



TAMPEREEN TEKNILLINEN YLIOPISTO
TAMPERE UNIVERSITY OF TECHNOLOGY

GUILLERMO ROA SANTAMARÍA

**DEVELOPMENT OF DESIGN TOOLS FOR THE EVALUATION OF
COMPLEX CAD MODELS**

Master of Science Thesis

Examiner: Professor Asko Ellman
Examiner and topic have been
approved by the Council meeting of
the Faculty of Engineering Sciences
on 4th March 2015.

ABSTRACT

TAMPERE UNIVERSITY OF TECHNOLOGY

Exchange student

ROA SANTAMARÍA, GUILLERMO: Development of design tools for the evaluation of complex CAD models.

Master of Science Thesis, 62 pages, 1 Appendix page.

April 2015

Major: Mechanical Engineering

Examiner: Professor Asko Ellman

Keywords: Virtual Reality, Real-Time, SOLIDWORKS, UNITY, FBX-format, Plugins, Oculus Rift.

The goal of this thesis is to invent means to enable 3D-models, created in SOLIDWORKS CAD-software, to be explored in a low-cost head-mounted display device Oculus Rift. Such a design tool would make possible to study 3D CAD-models in Virtual Reality with a very low cost and share design information for people who are not so familiar with CAD-models. As a result this can lower the product development effort for new products enabling, lower working time and improve both the quality of the final product and customer satisfaction.

CAD-models as well as exported graphics format from SOLIDWORKS can't directly be imported to Oculus Rift device. Therefore transformation pipeline need to be invented between the SOLIDWORKS and Oculus Rift. There are some options for this and their issues are studied in this thesis. As a best option it was found to use a conversion plugin tool to SOLIDWORKS which convert CAD-models to FBX-format. This format can be imported to Unity game engine software which is supported by Oculus Rift device.

Graphics file conversion process is known to be imperfect. Therefore conversion of three SOLIDWORKS CAD-models have been analyzed; a Crane, a Pipe system, and a Hydraulic cylinder. This testing gave a good overview of possibilities and limitations of the graphics pipeline. It was found that ordinary CAD-models can be transferred to Oculus Rift without problems. However, there are some special features in CAD models which cause problems for in graphics file conversion.

PREFACE

This master's thesis was carried out at the Department of Mechanical Engineering and Industrial Systems (MEI) at Tampere University of Technology, (TUT), Finland. The supervisor of the thesis was Professor Asko Ellman.

I would like to thank him for giving me an opportunity to participate in an interesting new research project. Also for his help, advise comments and appropriate suggestions. I am also grateful to Mr. Ilari Laine for his help and work with the SOLIDWORKS files he provided me during my experimentation period. My special thanks to all of colleagues in our office for continuous support during this work.

Furthermore, I wish to thank my parents, my brothers and who supported me during my studies. As well as my university colleagues, my friends and all the great people I have met during this year of study in Finland who made possible a really awesome stay.

Tampere 20/04/2015

Guillermo Roa Santamaría
e-mail: guilleroasanta@gmail.com

TABLE OF CONTENTS

ABSTRACT.....	i
PREFACE	ii
ABBREVIATIONS.....	v
1. INTRODUCTION.....	1
1.1. Background	1
1.2. Target of the project.....	2
1.3. Graphics Pipeline	3
1.4. Research Methods	5
1.5. Thesis Structure	6
2. ELEMENTS OF GRAFIC PIPELINE.....	7
2.1. SOLIDWORKS Formats	7
2.2. UNITY 3D Formats	12
2.2.1. Importing 3D models to Unity	15
2.2.2. FBX & OBJ which format is better?	17
2.3. Oculus Rift Introduction.....	19
2.3.1. Oculus Rift User Interface	19
2.3.2. Oculus Rift Plug-in.....	22
2.4. Interchange Format	25
3. EXPERIMENTAL EVALUATION	27
3.1. Analysis of Exporting Options for FBX Files.....	27
3.2. Examples: crane, hydraulic cylinder, pipe system.....	27
3.3. Standard Verification of Compatibility	29
3.3.1. Geometric Level.....	29
3.3.2. Texture Level	31
3.3.3. Material Level	33
3.3.4. Scale Level	35
3.3.5. Light & Shadow Level	35
3.3.6. Camera Level	36
3.3.7. Animation Level	36
3.4. Evaluation of success	41
4. NURB LIMITATIONS	42
4.1. Differences between Nurbs and Polygons.....	42
4.2. Limitation, Main problem	44

4.3. Alternatives and Solutions	46
5. OPTIMIZERS IN UNITY.....	47
5.1. Optimizers and their use in Unity.....	47
5.2. Asset Store	53
5.2.1. Mega Fiers	53
5.2.2. Unilod.....	54
5.3. Manager Settings.....	54
6. CONCLUSIONS AND PERSPECTIVES.....	59
REFERENCES	60
APPENDIX	63

ABBREVIATIONS

3D 3-Dimensional

CAD Computer Aided Design

HMD Head Mounted Display

MEC Department of Mechanics and Industrial Systems

PC Personal Computer

TUT Tampere University of Technology

VR Virtual Reality

OR Oculus Rift

SW SOLIDWORKS

1. INTRODUCTION

1.1. Background

Virtual Reality (VR), as defined as a scientific and technical domain exploiting the possibilities of computers and behavioral interfaces to simulate in a virtual world the behavior of 3D entities, which interact in real-time with each other and one or more users in pseudo-natural immersion. [1] Virtual reality is considered to have wide ranging benefits for the healthcare sector but, it can be used in other sectors as well which include; education, architecture, military, gaming, the space, robotic devices, virtual prototypes of new vehicles, etc. The use of Virtual Reality applications increases, both commercially and non-commercially.

Virtual reality is often used to describe a wide variety of applications commonly associated with immersive, highly visual, 3D environments. For an increasing realism, some physical elements have been added to this technology, virtual prototypes. Virtual prototypes enable the evaluation of several design alternatives; they are useful in the design process helping the cooperation between designers and users, making those consumers, even though these non-technical people can understand and evaluate efficiently and successfully the product design. [2]

To make possible a correct integration into the virtual environment in our project is needed the implementation of a virtual reality platform composed of a 3D immersive display, and a software layer allowing the running of interactive real-time scenarios which enables the association of manipulation properties to the 3D objects stemming from a CAD model of an industrial system. This project focuses on HMD frame. HMD (Head-mounted-Display) is a significant tool to share with other operators or industrial sites the immersive visualization of the 3D digital model of the system to be easily understood; to interact with the 3D model in order to reach intuitive simulation and visualization taking advantage of the head-tracking sensor and the immersive possibilities it offers.

The methods and tools for 3D imaging and interactive simulations in virtual reality have now penetrated industrial activities. They are used in particular for the implementation of Computer Aided Design (CAD) models in manufacturing industries. Today, the 3D modeling and animation sketching is becoming one of the most important activity in the production of virtual reality application. [3] 3D designers or CAD engineers should follow a long pipeline, starting from the sketching to the final rendering or the object creation. Apparently, it seems to be important to find the equilibrium and compatibility between Virtual reality software and design programs

software in order to combine virtual reality with as much as possible industrial applications.

Into the field of the software this can be done through format conversion and standardization of formats. The aim of this is generate files that could be read by different design software and also be implement by virtual reality environment engine software as Unity 3D.

1.2. Target of the project

Within the environment of real-time three-dimensional modeling, there are many useful applications to model, texture and animate characters-humanoids or different industrial objects. Each of them has unique qualities, which is why numerous programs are used in this modeling field, not just one. But the most important limitation is the upgrade of interchange format-data exporters. It is very common that modelers need to transfer models made of one application to another without losing previous work.

Given these problematic, there is the possibility that a modeler use a different program to work, such as SOLIDWORKS design programs, to that used to export and prepare the models in real-time. So it is necessary to review different formats export of models that can be imported into various applications with minimal loss of information.

After research work, we reach the conclusion that UNITY game engine environment is working with FBX-format import files. That will be the properly way of import CAD SOLIDWORKS Models within Virtual Reality environment. This file facilitates the transfer of content, helping to develop three-dimensional content between applications.

Target of the project is to integrate SOLIDWORKS files into a functional real-time environment with a demonstration simulation in the Oculus Rift device.

