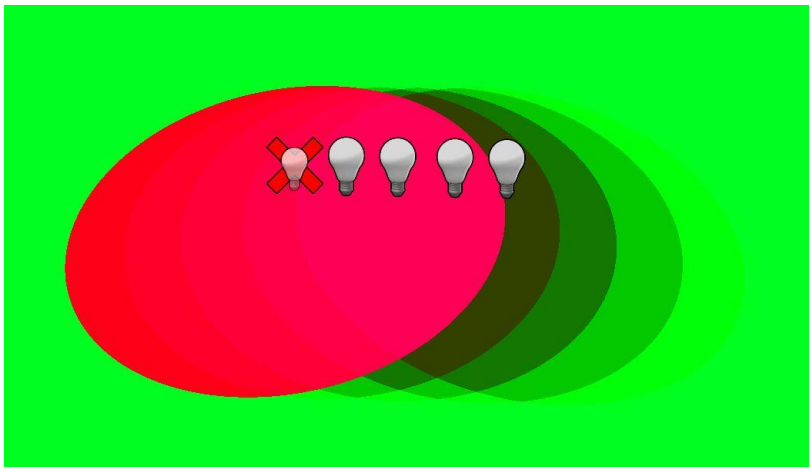

Image 4. No more that 4 Stationary lights can overlap

Movable lights are most performance heavy option for lighting the scenario. All the shadows are calculated during gameplay. The advantage of using movable lights is that they are fully dynamic. They can be modified, moved around and turned off. In a scene where there is daylight and the sun moves during gameplay, sunlight's mobility needs to be set to movable. [27]

3.7 Lighting Types

There are different types of lights on UE4: [28]

- Point Light is a sphere-shaped light. The size of the sphere can be changed.
- Rect Light is a rectangular shaped, one or two-sided light.
- Spot Light have a target location where they point at.
- Directional Lights is an unlimited sized parallel light that come from outside the project map. It can be used for sunlight.
- Sky Light covers the sky and illuminates from all directions.

Point, Rect and Spot lights can have IES profiles which defines what specific shape these lights look like. Many IES profiles can be found on the internet for free.

UE4 has a Sky Sphere blueprint in its Starter Content. This blueprint is a combination of Skylight, a Sun and clouds. Using it is an easy way to make simple sky.

In the project that is documented in this thesis, a Sky Sphere Blueprint is used for illumination of the sky. A Directional Light is used for illumination from the Sun. In front of the windows, Rect Light's are placed for better quality lighting for the interior. Point Light's are placed for lamps. Spot Light's are placed for spot lights on the ceiling (image 5).

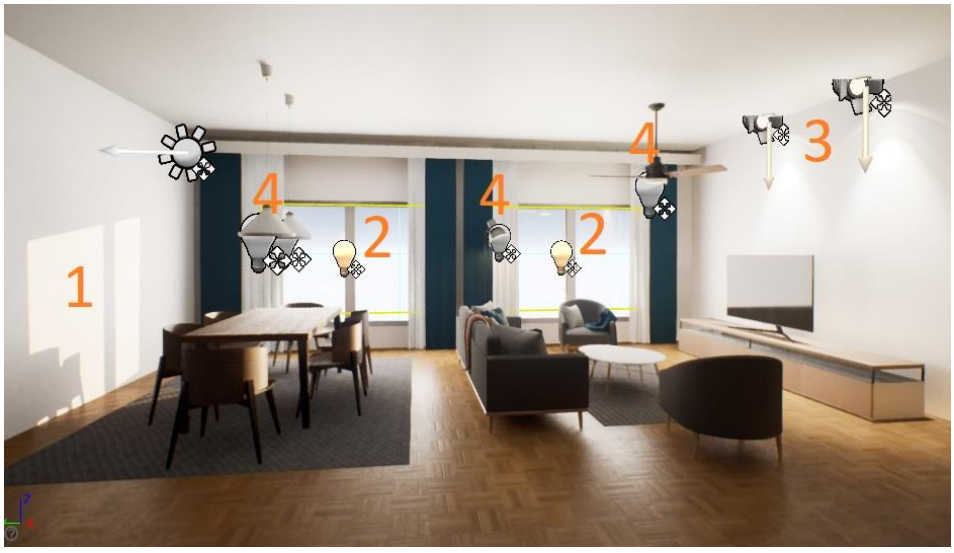

Image 5. 1. Directional Sun Light 2. Rect Lights for windows 3. Spot Lights 4. Point Lights

All lights for Regular Map are set to Movable because the lighting and shadows are being handled by Ray Tracing. In VR Map, Sky Sphere, Directional Light and Rect Light's are set to Static because they won't change. Point Light's and Spot Light's are set to Stationary so that they can be turned on and off during gameplay. However, their area of effect are carefully adjusted so that no more than 4 of Stationary lights are overlapping. When the project is finished, a final lighting build is performed for the VR Map with its Lighting Quality setting set to Production.

