Development of USTH interactive virtual world using Unreal Engine

DO Duy Huy Hoang

University of Science and Technology Hanoi

ICT Department

2019-06-18

Declaration

I declare that the thesis is entirely my own under the guidance of Dr. Tran Giang Son. I certify that the work and result are totally honest and unprecedented previously published in the same or any similar form. In addition, all assessments, comments and statistics from other authors and organizations are indicated and have been cited accordingly. If any fraud is found, I will take full responsibility for the content of my thesis.

Acknowledgement

Firstly, I would like to express my thanks to my research supervisor Dr. Tran Giang Son for giving me the opportunity to do research and providing invaluable guidance during this internship. Secondly, my sincere thanks also goes to Dr. Nghiem Thi Phuong for her encouragement, insightful comments and hard questions during the research. Thirdly, I thank my fellow labmates, friends in ICTLab: Kieu Tran, Linh Loc, An Luu, Duc Pham, Trung Bui for the stimulating discussions and for all the fun we have had in the last three years. I also very appreciate to have been a student at USTH where all professors and staffs are always willing to support their students. Besides, I would like to thank the Unreal Engine and Sketchup community for answering and providing many valuable knowledge I asked during the research. Last but not least, I express my deep and sincere gratitude to my parents who raise me up throughout my life by their patient and their love.

List of Tables

List of Figures

Contents

I. Introduction

1.1 Context and Motivation

A virtual world or a digital world is a computer-based online community environment that is designed and shared by individuals, or may be populated by many users who can create a personal avatar so that they can interact in a custom-built, simulated world. These avatars are graphically rendered using computer graphics imaging (CGI) or any other rendering technology which can make people feel like they are in the real world. Individuals control their avatars using input devices like the keyboard, mouse and other specially designed command and simulation gadgets.

Today's virtual worlds are purpose-built for entertainment, social, educational, training and various other purposes. It is being more better than the real world. So the virtual reality has take over everything. For instance the illustrations showed in a game are more better than the reality we live in. It provides the gamer a whole new experience by taking over his imaginations it's as if we are playing the real game with all those real gaming characters in the real life. Virtual world is bringing out all our imaginations so that we can experience it in reality. Similarly for the automobile industry we can take a virtual ride of the desired vehicle we want to buy which provides a whole new dimension to the world of buying things.

Moreovers, virtual world has been adopted in education for teaching and learning situations. Students are able to interact with each other and within a three dimensional environment. Students can also be taken on virtual field trips, for example, to museums, taking tours of the solar system and going back in time to different eras.

1.2 Objectives

Develop an interactive 3D virtual world of USTH building from easily captured 2D images. Specific goals include - Collect a dataset of 2D images from cameras for different views of USTH - Conduct fully 3D models from collected 2D images which will be a 3D model foundations for other reaseacher or developer to create others world of

USTH - Build a realtime and interactive 3D virtual world of USTH from constructed 3D models.

1.3 Thesis structures

II. State of the art

III. Programs, Materials, Methodologies

3.1 Programs

3.1.1 Sketchup

SketchUp is a 3D modeling computer program for a wide range of drawing applications such as architectural, interior design, landscape architecture, civil and mechanical engineering, film and video game design. This program was used in this project in order to create and conduct a fully 3D models from scratch. Therefore, they also provide an open library called 3D Warehouse which SketchUp users may upload and download 3D models to share. The models can be downloaded right into the program without anything having to be saved onto your computer's storage and it is free so anyone can download files for use in SketchUp or even other software such as AutoCAD, Revit and ArchiCAD.

3.1.2 3DS Max

Autodesk 3ds Max is a professional 3D computer graphics program for making 3D animations, models, games and images. With this feature, the software will help me to correct the position of the texture and also to edit UV mapping of some models.

3.1.3 Substance Painter

Substance Painter gives me all the tools I need to texture my 3D assets and also this software can help me to bake a normal map and metalic map.