1.3. Graphics Pipeline

The engineering design process is a creative process seeking to solve a problem or facilitate certain activities. When developing a new product the design is one of the main steps, it can be defined as an interactive feedback process and graphically represented in Figure 1.2

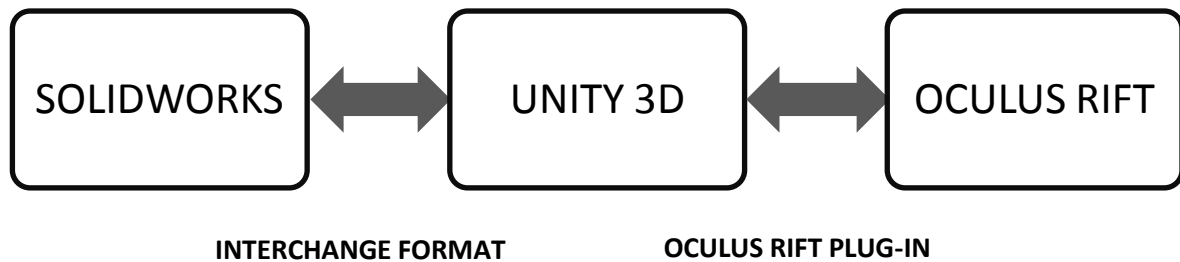


Figure 1.2 *Graphics Pipeline*

Starting with two well-defined conditions, such as: an initial CAD model created in SOLIDWORKS and Oculus Rift device, it aims to provide coverage for the system to function properly. To implement this design there is a wide range of possibilities, but agreeing on the need to use a virtual platform that allows us to integrate all the components required in our design. In turn, it should serve as a bridge between the two initial starting points.

Given some performance specifications for a device, a development virtual environment UNITY platform is selected mainly because support that the company Oculus provides, through Oculus Rift Plugin. Furthermore UNITY offers efficient capabilities and useful working tools. The last step before the simulation is the correct compatibility between software's so the interchange format plugins are implemented in this section to allow a proper and efficient export/import.

Finally the model is tested, from this process new specifications are obtained, which is new data to be used to improve the model. It should be noted that our design works in real-time so the feedback will be a key point in checking and improving outcomes. Treating design at a system level, this thesis will focus its attention in a testing model based perspective, where limitations of the system are studied and project scope will be determined. For that purpose some helpful tools and optimizers are used. In order to succeed in this process a thorough analysis of the design along with its optimization is needed. An adequate implementation of this process in engineering designs during early-phases will lead the designers to achieve a successful and profitable product.

The thesis contains some theory and basic concepts about the wide kind of formats we can work with in SOLIDWORKS and UNITY. The results of the study and the discussion of these are thereafter presented. Furthermore, it includes some theory of Virtual Reality environment and Oculus Rift Head mounted display, which helps to understand better this device.

ALTERNATIVES

In our study, we evaluate two alternatives to success in the data translation between SOLIDWORKS and UNITY. From here, we will focus on the part concerning the interchange format SOLIDWORKS choice, which will be the key part around will guide our work, and subsequent implement of Oculus Rift device through Oculus Rift Plug-in.

Research shows us the possibility to use different data interchange conversion program that allow our design the import SOLIDWORK data, but basically divided into two big groups. Direct Converters and intermediate bridge converters:

1- Direct Converters, which directly convert the data-file information from SOLIDWORKS to and intermediate format, FBX-format. This FBX-format is directly supported for UNITY. The direct converter used in this designed will be, Interchange Format Simlab SOLIDWORKS Plugin.

2- Intermediate Converters. They are used as an intermediate bridge converter, which converts and compatible SOLIDWORKS data-files into other engines programs such as Autodesk Maya or 3Dmax. These communicate seamlessly with UNITY, which offers support for 3D package to these programs. (Check it out in the chapter 2, Table 2.2.1). So as we can notice two exports must be done. Coming up next, find the following examples: Anark Core [4] and Autodesk DirectConnect [5]

Anark Core produce Quick and easy aggregation and combination of 3D models. It is a robust software platform that allows manufacturing companies re-purpose fast, easy and cost effective way of 3D product data. Otherwise, allows the data-translation with SOLIDWORKS. This will be extremely useful if we need to modification SOLIDWORKS files before export into unity.

Autodesk DirectConnect is a family of data translators. Each of these translators import a CAD file format specific to one or more of the following software products Autodesk Maya or Autodesk-3DMax.

SELECTION CRITERIA

After research and possible solutions already on the market, it is concluded that the design of a system of direct conversion greatly reduce the cost of it, because the cost of software license are much lower. Otherwise, in our CAD models we have complete model-files in SOLIDWORKS that do not require further modification. Besides obviously

understand that as more conversions are made and more programs are used for the same operation, more probable the same file will lose information.

1.4. Research Methods

For this thesis the Simlab Tool, an interchange format-data plugin for SOLIDWORKS, has been implemented. SOLIDWORKS is very powerful software that combined with the possibility of add plugins, what gives the user multitude of possibilities in the field of format conversions.

The primary objective of this Simlab tool is to analyze the standard verification of compatibility to understand how accurate will be the interchange data as well as discover the method drawbacks such as loss of information, so that they can be modified and optimized to reach an accurate and standard design. This Interchange format facilitates the creation, development, implementation and management of data conversion, and based on relevant standards process. SimLab SolidWorks plugin enables SOLIDWORKS users to export their CAD 3D models to FBX-format file, which facilitates higher-fidelity data exchange between SOLIDWORKS and UNITY platform. The Simlab tool works within a SOLIDWORKS toolbar. The conversion operation is realizing in quick and easy way by using the main window of the program. With these results the user will be able to understand better the design and the relationships between the different formats.

Afterwards, to reach the integration formats on the system, the exportation of the format is performed within Unity. This goal is achieved with an importation that the platform UNITY easily allows. Basically what this technique does is to read and optimize a set of files getting them ready for a posterior manipulation. That is possible because UNITY allows FBX-format data translation.

After that, the user, interpreting these results, can obtain a great overview of the design, identifying the critical parts of the system, which need more attention. Obviously, the real-time working allow us to modify and sub-edit the parts of the files is not working properly, as well as find some optimizers and already created applications into Unity Asset Store, which could help us to optimize the final environment. Hence, take the correct decisions to improve the design and achieve a commercially profitable product.

Finally, the compatibility between Unity and Oculus Rift Software SK2, which is an open source code for developers, is established through Oculus plugin support. The Plug-in folder contains the library OculusPlugin.dll, which enables the Rift to communicate with UNITY on Windows (both 32 and 64-bit versions). Therefore we will

have achieved full communication between SOLIDWORKS CAD-models and Oculus Rift Head Mounted Display.

1.5. Thesis Structure

The list of abbreviations is in front of this thesis. Appendix is the List of figures, and it is at the end of the thesis.

This thesis is broken up into two main parts: investigation- theory part, in order to find the best design alternatives and experiment part, where every export possibilities are checked. The theory part according to how to integrate the Virtual Reality environment and how to import SOLIDWORKS files within UNITY 3D will be in the Chapter 1 and 2. Here is added Oculus Rift device, letting us understand how to visualize the files already exported in real-time.

Chapter 3 is the main part of the project, carrying on the experimental evaluation and analysis of export, where we could observe the strengths and the limitations in the conversion data-files from SOLIDWORKS to FBX-Format.

Chapter 4 explains the differences between Nurbs and polygons mesh and the complex way to convert into FBX files. Obviously we will study the possible solutions.

Chapter 5 indicates us how to use Unity assets store, where we can find numerous of useful applications to improve our designs. Furthermore we find out the optimizers tools that will be necessary to fix the problems that appear in the conversion file operation.

Chapter 6 study quality and render settings that are fundamental for a properly build and run of the final game.

Finally, In Chapter 7, we draw conclusions about our project.

2. ELEMENTS OF GRAFIC PIPELINE

In 3D computer graphics, the graphics pipeline refers to the sequence of steps used to create a 2D raster representation of a 3D scene. Once a 3D model has been created, for instance in a video game or any other 3D computer animation, the graphics pipeline is the process of turning that 3D model into what the computer displays. In this chapter we study the more important steps and user programs to achieve a successful design.