3.8 Reflections

Reflections are very important for realism. Every surface in real life have some level of reflection. Most surfaces are rough, meaning their reflection is blurry and spread. While setting these materials is easy, the amount of needed calculation for GPU is heavy. There are different ways to battle this on UE4.

3.8.1 Ray Traced Reflections

Ray Traced Reflections are latest and most accurate ways to calculate reflections. Rays are thrown from the camera and their path is traced. How many rays are thrown, how far they are traced and how many times they bounce depend on the settings and higher values require more power from GPU. Static stills can be calculated with very high settings and the amount of time it takes to calculate would not matter too much. However, in a real-time application, the speed is very crucial. [29] Low quality settings or slow GPU can produce unwanted visible noise (seen in image 6).

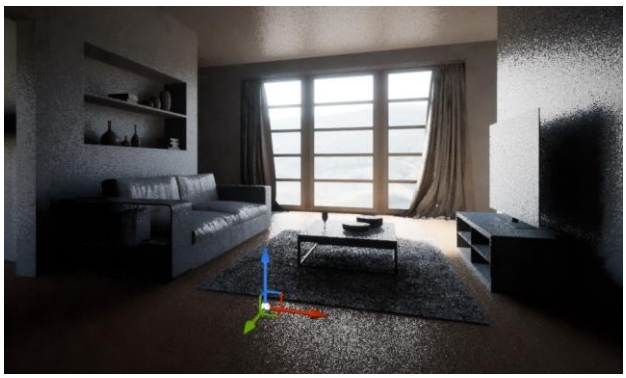

Image 6. Visible noise because of low settings or slow performance

With current speed of GPU's, some level of quality can be produced. But the future will bring much higher quality with increasing power of newer GPU's. At the moment, this technique is in early development and experimental.

In this project, Regular Map will use Ray Traced Reflections.

3.8.2 Screen Space Reflections (SSR)

Screen Space Reflections are default method for UE4 to achieve realistic reflections. They are medium performance heavy. With them, reflections are somehow blurry and not sharp. However, they are useful for floors which have blurry reflections. The way SSR works is that the objects that are in reflections must be themselves on the view. If there is a flat perfect mirror and the user looks directly to it, the objects that are behind the user and not in user's first view, they also won't be visible in the mirror (seen in image 7). Also, since SSR are not very sharp, a mirror wouldn't look very realistic. But for blurry surfaces, SSR are useful. [30]

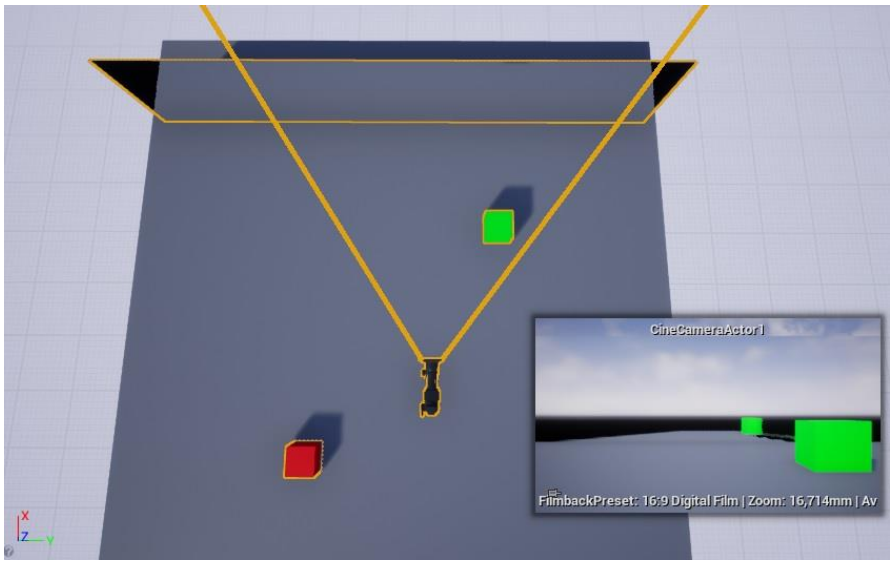

Image 7. Red cube is not in view of camera, thus also not in the reflection

In this project, VR Map will use SSR for blurry surfaces such as wood.