3.1.3 Unreal Engine

3.2 Materials

3.3 Work Breakdown

3.3.1 Work Breakdown Diagram

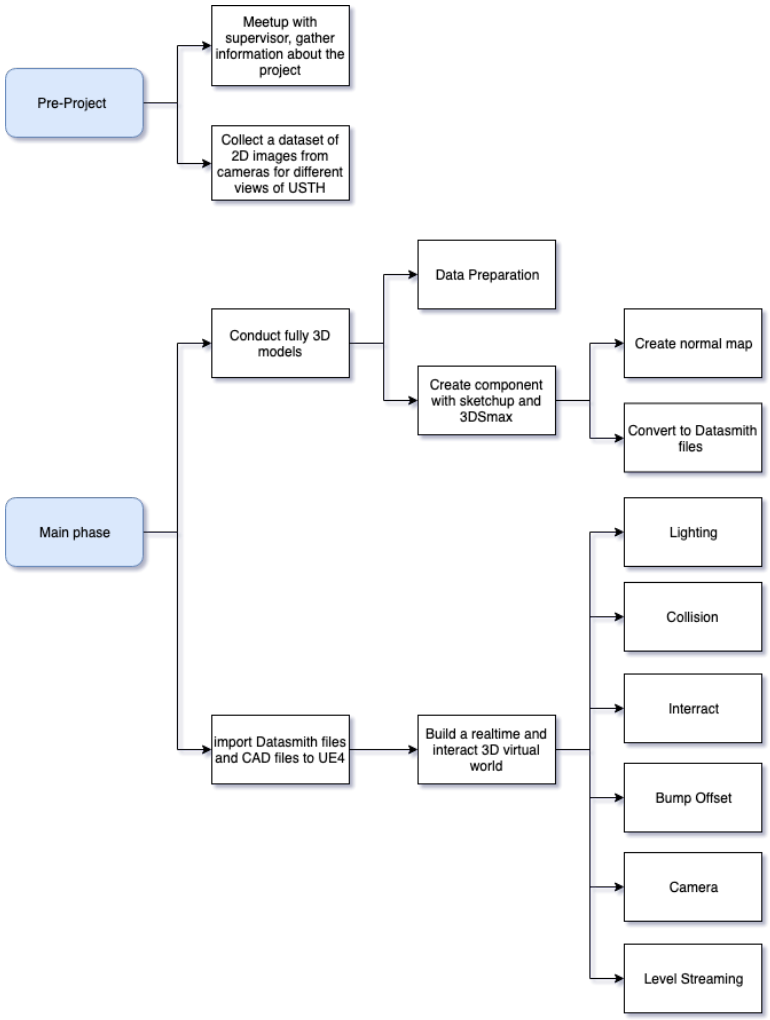

Figure 1: Work Breakdown Diagram

3.3.2 Work Breakdown Description

This section is a visualiazation and a brief summary of the work, We shall be discussing these terms in depth in the methodology section.

a. Pre-project

- **Conduct a meeting with the participant** of the Planning Phase, which involves setting expectations, articulating likely risk, etc.
- **Develop Work Breakdown Structure:** break the project into smaller, measurable of work packages with a work breakdown structure.
- **Collect a dataset** of *2D images* USTH: We will use a simple camera from a smartphones to perform this task. After taking picstures with different view and also furnitures of USTH, these photos will be classified to specific folder (e.g: Room 5/Lab ICT). This approach has two drawbacks howerver, it is time-consuming and easy to make mistakes.

b. Main phase

- **Conduct** fully 3D models from collected 2D images which will be a 3D model foundations for other reaseacher or developer to create others world of USTH.

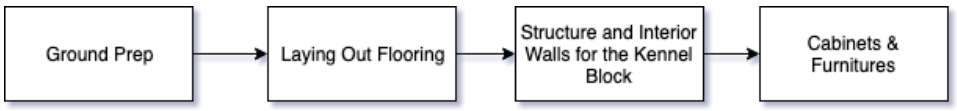

Figure 2: 3D modelling steps

First of all we need to create a Floor Plans. We use a 2D legacy floor plans image and transfer it to a digital layout as the figure below.