2.1. SOLIDWORKS Formats

SOLIDWORKS is solid-modeling CAD (computer-aided-design) software that runs on Microsoft Windows and has been produced by Dassault Systems Corp. Dassault Systems SOLIDWORKS Corp. offers complete 3D software tools to create, simulate, publish, and manage data. SOLIDWORKS products are easy to learn and use, which help users design products better, faster, and more cost-effectively. SOLIDWORKS focus on ease-of-use allows more engineers, designers and other technology professionals than ever before to take advantage of 3D in bringing their designs to life.

As a powerful design program and to reach compatibilities between other programs, SOLIDWORKS has many possibilities of import/export data, supporting many formats for saving models. In this chapter will be studied all the different kind of formats that SOLIDWORKS is able to export with. This program by default can save the data in three main native extensions: SOLIDWORKS part (.sldprt); SOLIDWORKS assembly (.sldasm); SOLIDWORKS drawing (.slddrw)

SOLIDWORKS files use the Microsoft Structured Storage file format, which is a technology developed by Microsoft as part of its Windows operating system for storing hierarchical data within a single file. This means that there are various files embedded within each .slddrw (drawing files), .sldprt (part files), .sldasm (assembly files). [6]

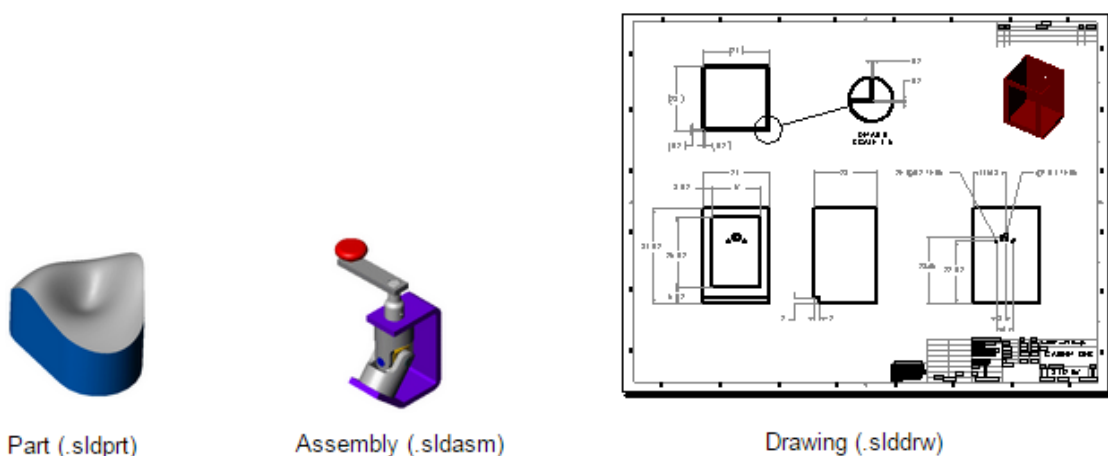


Figure 2.1 SOLIDWORKS native extensions

Those are the extension files that SOLIDWORKS works with, and are only recognized by this software. However, any of these files can be saved out as any standard formats and many others, in order to import/export files between all this design environment programs.

OVERVIEW OF IMPORT/ EXPORT CAD MODEL-DATA

If the data importation/exportation is not made in the native format (sldprt, sldasm, slddrw), SOLIDWORKS provides more than 30 translators to convert incoming CAD data into SOLIDWORKS 3D CAD format or to export SOLIDWORKS data to other CAD products.

SOLIDWORKS software can also import files from other applications. It can also export SolidWorks documents to a number of formats for use with other applications and programs. The following table displays the data translation methods available:

Application	PARTS		ASSEMBLIES		DRAWINGS	
	Import	Export	Import	Export	Import	Export
3D XML		X		X		
ACIS	X	X	X	X		
Adobe Illustrator	X	X		X	X	
Adobe Photoshop	X	X	X	X	X	X
Adobe Portable Document Format		X		X		X
Autodesk Inventor	X		X			
CADKEY	X		X			
CATIA Graphics	X	X	X	X		
CATIA V5	X		X			
DXF/DWG files	X				X	X
DXF 3D	X	X				
eDrawings		X		X		X
Highly Compressed Graphics		X		X		

HOOPS		X		X		
IDF 2.0, 3.0 (CircuitWorks Lite)	X					
IDF 2.0, 3.0, 4.0 (CircuitWorks)		X	X	X		
IFC	X	X	X	X		
IGES	X	X	X	X		
JPEG		X		X		X
Mechanical Desktop	X		X			
PADS (*.asc) (CircuitWorks)		X	X	X		
Parasolid	X	X	X	X		
PDF		X		X		X
Pro/ENGINEER	X	X	X	X		
ProStep EDMD (*.idx) (CircuitWorks)		X	X	X		
Rhino	X					
ScanTo3D	X	X				
Solid Edge	X		X			
STEP	X	X	X	X		
STL	X	X	X	X		
TIFF	X	X	X	X		X
Unigraphics	X		X			
VDAFS	X	X				
Viewpoint		X		X		
VRML	X	X	X	X		
XPS		X		X		X
Application	Import	Export	Import	Export	Import	Export

Table 2.1 *Import & Export SOLIDWORKS Formats [7]*

EXPORT CONCLUSION FROM SOLIDWORKS

In our project we are focus on the Exportations from SOLIDWORKS to UNITY so it is essential to review which the formats that can be exported are. The most common files that you can EXPORT without problems are the followings:

ACIS (.sat), Adobe Portable Document Format (.pdf), DXF/DWG (.dxf, .dwg) files, eDrawing (.eprt, .easm, or .edrw), IGES (.igs), Parasolid (.x_t, .x_b), STEP (.step), STL (.stl), Tif (.tif), VDAFS (.vda), VRML (.wrl). Later, notice some of the main features of each format. [8]

ACIS

The ACIS translator supports import/export of body and face colors, curves, and wireframe geometry. You can export the entity attribute information of faces and edges to ACIS files, and this information is retained in the ACIS file. Wireframing is one of the methods used in geometric modelling systems. A wireframe model represents the shape of a solid object with its characteristic lines and points

PDF

Portable Document Format, is a format for storing digital documents independent of software or hardware platforms. You can export SolidWorks part, assembly, and drawing documents as Adobe Portable Document Format (PDF) files and part and assembly documents as 3D PDF.

DXF/DWG

DXF, Drawing Interchange Format is a CAD data file format developed by Autodesk for enabling data interoperability between AutoCAD native file format, DWG (Drawing) and other programs. You can import/export DXF/DWG files. You can also insert DXF/DWG files into part documents.

eDrawings

In SolidWorks eDrawings you can view and animate models and drawings and create documents convenient for sending to others. SolidWorks eDrawings is installed automatically with SolidWorks Professional and SolidWorks Premium.

IGES

The Initial Graphics Exchange Specification (IGES) is a file format which defines a vendor neutral data format that allows the digital exchange of information among CAD systems. There are many IGES types that support import/export.

Parasolid

Is a 3D solid modeling software component used in CAD packages. When exported from the parent software package, a Parasolid commonly has the file extension .x_t. Another format is .x_b, which is in binary format. Most Parasolid files can communicate and migrate only 3D solids and/or surface data. Parasolid files currently cannot communicate and migrate 2D data such as lines and arcs. Component names in assemblies are retained for both import and export. The Parasolid translator does not

support the import or export of point data. The Parasolid translator supports import and export of curves and wireframes

STEP

Standard for the Exchange of Product model data. It is an international standard for representation and exchange of industrial product information. Typically, STEP can be used to exchange data between CAD and CAM.

STL

Stereo Lithography is a computer file format computer design (CAD), which defines geometry of 3D objects, excluding information such as color, texture or physical properties that do include other CAD formats.

Tif

TIFF (Tagged Image File Format) is a computer file format for images

VDAFS

The VDA surface interface (VDAFS) is a neutral CAD interface which permits the interchange of 3-D geometries between different CAD systems. It was conceived for the transfer of freeform surfaces and their supporting objects.

VRML

Virtual Reality Modeling Language, is a standard file format for representing 3-dimensional interactive vector graphics, design particularly with the World Wide Web in mind.