3.8.3 Reflection Capture Actor

Reflection Capture actor is a light performance method for good looking reflections. When this actor is placed, a build phase is needed. When reflections are built, they are recorded from the actor's perspective. The actor is either a sphere or a box.

Reflection Capture actor has its resolution set in the settings. If there are many of this actor and their resolution is set very high, GPU memory can run out resulting a crash.

While Reflection Capture actors are cheap and easy, their problem is that the reflections are always from the actor's perspective, meaning the object that is reflecting will always have incorrect reflection angle to the user (seen in image 8). For this reason, they are usually used on places that the reflections are not on a perfect mirror or flat and angle cannot be visible too clearly. [31]

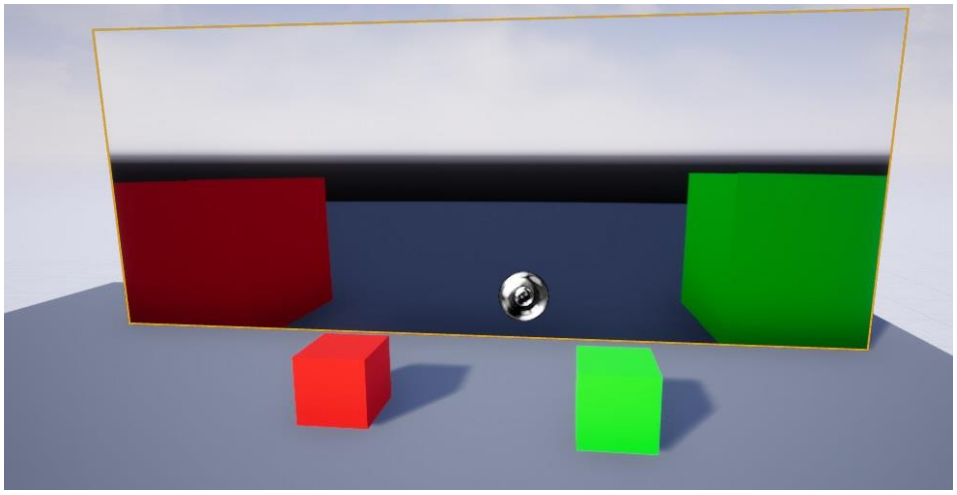

Image 8. Incorrect reflections from Reflection Capture actor

In this project, on the VR Map, there will be several Reflection Capture actors in the rooms to hide SSR imperfection on the corners of the view.

3.8.4 Planar Reflections

Planar Reflections are very performance heavy, but they produce very sharp reflections. UE4 needs to redraw everything a second time to achieve this. The upside is that they

are a perfect mirror. A flat mirror in a bathroom can use this type of reflection if there is no performance issue. Water surface is also a good place to use Planar Reflections. [32]

While Planar Reflections are great for sharp reflections, they cannot make blurry reflections. Because Planar Reflections are very performance heavy and they cannot make blurry reflections, they won't be used for this project.

3.9 Programming

As previously mentioned, because of the different functionality for user operation and performance issues of VR, two different maps will be created with two different settings and user inputs.

3.9.1 Regular Map

In Regular Map, keyboard and mouse are used to move around and interact. A player pawn blueprint is created and named "FirstPersonCharacter". A GameMode a blueprint is added, and player is changed to FirstPersonCharacter. In the World Settings, game mode is set to the new GameMode. In Project Settings on Input tab; keyboard buttons "E" key is added to Action Mappings as "EditButton". Mouse input and movement keys (W, A, S, D and arrow keys) are added to Axis Mappings.

Inside FirstPersonCharacter, a camera is added. Movement inputs that are added to input list are attached to Add Movement Input nodes. Mouse movement is attached to Add Controller Pitch and Yaw Input nodes. The FirstPersonCharacter blueprint is added to level. At this stage, when Play button is pressed, gameplay starts, and player is able to move around with keyboard and look around with mouse.

An int variable is added. If the int value is set to 0 it is "WalkingMode". If it's set to 1 it is "EditModeEnabled" and if set to 2 then it is "MoveModeEnabled". When EditButton is pressed, this int changes its value between 0, 1 and 2. LineTraceByChannel node will enable the user to select objects. When EditModeEnabled is selected, at FirstPersonCharacter camera location towards forward vector, a line is drawn by UE4 and the

object hit is printed to debug. This will later be programmed to allow user to select objects that have editing possibility and change their attributes.