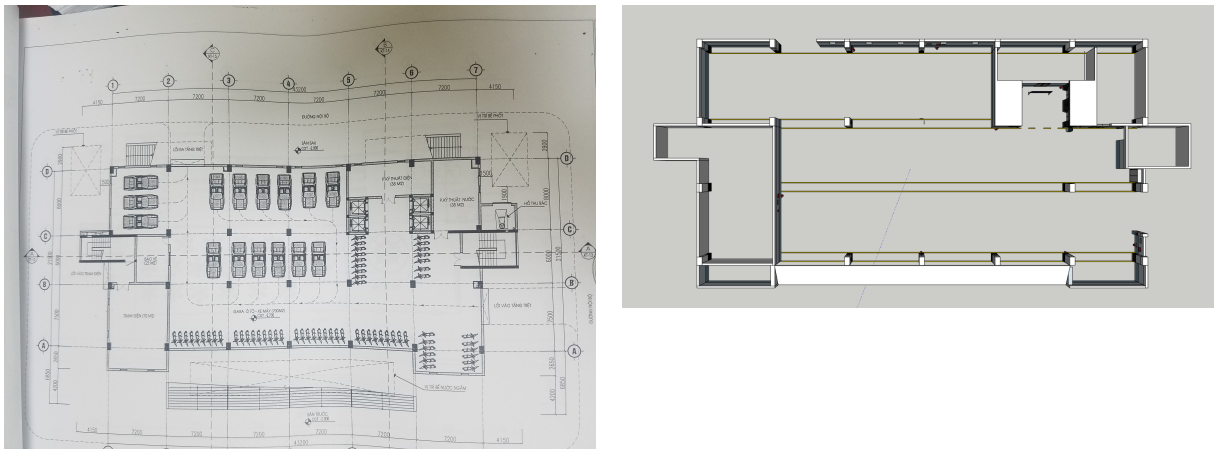

Figure 3: Ground Prep

After created a floor plans with sketchup layout, we start to structure and interior walls and Finally in order to make the 3D model complete, we import sketchup textures, create custom materials and also add furniture objects. These furnitures can be found in the 3D warehouse store which comes with SketchUp when installed. Sometime I also need to create some special models in USTH thus it cannot be found in the store.

Figure 4: Structure wall and Completed model example

To finish the 3D models and prepare models for Unreal Engine, the final step is to turn a model into a normal map for exporting to be used in 3rd party renderers. We use

Substance Painter application and also a plugin called **CLF Normal Map Maker** from Sketchup. It will paint the model as a normal map, export an image, and then revert all materials back to what they were.