Before export our model we should consider some points related with the way each user is working with SOLIDWORKS. Depending on the scope of the work you plan to do at any given time, you can specify an appropriate suppression state for the components. This can reduce the amount of data that is loaded and evaluated as your work. The assembly displays and rebuilds faster, and you make more efficient use of your system resources. There are four suppression states for assembly components, and depending on them we should pay more attention to the export: [9]

-Resolved, is the normal state for assembly components. A resolved component is fully loaded in memory, fully functional, and fully accessible.

-Suppressed, can be used to remove a component temporarily from the assembly, without deleting it. It is not loaded in memory, and it is no longer a functional member of the assembly. You cannot see a suppressed component, or select any of its entities. A suppressed component is removed from memory, so loading speed, rebuild speed, and display performance is improved. Because of the reduced complexity, evaluation of remaining components occurs faster.

-Lightweight, You can load an assembly with its active components fully resolved or lightweight. Both parts and sub-assemblies can be lightweight. When a component is fully resolved, all its model data is loaded in memory. When a component is

lightweight, only a subset of its model data is loaded in memory. The remaining model data is loaded on an as-needed basis. You can improve performance of large assemblies significantly by using lightweight components. Loading an assembly with lightweight components is faster than loading the same assembly with fully resolved components. Assemblies with lightweight components rebuild faster because less data is evaluated.

-Hidden parts, Think of hidden components in a SolidWorks assembly as just that not visible parts. From there, they are essentially exactly like a visible part. Hidden parts also take the stress off the video card since it does not have to display them. However, hidden parts are still loaded into memory, therefore their math data takes up part of the memory footprint.

ACCORDING TO THAT, WE MUST CONSIDER A FEW THINGS IN THE EXPORTATION:

-If you export an assembly that contains lightweight parts, the Resolve Lightweight Parts dialog box appears. You must resolve any lightweight parts before you can export an assembly.

-If you export an assembly that contains hidden or suppressed parts, a message box appears asking if you want to resolve these parts. To export these parts, you must resolve them.

-If you select any faces, surfaces, solid bodies or surface bodies of a part, the Export dialog box appears verifying if you want to export only the selected objects. Click selected face(s) to export only the selected faces, selected bodies to export only the selected bodies, or all bodies to export all solid bodies.

2.2. UNITY 3D Formats

UNITY is a 3D application in real-time and multimedia, as well as being 3D and physics engine used to create gaming, animation in real-time, interactive content consisting of audio, video and 3D objects.

UNITY 3D is a game engine that uses the graphics libraries OpenGL and DirectX, which can generate three-dimensional interactive applications. Within the capabilities of the engines is the management of ASSETS, which may be, geometries, textures, animations, and so on. Standard Assets are the set of tools and prefabs for you to use as a building block in creating your game or even just to get your kickstarter in prototyping. These assets allow the inclusion of three-dimensional graphic content, which is the study of matter about the project. By then achieve inclusion of these assets graph type must use the FBX format, this format originally developed by the

company Kaydara, is now a standard in the digital entertainment industry and is used as file transfer content between more usual dimensional modeling programs.

WHY USE UNITY 3D

The main reason we choose to implement UNITY game engine is due to compatibility relations between Unity 3d corporation and Oculus Rift.

Obviously, very positive agreements created between Unity 3D technologies and Oculus Rift to push the game development using virtual reality tools. The SDK Oculus Rift software includes default support for the UNITY engine, so you would think that if it's extensive list of existing games with support Oculus Rift, the catalog will expand greatly, mainly because of the way in which it is provided integration with engines of development, have a multiplatform SDK, and the low price of this technology that makes purchasing opportunities increase.

Other advantage is the acceleration of creation of virtual environments. Such as its ease of use, not very expensive license, and efficient capabilities and working tools. You can incorporate graphics and audiovisual content efficiently saving time programming, besides that this is easy to incorporate high-level languages C #, Java Script and Boo.

Using UNITY opens the questions about the formats of patterns, textures and other graphic and audiovisual content that supports the engine, creating the following questions. Which formats are compatible? How to import them? In response, raises a number of routine checks on the most popular formats that allow the incorporation of content from an application or external applications, as SOLIDWORKS.

REVISION OF THE POSSIBILITIES OF IMPORT GRAPHIC CONTENT IN UNITY 3D.

The use of more than one application to develop real-time graphic content leads to the need to use any file format compatible with several programs of modeling, texturing and animation. Our aim is standardize a format.

The process of export of three-dimensional models from SOLIDWORKS to UNITY is performed through an interchange format, FBX format. FBX-format is a 3D asset exchange format compatible with many 3D tools. This format allows the transfer of geometries, textures, animations, simple materials, and even has the option to export cameras and lights, although, is not recognized by Unity game engine. All of these possibilities will be implemented in the next chapter of the Thesis [3. Experimental Evaluation].UNITY allows importing these files from FBX-format, so this will be our starting point. We must find a way to convert files of other types of programs, such as SOLIDWORKS format into Unity FBX-format.

Based on the reference tables of UNITY represented below, table 2.2.1, the import of graphic content into UNITY 3D is verified through the FBX interchange format. This table 2.2.1 shows the programs that are compatible with FBX-format, and the features of them.

First the components of the package support are introduced: [10]

MESHES

Meshes make up a large part of your 3D worlds. A mesh consists of triangles arranged in 3D space to create the impression of a solid object. A triangle is defined by its three corner points or vertices. In the Mesh class, the vertices are all stored in a single array and each triangle is specified using three integers that correspond to indices of the vertex array. The triangles are also collected together into a single array of integers; the integers are taken in groups of three from the start of this array, so elements 0, 1 and 2 define the first triangle, 3, 4 and 5 define the second, and so on.

TEXTURES

A model will make use of texturing to create fine detail on its surface. A texture is a bit like an image printed on a stretchable sheet of rubber. For each mesh triangle, a triangular area of the texture image is defined and that texture triangle is stretched and “pinned” to fit the mesh triangle. To make this work, each vertex needs to store the coordinates of the image position that will be pinned to it. These coordinates are two dimensional, commonly called UV coordinates.

ANIMATION

Unity’s animation system is based on the concept of Animation Clips, which contain information about how certain objects should change their position, rotation, or other properties over time. Each clip can be thought of as a single linear recording.

BONES

In most of the methods of computer animation, an animator creates a simplified representation of the anatomy of a character-humanoid; it has less difficulty to be animated. In biped or quadruped characters, many parts of the skeleton of the character corresponding to the actual bones. The animation with bones also used to encourage many other things, such as facial expressions, a car or other object that you want to provide movement.

3D Package Support

	Meshes	Textures	Anims	Bones
Maya .mb & .ma ¹	✓	✓	✓	✓
3D Studio Max .max ¹	✓	✓	✓	✓
Cheetah 3D .jas ¹	✓	✓	✓	✓
Cinema 4D .c4d ^{1 3}	✓	✓	✓	✓
Blender .blend ¹	✓	✓	✓	✓
modo .lxo ²	✓	✓	✓	
Autodesk FBX	✓	✓	✓	✓
COLLADA	✓	✓	✓	✓
Carrara ¹	✓	✓	✓	✓
Lightwave ¹	✓	✓	✓	✓
XSI 5.x ¹	✓	✓	✓	✓
SketchUp Pro ¹	✓	✓		
Wings 3D ¹	✓	✓		
3D Studio .3ds	✓			
Wavefront .obj	✓			
Drawing Interchange Files .dxf	✓			

¹ Import uses the application's FBX exporter. Unity then reads the FBX file.

² Import uses the application's COLLADA exporter. Unity then reads the COLLADA file.

³ Cinema4D 10 has a buggy FBX exporter.

Table 2.2.1 *Unity 3D package support*

We can see each of the programs that are compatible and allow importing files into UNITY platform within FBX-format. As we can notice **SOLIDWORKS has not native support with UNITY**. That is the reason why an interchange data plug-in is needed, which allows the entire communication between these two programs.

2.2.1. Importing 3D models to Unity

To import a 3D model into Unity you can drag a file into the project window. In the inspector > Model tab Unity supports importing models from most popular 3D applications.