Before adding functionality to change materials, the apartment's design need to be finished first. Either from Marketplace, internet sources or modelling by hand on 3ds Max, the whole apartment is filled up with furniture with high details and high-quality materials. Most of the objects have PBR materials with diffuse, metallic, specular, roughness and normal textures attached. Glass has refraction and opacity. A spinning ceiling fan has added to express movement of time because most of the objects are still.

For some of the objects such as wood, table and sofa, several materials have been created. Inside the FirstPersonCharacter blueprint, a function is added when EditModeEnabled is active and player clicks on an object, the object name gets through a Switch node. If the object name matches with the multiple material available furniture, the furniture gets outlined with a light blue color. This indicates that furniture material is switchable. A UI layer appears on the screen and shows the available options for material selection. When one of the materials is selected, Change Material node is called, and different material is applied.

When MoveModeEnabled is selected, the user can click on several objects to move their positions. When mouse is clicked, the object sticks to the position of the mouse and if mouse wheel is turned, it rotates.

With these functions are added, user is able to walk around, change materials and move objects. For this project, these are the only functions necessary.

3.9.2 VR Map

When Regular map is finished, it is copied and named VR Map. In this version keyboard functionality will be switched to VR Controller.

In the VR Starter project, these is a VRController blueprint with pickup and move around functionalities. This blueprint is migrated and added to VR Map. The buttons for activating teleport and pickup functions are added to Input Settings. Also, an Edit Mode function is added to top button.

The FirstPersonCharacter blueprint has the editing functions. These functions are copied to VRController blueprint and modified to be activated by the VR buttons. Moving around with keyboard buttons function is not copied because VRController already has a teleporting function.

When EditModeEnabled is selected and UI node appears on the screen, it is adjusted for a monitor screen. In VR Map, this node is modified to spawn as a real 3D object in front of the furniture being modified.

With these changes, VR Map is ready. User can teleport around the apartment, change materials and pickup objects to move them around.

As final steps, the lighting is built with Production quality is selected. After this is done, the project is cooked for Windows 64-bit and is able to be run directly from the executable file.

4 Conclusion

Virtual Reality is a developing and exciting technology. Computers are getting faster each year. More and more applications are being available to developers. With more sources available, community around CGI, VR and development is growing. Combined with other practices such as architecture, realism and functionality of future projects will only get better.

In this thesis, an example of a realistic looking VR application for an architect to show his design is created. Problems of VR is discussed. Free community resources are used. Steps to reach realism are explained. Latest real-time graphics with Ray Tracing is used. Limitations of performance with VR is solved by using pre-built lighting and Screen Space Reflections.

This project will be delivered to architecture companies around Finland in order to create collaborations.

References

1 All About Architecture, What is Architecture? 31 July 2018
URL:
Accessed: 06.05.2020

2 RedAlkemi, Pros & Cons of Internet of Things, 7 May 2018
URL: https://www.redalkemi.com/blog/post/pros-cons-of-internet-of-things
Accessed: 06.05.2020

3 WHO, Coronavirus Disease (COVID-19) Advice for the Public, 29 April 2020
URL: https://www.who.int/emergencies/diseases/novel-coronavirus-2019/advice-for-public
Accessed: 06.05.2020

4 Virtual Reality Society, What is Virtual Reality?
URL: https://www.vrs.org.uk/virtual-reality/what-is-virtual-reality.html
Accessed: 06.05.2020

5 Vilayanur S. Ramachandran, Diane Rogers-Ramachandran, Two Eyes, Two Views: Your Brain and Depth Perception, 1 September 2009
URL: https://www.scientificamerican.com/article/two-eyes-two-views/
Accessed: 07.05.2020

6 Antonin Artaud, The Theatre and its Double, Translated by Mary Caroline Richards, New York: Grove Weidenfeld, 1958

7 David Heaney, How VR Positional Tracking Systems Work, 29 April 2019
URL: https://uploadvr.com/how-vr-tracking-works/
Accessed: 07.05.2020

8 Matt Wales, Eurogamer, HTC Unveils Three New Versions of Its Vive Cosmos VR Headset, 20 February 2020
URL: https://www.eurogamer.net/articles/2020-02-20-htc-unveils-three-new-versions-of-its-vive-cosmos-vr-headset
Accessed: 06.05.2020