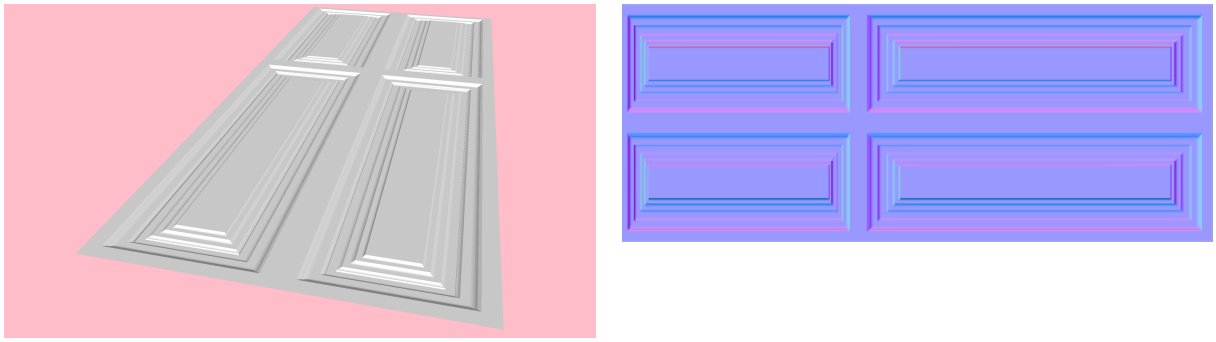

Figure 5: Normal map example

- **Preprocess data**: Before we can get started working with SketchUp content in the Unreal Engine, we need to install the Unreal Datasmith Exporter plugin for SketchUp and able to export a scence from SketchUp as a *.udatasmith* file and also *.fbx* files, thus the previous application creates models in a .SKP file.
- **Build a realtime and interact 3D virtual world**: We start making our 3D virtual world by adding player, light, interact, and some technique like *level streaming*....

3.3 Methods

This section will explain all the methods of the work.

3.3.1 Preparation

Unreal Engine can be dowloaded from the official website - it is free for everyone. As I mentioned before, to be able to run Unity, and create a new project, we'll need at least direct x11 for simulation of lighting, shadows on the scene.

Start with importing our 3D models to UE4. Datasmith brings structure data from an combination of source applications into Unreal Engine, frequently to fabricate steady representations and experiences around that data. In any case, routinely, whereas you are working on buiding those visuallizations and experiences in UE, our sence needs to be changed in order to meet new requirements or incorporate feedback from stakeholders.

In our 3D models, i tried to making as many as repeating elements as possible such as windows, doors, chairs, thus when Datasmith detects multiple copies of the same component in your SketchUp scene, it only creates one set of Static Mesh assets for that component, and places multiple instances of those Static Meshes into the scene. So that it is typically better for the runtime memory requirements and performance, as well as making it easier to manage the number of Static Mesh assets.

Units and Scale In the Unreal Engine, all separations are constantly estimated in centimeters. However, other 3D design applications typically offer a choice of units of measurement. But Datasmith naturally deals with modifying the size of our scene so our geometry shows up at the very same genuine size in the Unreal Engine, and at the correct areas in 3D space. So we do not need to change anything from the previous work with Sketchup application.

Materials The Datasmith import process creates a new Material Asset in the Unreal Engine Project to represent ech different set of geometry surface properties it recognizes in the scene it imports. But we also need to tweak these Materials after import thus most of the materials assets are *Material instances*, it is referred to as the Parent Material, which can be used as a base to make a wide varitety of different looking Materials.

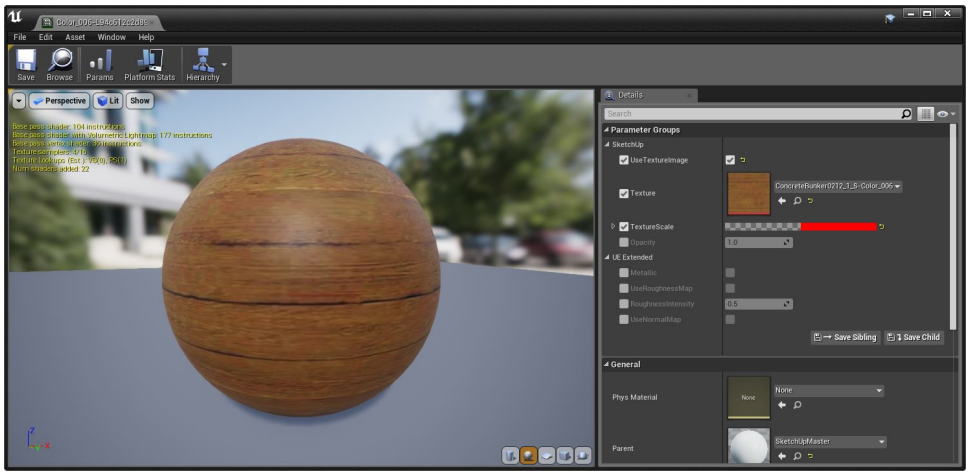

Figure 6: Unreal Engine Materials

The Datasmith Color Material Sketchup use simple surface color to shade geometry so we can easily use Datasmith exporter to brings these surfaces into Unreal as Instances of the *Datasmith-color Material*. The color of this Material is typically pre-set to exactly match a color from our Sketchup models, but as i mentioned before, we sometimes need to change/tweak some values. For example the brightness of the color maybe too bright so we need to reduce in order for our lighting to look realistic.