Assets> Import New Assets

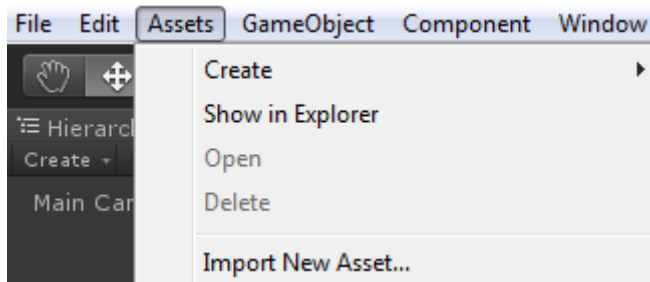


Figure 2.2.1 *Import Inspector*

3D FORMATS

Importing meshes into Unity can be achieved from two main types of files: [11]

1. Exported 3D file formats, such as .FBX or .OBJ
2. Proprietary 3D application files, such as .Max and .Blend file formats from 3D Studio Max or Blender for example. Either should enable you to get your meshes into Unity, but there are considerations as to which type you choose:

EXPORTED 3D FILES

Unity can read .FBX, .dae (Collada), .3DS, .dxf and .obj files, FBX exporters can be found in the chapter [2.3.interchange software].

Advantages:

- Only export the data you need
- Verifiable data
- Generally smaller files
- Encourages modular approach e.g. different components for collision types or interactivity
- Supports other 3D packages whose Proprietary formats we don't have direct support for

Disadvantages:

- Can be a slower pipeline for prototyping and iterations
- Easier to lose track of versions between source(working file) and game data (exported FBX for example)

PROPIETARY 3D APPLICATION FILES

Unity can also import, through conversion: Max, Maya, Blender, Cinema4D, Modo, Lightwave & Cheetah3D files, e.g. .MAX, .MB, .MA etc.

Advantages:

- Quick iteration process (save the source file and Unity reimports)
- Simple initially

Disadvantages:

- A licensed copy of that software must be installed on all machines using the Unity project
- Files can become bloated with unnecessary data
- Big files can slow Unity updates
- Less validation, so it is harder to troubleshoot problems)

2.2.2. FBX & OBJ which format is better?



FBX or OBJ, which format is better? If you compare the two options FBX is considered to be a higher quality modern 3D exchange format that retains more information while OBJ is a simple model format that is popular for 3D data exchange.

OBJ [12] has its origins with Alias Wavefront, a simple data-format that represents 3D geometry namely, the position of each vertex, the UV position of each texture coordinate vertex, vertex normals, and the faces that make each polygon defined as a list of vertices, and texture vertices. Vertices are stored in a counter-clockwise order by default, making explicit declaration of face normals unnecessary.

The OBJ file format is used in the Digital-Tutors asset library for assets that consist of only the 3D model. They are provided as an alternative to software proprietary formats to allow you the most flexibility for importing the 3D mesh data into whatever 3D software you are using.

Limitations of this file type: The OBJ file format is one of, if not the, most common transfer format for 3D models as it can be read by nearly every industry-standard 3D application. OBJ files cannot hold much more than the polygon data, UV maps and some standard materials. OBJ files cannot contain any sort of rigged or animated assets.

FBX [13] is a proprietary file format originally developed by Kaydara, a 3D asset exchange format compatible with many 3D tools. FBX facilitates higher-fidelity data exchange between Autodesk content creation packages and supports certain third-party and propriety applications. With FBX it's easier to transfer files, retain more data and work more efficiently.

The FBX file format is used in the Digital-Tutors asset library for any and all rigged or animated assets. They are provided as an alternative to software proprietary formats

to allow you the most flexibility for importing the rigs and animations into whatever 3D software you are using.

Limitations of this file type: In addition to the models themselves, the FBX file format can support mapping, standard materials, rigging and animation. For that reason, this format is one of the most common formats for transferring rigs and animations between different software programs. There are a few different versions of FBX so depending on what version your FBX importer is for your software might determine how well some of the rigs and animations get imported. FBX files cannot carry proprietary materials or shaders.

In the table below we can see a comparative between FBX and OBJ files.

Description	FBX	OBJ
Animation	Yes	No
Mesh	Yes	Yes
Skeleton	Yes	No
Morphs	Yes	No
Vertex Animation	Yes	No
Texture	Link	Link
Material	Yes	Link
LightMap DetailMap NormalMap	Link	No
Animation Takes	Yes	No
Binary Format	Yes	No
Instancing	Yes	No
Scene Hierarchy	Yes	No
Multiple Channels Mapping	Yes	No
Interchange File Format	Yes	Yes

Table 2.2.2 Comparison of FBX and OBJ formats properties

2.3. Oculus Rift Introduction

2.3.1. Oculus Rift User Interface

Oculus Rift is a virtual reality device with a wide field of vision made as a Head-mounted display (HMD) and low latency. It is being developed by Oculus VR Company.



Figure 2.3.1 *Oculus Rift HMD*

OVERVIEW OF OCULUS RIFT DEVELOPMENT KIT (DK2)

DK2 is the latest development kit for the Oculus Rift that allows developers to build amazing games and experiences for the consumer Oculus Rift.

OVERVIEW OF OCULUS RIFT SOFTWARE (SDK)

The games and platforms must be specifically designed to work properly with Oculus Rift. To do this, Oculus has created a software development kit (SDK) to help developers integrate Oculus Rift in their games.

The Oculus SDK software is designed to be as easy to integrate as possible. The SDK includes C++ source code, libraries, headers, firmware, samples, tutorials and documentation. The Oculus SDK is publicly available and open source, meaning that anyone can obtain, use and even modify and distribute the code. Developers can access the Oculus VR Developer Center to retrieve the latest versions of the SDK components and for online support. These resources should help greatly in developing or porting more games and other content to the Oculus Rift. [14]

The Oculus SDK is using a dynamic link library (DLL) model. Using a DLL offers several advantages:

- As long as the arguments and return values are the same, experiences do not need to be recompiled to take advantage of the updated library.
- Localization into new languages is easier because the functions remain consistent across languages.

- The DLL can be updated to take advantage of new features and headsets without affecting current games and experiences.

RIFT DISPLAY MODE

The Rift Display Mode dialog box enables you to change the display mode for development and to run different types of applications. The following figure shows the dialog box:

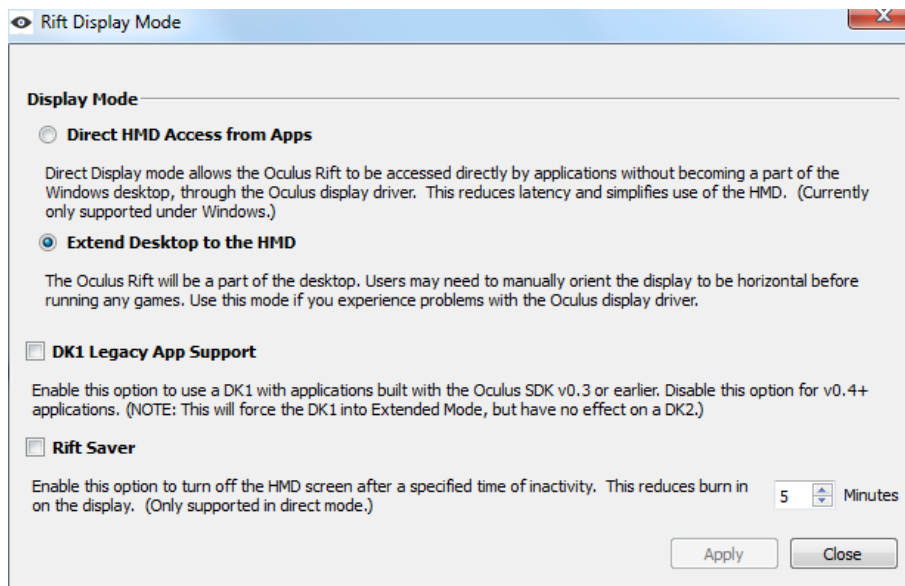


Figure 2.3.1.1 *Rift display mode dialog Box*

Select from the following:

- **Direct HMD Access from Apps:** this is the recommended mode. Rifts connected to the PC will not appear as additional monitors and will be automatically used by Rift-enabled applications.
- **Extend Desktop to the HMD:** this is provided for debugging purposes in the event that there are problems with the Direct HMD Access mode.