9 Robbie Collin, Ready Player One Review: Steven Spielberg's White-knuckle Ride Through a World of Pure Imagination, 29 March 2018
URL: https://www.telegraph.co.uk/films/0/ready-player-one-review-urgent-message-steven-spielberg-eternal/
Accessed: 06.05.2020

10 Tony Hopkins, Designblendz, What Is Architectural Visualization? 16 August 2018
URL: https://www.designblendz.com/blog/what-is-architectural-visualization
Accessed: 06.05.2020

11 Arch2o, How to Make an Impressive Architecture Model? Your Complete Guide
URL: https://www.arch2o.com/architecture-model-complete-guide/
Accessed: 06.05.2020

12 TMD Studio LTD, Virtual Reality Uses in Architecture and Design, 21 January 2017
URL: https://medium.com/studiotmd/virtual-reality-uses-in-architecture-and-design-c5d54b7c1e89
Accessed: 06.05.2020

13 Jorge Fontan AIA, 5 Architecture Phases of Design Explained, 20 March 2016
URL: https://jorgefontan.com/architectural-design-phases/
Accessed: 06.05.2020

14 BIM Service India, What Is Revit Software and Why It Is So Essential to Architects, Engineers & BIM Professionals? 22 August 2019
URL: https://www.bimservicesindia.com/blog/what-is-revit-software-and-why-it-is-so-essential-to-architects-engineers-bim-professionals/
Accessed: 06.05.2020

15 Thomas Denham, What is Unreal Engine?
URL: https://conceptartempire.com/what-is-unreal-engine/
Accessed: 07.05.2020

16 Epic Games, Features of Unreal Engine
URL: https://www.unrealengine.com/en-US/features
Accessed: 07.05.2020

17 Real-Time Ray Tracing
https://docs.unrealengine.com/en-US/Engine/Rendering/RayTracing/index.html
Accessed: 15.06.2020

18 Datasmith Overview
https://docs.unrealengine.com/en-US/Engine/Content/Importing/Datasmith/Overview/index.html
Accessed: 15.06.2020

19 Post Process Effects
https://docs.unrealengine.com/en-US/Engine/Rendering/PostProcessEffects/index.html
Accessed: 15.06.2020

20 Ray Tracing Features Settings
https://docs.unrealengine.com/en-US/Engine/Rendering/RayTracing/RayTracingSettings/index.html
Accessed: 15.06.2020

21 Essential Material Concepts
https://docs.unrealengine.com/en-US/Engine/Rendering/Materials/IntroductionToMaterials/index.html
Accessed: 15.06.2020

22 Material Inputs
https://docs.unrealengine.com/en-US/Engine/Rendering/Materials/MaterialInputs/index.html
Accessed: 15.06.2020

23 Texture Properties
https://docs.unrealengine.com/en-US/Engine/Content/Types/Textures/Properties/index.html
Accessed: 15.06.2020

24 Virtual Texturing
https://docs.unrealengine.com/en-US/Engine/Rendering/VirtualTexturing/index.html
Accessed: 15.06.2020

25 Static Lights
https://docs.unrealengine.com/en-US/Engine/Rendering/LightingAndShadows/LightMobility/StaticLights/index.html
Accessed: 16.06.2020

26 Stationary Lights
https://docs.unrealengine.com/en-US/Engine/Rendering/LightingAndShadows/LightMobility/StationaryLights/index.html
Accessed: 16.06.2020

27 Movable Lights
https://docs.unrealengine.com/en-US/Engine/Rendering/LightingAndShadows/LightMobility/DynamicLights/index.html
Accessed: 16.06.2020

28 Types of Lights
https://docs.unrealengine.com/en-US/Engine/Rendering/LightingAndShadows/LightTypes/index.html
Accessed: 18.06.2020

29 Real-Time Ray Tracing
https://docs.unrealengine.com/en-US/Engine/Rendering/RayTracing/index.html
Accessed: 19.06.2020

30 Screen Space Reflections
https://docs.unrealengine.com/en-US/Engine/Rendering/PostProcessEffects/ScreenSpaceReflection/index.html
Accessed: 19.06.2020

31 Reflection Environment
https://docs.unrealengine.com/en-US/Engine/Rendering/LightingAndShadows/ReflectionEnvironment/index.html
Accessed: 19.06.2020

32 Planar Reflections
https://docs.unrealengine.com/en-US/Engine/Rendering/LightingAndShadows/PlanarReflections/index.html
Accessed: 19.06.2020