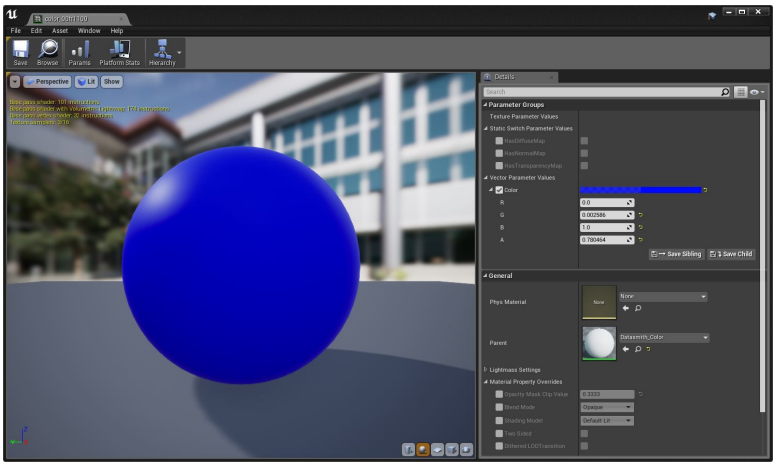

Figure 7: Unreal Engine Object Color

Layer While creating 3D models with SketchUp, I use *Layer* tool a alot to organize the content, eg: a roof-floor will be a layers. Datasmith preserves that organization in the Unreal Editor so that we can easily show or hide layer to quickly find and select all objects in a layer.

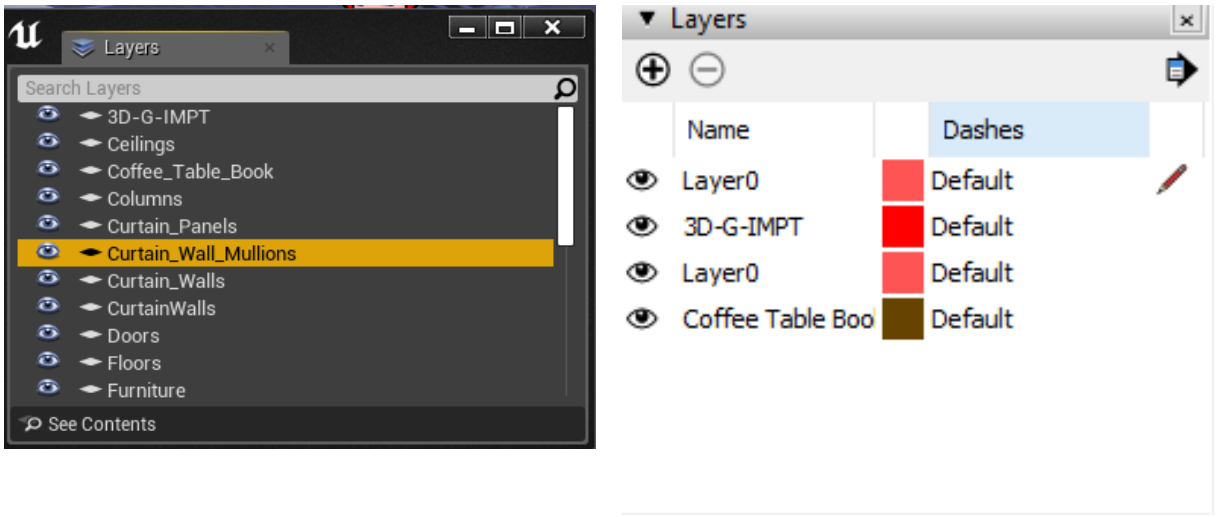

Figure 8: Left: Layer in UE4. Right: Layer in SketchUp

The 3D models files might be split into smaller part. For example, as I mentioned earlier, we will try to reduce the object as much as possible to increase performance, so i make as many as repeating elements as possible and convert these model to *Filmbox* file type to import to Unreal Engine. We still keep the floor plan, walls and the basic hallway then import to UE4 but all the furnitures will be split into individual files so that we can only need to import these files once. These pictures below is an example that i create a door and student table using 3DSmax and then import back to UE4. This technique not only help me to improve the application performace using *Level Streaming method* but also by making individual object with 3DSMax, i can control the pivot of these object which cannot be done in Sketchup.