MONITOR DISPLAY MODE

Display devices identify themselves and their capabilities using Extended Display Identification Data. The display within the Oculus Rift interacts with the system in the same way as a typical PC monitor. Notice the following configuration options: [15]

DUPLICATE DISPLAY MODE

In duplicate display mode, the same portion of the desktop is shown on both displays; each device uses the same resolution and orientation settings. Your computer attempts to choose a resolution that is supported by both displays, while favoring the native resolutions described in the Extended Display Identification Data information reported by the displays. Duplicate mode can be useful for configuring the Rift. However, it suffers from vsync issues. The vertical sync is a feature found on many

games that when activated prevents the effect known as "tearing" that it does is that when you move the camera frame is choppy in lines, which can be annoying.

EXTENDED DISPLAY MODE

In extended mode, the displays show different portions of the desktop. You can use the Control Panel to set the desired resolution and orientation independently for each display. Extended mode suffers from shortcomings due to the fact that the Rift is not a viable way to interact with the desktop. Nevertheless, it is the current recommended configuration option.

After connecting a Rift to the PC, you can modify the display settings through the Windows Control Panel. In Windows 7, select Control Panel -> All Control Panel Items -> Display -> Screen Resolution

The following figure shows the DK2 correctly configured in extended display mode in Windows.

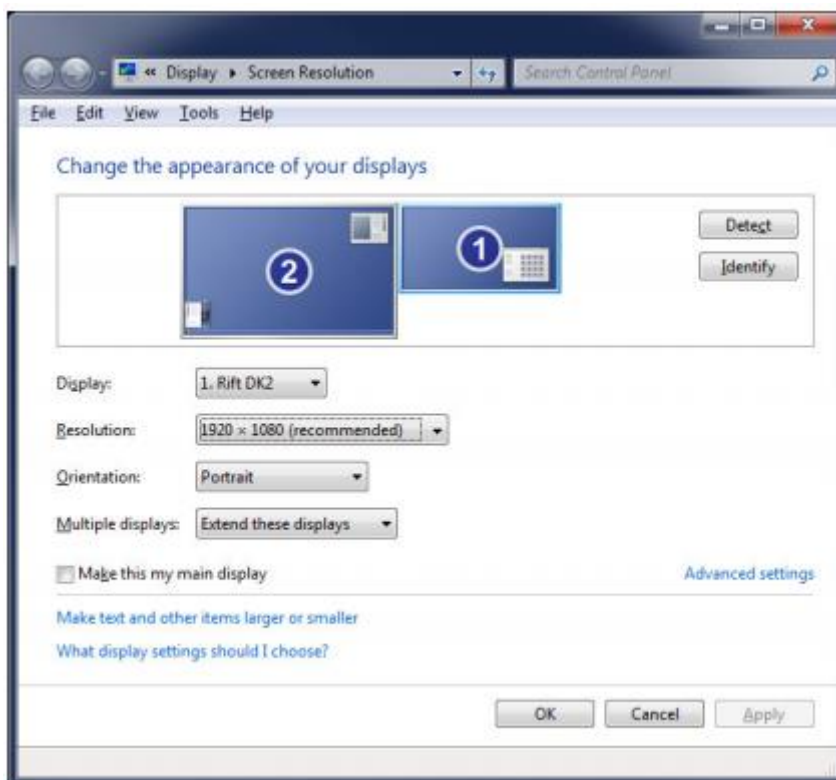


Figure 2.3.1.2 *Extended display mode*

HEAD TRACKING AND SENSORS

The Oculus Rift hardware contains a number of micro-electrical-mechanical (MEMS) sensors including a gyroscope, accelerometer, and magnetometer. Starting with DK2, there is also an external camera to track headset position. The information from each of these sensors is combined through a process known as sensor fusion to determine

the motion of the user's head in the real world, and to synchronize the user's virtual view in real-time.

The following figure shows the DK2 position tracking camera mounted on a PC monitor and a representation of the resulting tracking frustum.

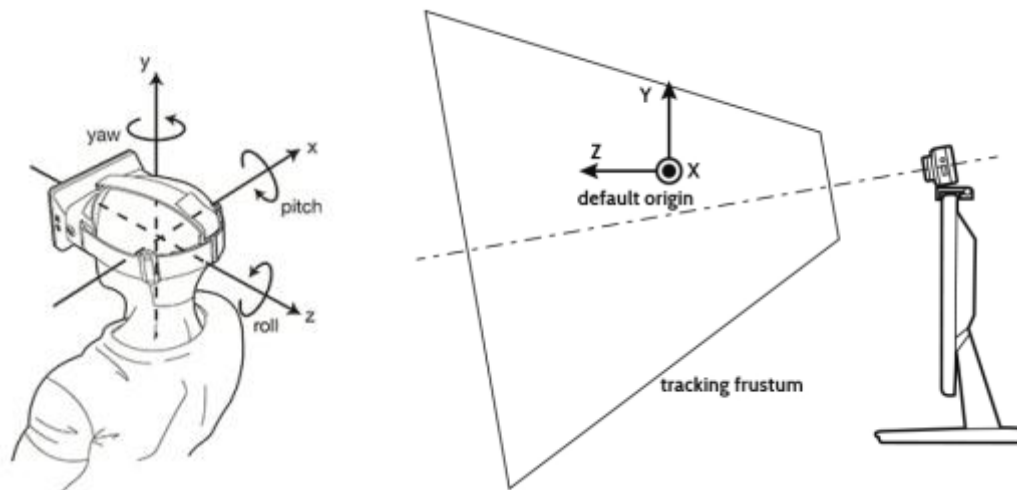


Figure 2.3.1.3 Head tracking-sensors [16]

2.3.2. Oculus Rift Plug-in

In this chapter is studied the plug-in which allows compatibility between UNITY and Oculus Rift. This plug-in version is available for free download on the Oculus developer site. Include that the version of Oculus Rift consumer is currently in development, so notice that this plug-in will not be the final product.

First of all, GameObjects and Prefabs are introduced, which are necessary to understand the correct functionality of this UNITY plug-in:

GAMEOBJECTS are the fundamental objects in UNITY that represent characters, props and scenery. They do not accomplish much in themselves but they act as containers for Components, which implement the real functionality. It is convenient to build a GameObject in the scene by adding components and setting their properties to the appropriate values. This can create problems, however, when you have an object like an NPC (any character that the player doesn't control), prop or piece of scenery that is reused in the scene several times. Simply copying the object will certainly produce duplicates but they will all be independently editable. Generally, you want all instances of a particular object to have the same properties, so when you edit one object in the scene, you would prefer not to have to make the same edit repeatedly to all the

copies. Fortunately, UNITY has a prefab asset type that allows you to store a GameObject object complete with components and properties.

PREFABS acts as a template from which you can create new object instances in the scene. Any edits made to a prefab asset are immediately reflected in all instances produced from it but you can also override components and settings for each instance individually.

PLUG-IN

The Plug-in folder contains the library OculusPlugin.dll, which enables the Rift to communicate with Unity on Windows (both 32 and 64-bit versions).The current integration for adding VR support into Unity applications is based on two prefabs that may be added into a scene: OVRCameraRig and OVRPlayerController [17].

- OVRCameraRig, replaces the regular Unity Camera within a scene. You can drag an OVRCameraRig into your scene and you will be able to start viewing the scene with the Rift. OVRCameraRig is a component that controls stereo rendering and head tracking. It maintains three child "anchor" Transforms at the poses of the left and right eyes, as well as a virtual "center" eye that is half-way between them. This component is the main interface between Unity and the cameras. This is attached to a prefab that makes it easy to add VR support to a scene.