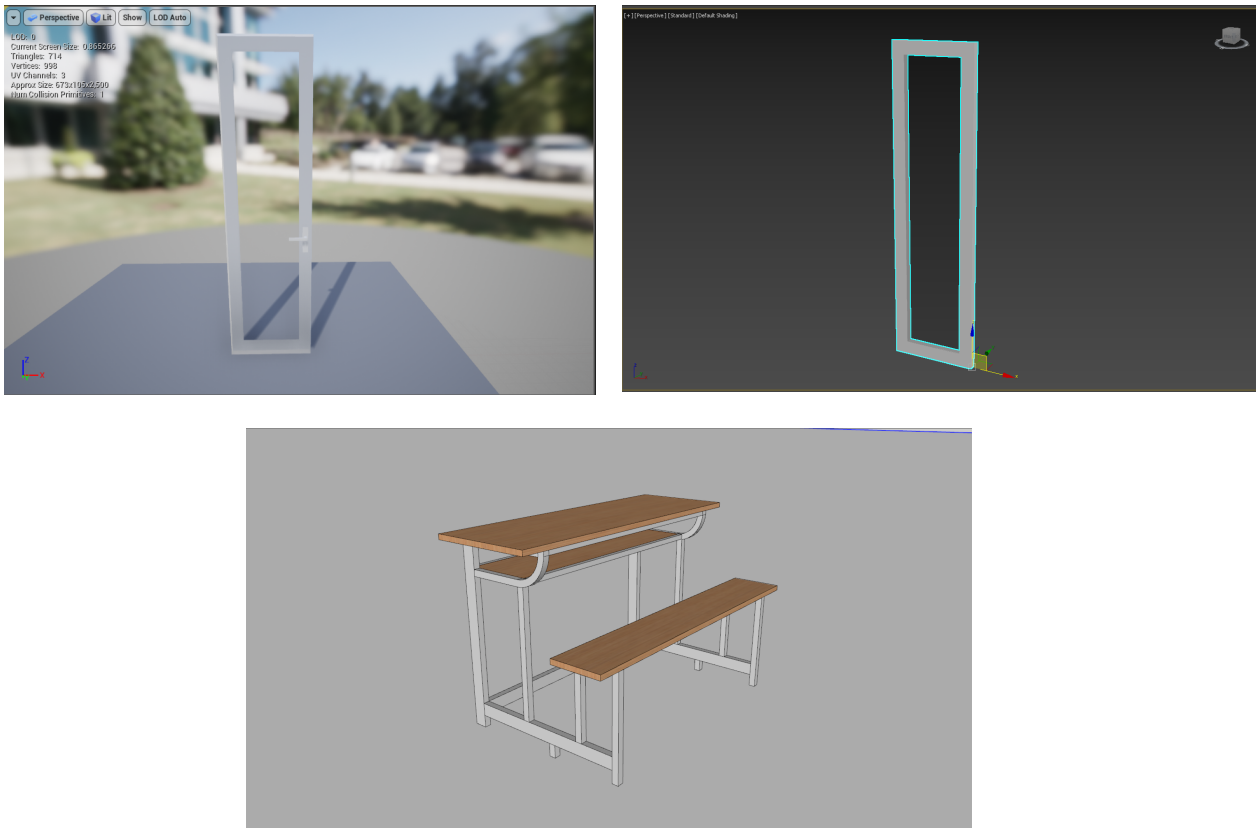

Figure 9: 3D object in UE4 and 3DSmax

3.3.2 Player/Cameras

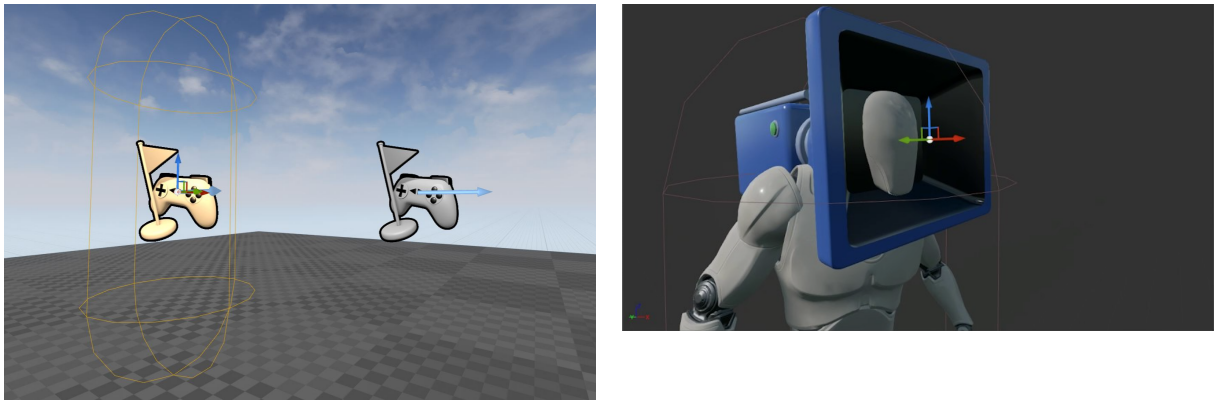

Figure 10: Player start and Cameras actors

To be able to interact with the objects, a player should be added into the scence. Initially the player will not move without any script attached to it, and the most basic way to do this is through Camera Actors. The Camera Actor can be found in the Modes panel under the All Classes category. To place one into the world, drag it from the Modes panel into the game world. Using a Camera Actor is as easy as selecting it from the Modes panel and then dragging it into the world.

Thus my main purpose of this application is for students can be taken on virtual school trip before they apply to USTH, so that I only need to make my application as a first person view game.In order to do that I just attached a Camera actor into the player, once we have the Camera Actor in the world, I can then use it in combination with the Blueprints or Matinee to use it as a view point for my level.

Because when I created 3D models with SketchUp, I also saved a lot of camera views with *Scences*. So that when developing for this application in UE4, these views is also included at the time we converted these models, and now i can let user change their starting point with some specific hotkeys.

3.3.3 Collision/RayCasting

Collision Response form the basis for how Unreal Engine 4 handles collision and ray casting during run time. Every object that can collide gets an Object Type and a series of responses that define how it interacts will all other object types. When a collision or overlap event occurs, both (or all) objects involved can be set to affect or be affected by blocking, overlapping, or ignoring each other. With the ability to get information from the object that the player affect, this is the main method for this project, as the character is able to interact with other objects in the virtual world.

3.3.4 Interact/Box trigger

3.3.5 Bump Mapping/Bump offset

In order to make the object surface look more realistic, bump mapping/bump offset is one of technique in computer graphic to do that simulating small displacements of the surface, create the illusions of depth and texture of a 3D model using computer graphics. Howerver, the number of polygons does not increase. The modified surface heavily relied on light reflection, define how the light should shine on the surface. Comparing to Displacement - one of another method for find geometric surface detail, it is much faster than displacement, no extra memory necessary so as i said before, the aim of this project is also focusing on performance but the quality is much lower. In this project, I will only using *Normal map* as a type of *Bump map*. They are a special kind of texture stores a color in each pixel of the texture that allow you to add surface detail such as bumps, grooves, and scratches to a model which catch the light as if they are represented by real geometry.

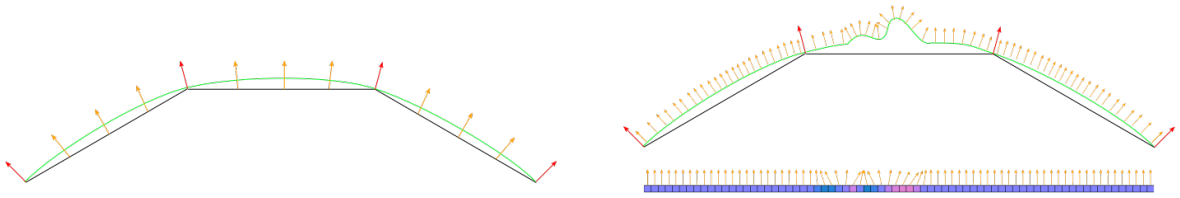

Figure 11: Left: Smooth shading on three polygons. Right:Normal mapping across three polygons / viewed as a 2D diagram

With the normal map, the surface will look like in the picture below, as how the light shines on the surface.

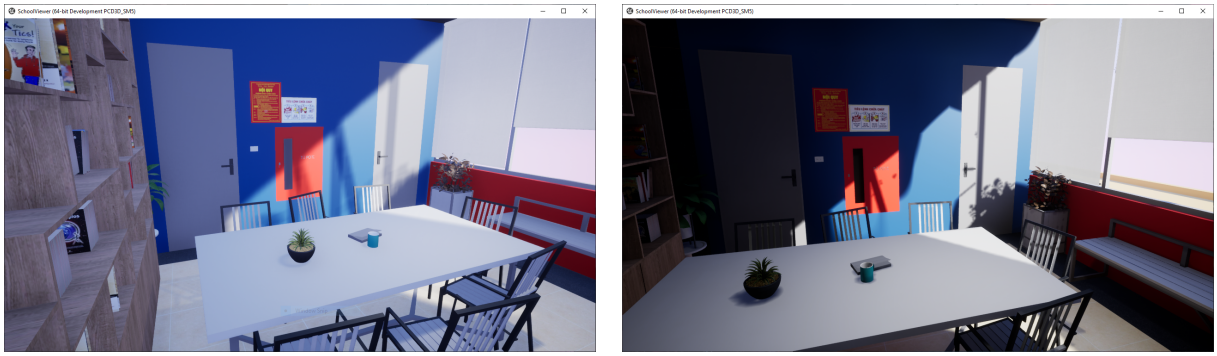

Figure 12: With/Without Normal map

3.3.6 Level Streaming

The Level Streaming feature makes it possible to load and unload map files into memory as well as toggle their visibility during play, making it easy to have large levels broken into pieces. You have just a small, whats referred to as a persistent level, to start and load other levels into and out of your persistent level as needed. So you set up a trigger so when the player walks through another part of the level is loaded. Is it a process

that can be seen so you must take care to make sure the player can't see that level until its completely loaded.

3.3.7 Lighting

Lighting is an essential part of every scene. In Unreal Engine, while the mesh, textures define the shape, the look of an object, the lights will define the color, the mood of a 3D environment. Lights can be added into the scene just by going to **Modes menu** -> **Light menu**. Different types of Light Components will allow you to add a light source as a sub-object to an Actor, depending upon the effect you aim to achieve. Once the light has been added into the scene, we can adjust the properties of the light to the way we want in the inspector, there are many different options within the Light Component. For example, we would like to adjust the position and rotation of the light using the position and rotation widgets like any other object.

There are some attribute that i will use as described in the table below.

Table 1: Lighting Attribute

Species	Attribute	Description
Intensity	Brightness	Intensity determines how much energy the light outputs into the scene. eg: 1700 lumens = 100W lightbulb.
Light Color	R/G/B	Light Color will adjust the color of the light and the sprite that represents the light in the editor will change its color to match.
Attenuatic Radius	Source radius Source lenght	Define the size of specular highlights on surfaces.

IV Result and Discussion

V. Conclusion and Future work