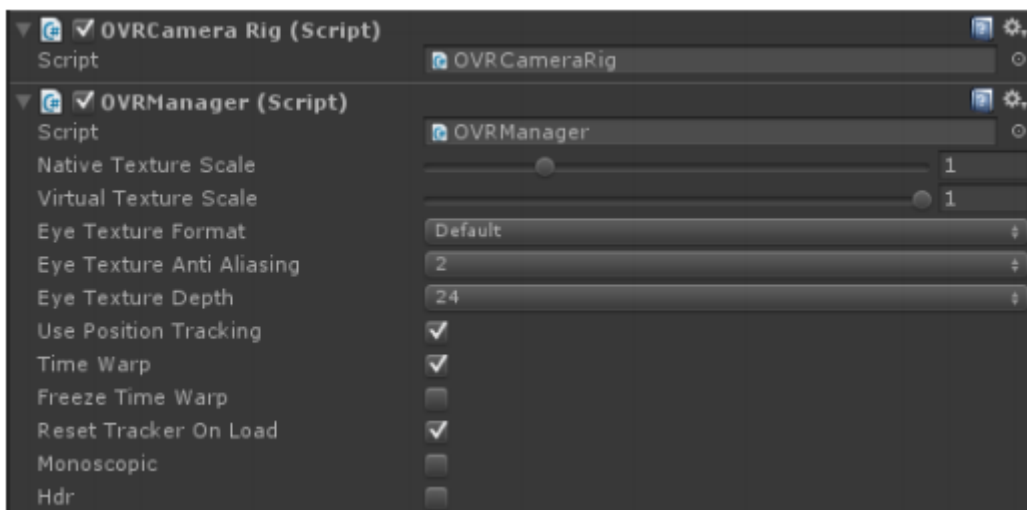


Figure 2.3.2.1 Prefabs OVRCameraRig, expanded in the editor

- OVRPlayerController, is the easiest way to start navigating a virtual environment. It is basically an OVRCameraRig prefab attached to a simple character controller. It includes a physics capsule, and a movement system. To use, drag the player controller into an environment and begin moving around using a gamepad, or a keyboard and mouse.

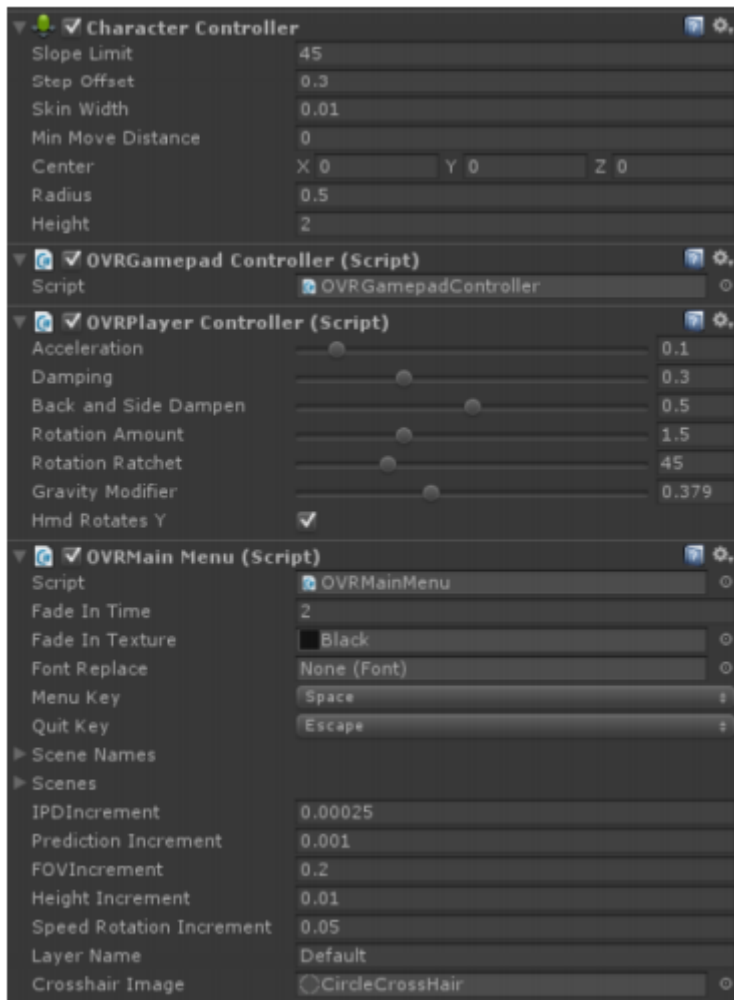


Figure 2.3.2.2 Prefabs OVRPlayerController, extended in the editor

OVRManager is the main interface to the VR hardware. It is a singleton that exposes the Oculus SDK to Unity, and includes helper functions that use the stored Oculus variables to help configure camera behavior.

This component is added to the OVRCameraRig prefab. It can be part of any application object. However, it should only be declared once, because there are public members that allow for changing certain values in the Unity inspector.

2.4. Interchange Format

Interchange format is a text file format used to import/export data between software programs produced by different companies. It facilitates management of data conversion, based on relevant standards process.

When choosing interchange format, FBX-format is considered as final format, because it is a condition set by UNITY. To enable data interoperability between SOLIDWORKS and UNITY, mainly the two following interchange formats are considered: Simlab 3D Plug-in and FBX converter.

SIMLAB 3D PLUG-IN

SIMLAB plugin is a specific Interchange format to convert every SOLIDWORKS extension files within FBX-format. This is the right option for user looking for special feature to standardize a 3d SOLIDWORKS design application. SimLab FBX exporter for SOLIDWORKS plugin enables SOLIDWORKS users to export their 3D models in FBX-file-format. This operation is realizing in and quick and easy way.

Supported versions: The plugin is supported on SolidWorks 2013, 2014, and 2015; 32, and 64bit.

HOW TO USE SIMLAB SOFT PLUG-IN [18]

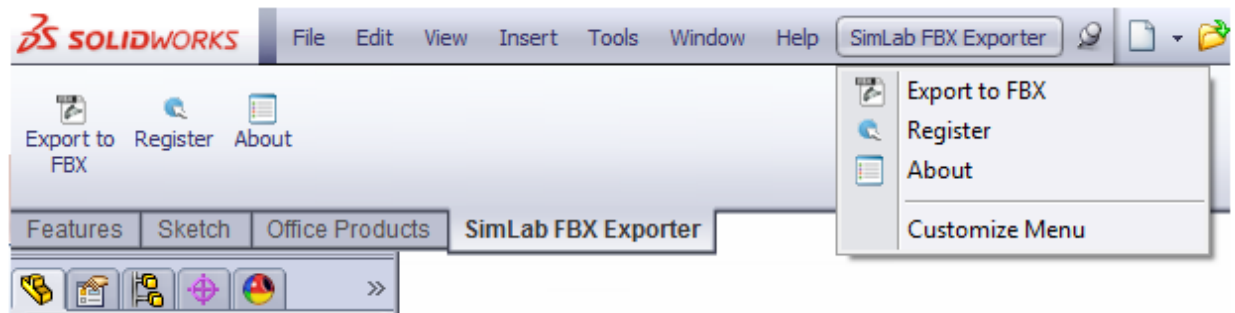


Figure 2.4 Interchange-format used in SOLIDWORKS

After installing the plugin and to access its functionalities a user needs to go Tools -> Add-Ins. In the Add-Ins dialog check SimLabSolidworksFBXExporter checkbox, and a new SimLab FBX Exporter tab and menu will be added to SolidWorks. To enable the plugin's functionalities each time SOLIDWORKS starts the Start Up checkbox should be checked as well.

For the first time a user will need to register the plugin's license, Trial or Professional. This can be done by clicking the Export to FBX or Register options in the plugin's tab or menu. Click Export to FBX option and start exporting your SolidWorks Parts/Assemblies into FBX.

In our project we choose Simlab 3D plug-in because is working in detail with SOLIDWORKS which will facilitate our design, and will help us to find the best solutions of the models.

FBX CONVERTER

FBX converter is focus on convert files from the Autodesk Family. If we value as an alternative trying to pass SOLIDWORKS to Maya or SOLIDWORWS to 3D Studio Max out efforts to jump into Unity. This could be done with FBX Converter.

FBX Converter Transfer files from one file format to another quickly and easily. This utility enables you to convert OBJ, DXF, DAE, and 3DS files to or from multiple versions of the FBX format. New tools are now available with the last FBX Converter version. You can view FBX animation files in real time with the FBX Viewer, explore and compare FBX file contents with the FBX Explorer, and manage animation takes with the FBX Take Manager. [19]

SINGLE-STEP INTEROPERABILITY

Maximize creative potential and optimize productivity with simple interoperability workflows between certain Autodesk 3D animation software tools:

- 3ds Max modeling, animation, rendering, and compositing software
- Maya 3D animation, modeling, simulation, rendering, and compositing software
- Maya LT 3D modeling and animation software
- MotionBuilder 3D character animation software
- Mudbox 3D digital sculpting and texture painting software

3. EXPERIMENTAL EVALUATION

3.1. Analysis of Exporting Options for FBX Files

The process of export of three-dimensional models from SOLIDWORKS to UNITY is performed through an interchange format, FBX format. This format allows the transfer of data from different levels of geometries, textures, animations, simple materials, and even has the option to export cameras and lights, although, is not recognized by Unity game engine.

Within the world of real-time three-dimensional modeling for, there are several useful applications to model, texture and animate characters-humanoids and industrial objects. Each of them has similarities and unique capabilities, that's the reason why modelers and animators use either program.

This use of more than one application to develop real-time graphic content as a final virtual reality environment, leads to the need to use any file format compatible with several design programs, in order to standardize the process. Working with Unity, is necessary the use of standard FBX-format, for the management of the data conversion. As a result, the export process of three-dimensional models from SOLIDWORKS to UNITY must be done through the interchange format, FBX-format. This format allows the transfer of data from different levels of geometries, textures, animations, simple materials, and even has the option to export cameras and lights, although, is not recognized by UNITY game engine.

Therefore is also necessary take tests with the FBX-format to understand their capabilities and limitations, so that modelers hold a useful format for the development of virtual environments. This format, can improve models with minimal loss of information. For this purpose, compatibility tests are carried out in the areas of export/import between SOLIDWORKS CAD modeling program, and Unity game engine. In this way we can make comparisons between benefits and problematic of the FBX format as three-dimensional models transfer between this applications.

3.2. Examples: crane, hydraulic cylinder, pipe system.

The export from SOLIDWORKS files to FBX-Format is implemented through 3 different CAD models created in SOLIDWORKS. A) crane, B) hydraulic cylinder, C) pipe system. Each of them has different characteristics, and all have been previously animated in SOLIDWORKS, in order to visualize their behavior in the export.

The crane is designed with a polygonal geometry, the hydraulic cylinder has revolution polygonal geometry too, and the pipe system is created by Nurbs (non-uniform

rational B-spline). Both designs are animated in SOLIDWORKS and provided with light and shadows. These 3 models are modified along the experimental evaluation in order to examine as many as possible converting options in the fields of textures, materials, scale, light and shadow along verification tests, studied in the next chapter.

Note that the main reason why we selected these different models with different construction geometries is because the biggest difficulty for an interchange format is the conversion to FBX-format from the mathematical model (NURBS). Thereby a more complete data interchange is tested. We considered most types of possible construction, in order to standardize this method of exportation with the highest level of detail possible. That said, it should be noted, that according to the types of designs that try to export could give us a little different solutions. This check is done just for these 3 models, so we cannot assure the same conclusions for all kind of models. Find the pictures of the 3 different model-files in the following figures.



Figure 3.2.1 C) Pipe system



Figure 3.2.2 A) Crane

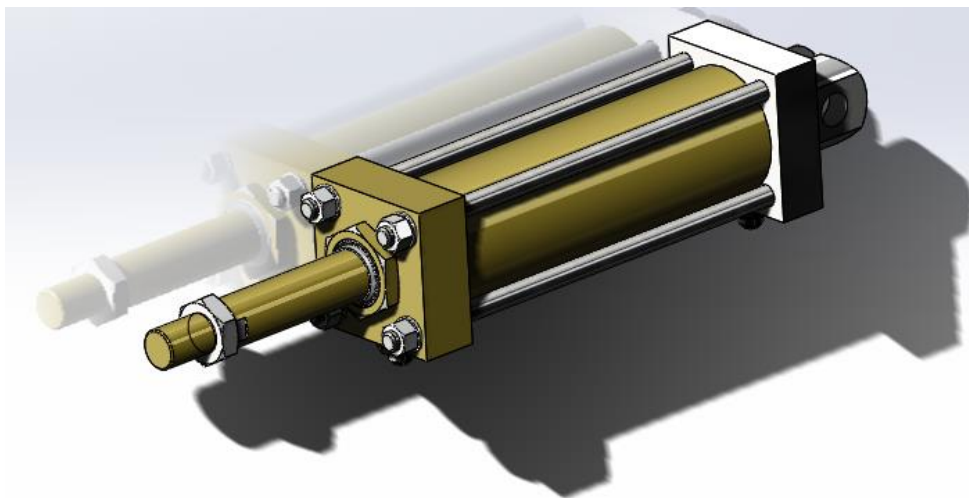


Figure 3.2.1 B) Hydraulic Cylinder

3.3. Standard Verification of Compatibility

Standard compatibility checks are a series of tests made with the 3 SOLIDWORKS model-files examples that will run the different content that are used within the three-dimensional real-time applications. This Allow us to verification of essential elements used in this models and if the export into FBX-format and posterior implementation in UNITY is working properly. To have a complete verification, geometrical level, texture level, material level, scale level, light and shadow level, camera level and animation level is checked. Also is explained the procedure applied to each verification level.

3.3.1. Geometric Level

Procedure applied to each type of geometry:

1. Create the type of geometry
 - Primitives
 - Splines
 - Polygons
 - NURBS
 - Patch
 - Geometry deformers
 - Compound geometry
2. Preparing to Export
3. Export to FBX format
4. Review of problematic export
5. Import into the Unity 3D engine
6. Review of the geometries within the engine and the internal organization of folders
7. Collection of useful data for using FBX files Unity3D
8. Analysis of data and drawing conclusions

PRIMITIVES

As there are three-dimensional primitives cylinders, tube, torus, sphere and cube, among others. The geometric primitives in 3D software may be edited for more complex geometries, adding new vertices, edges, and polygons.

FBX does not support primitives, when exported must be converted to mesh in UNITY. It never matter a primitive from an external package. It must be considered that by converting all allocation primitives, Skin and Morphs will be lost. If required a basic primitive is recommended to do it directly into UNITY.

SPLINES

In computer graphics, parametric curves whose coordinates are given by splines are popular because of the simplicity of their construction, their ease and accuracy of evaluation, and their capacity to approximate complex shapes through curve fitting and interactive curve design.

Does not support export of Splines. These are converted into unity as empty objects. As a side note, there is a commercial extension called *Rage Spline 15* [20] that allows the management of two-dimensional graphics in UNITY.

POLYGONS

Exporting polygon meshes converts quads into triangles. For each mesh triangle, a triangular area of the texture image is defined and that texture triangle is stretched and “pinned” to fit the mesh triangle. To make this work, each vertex needs to store the coordinates of the image position that will be pinned to it. If the mesh is extremely complex you can have problems steering axis turning some normal, doing that the coordinates change of place, deforming the mesh.

NURBS

Nurbs (non-uniform rational B-spline) is a mathematical model commonly used in computer graphics for generating and representing curves and surfaces.

FBX does not support NURBS curves; they must be converted to polygons (mesh) before passing it to UNITY. You should consider that while converting the NURBS curve, all allocation Morphs and Skin will be lost. Morphs and Skin are used basically for facial animation so it doesn't make sense in SOLIDWORKS. That should be considered for 3d computer graphic programs as 3dsMax or Maya. Hopefully, the interchange format selected in our project (Simlab Soft), automatically convert Nurbs to Polygons so we do not need to do this task before, simplifying the model.

PATCH

Patch, also called local surface is a mathematical model that consist of a portion of a surface that is bounded by a closed curve.

FBX does not support NURBS curves; they must be converted to polygons (mesh) before passing it to UNITY. You should consider that while converting the Patch curve, all allocation Morphs and Skin will be lost.

DEFORMING GEOMETRY

In SOLIDWORKS we find the tool 'Surface push', which deforms and modifies surfaces of target bodies by displacing (pushing) them with tool body surfaces. You select a customizable pre-built tool body, such as a polygon or sphere, or use your own tool body. The target body surfaces approximate the tool body surfaces while maintaining a one-to-one correspondence between every target surface (the number of faces, edges, and vertices remains unchanged in the final target body) before and after the deformation.

FBX does not support deforming geometry; this must be converted to mesh by UNITY. Take into account that the deformed object to convert any allocation of Skin and Morphs will be lost. On the other hand, there is an application called *Mega Fiers16* [21] that lets you use the distorting geometry into UNITY.

GEOMETRY COMPOSED

For composite geometry definition used by two or more geometric objects of similar characteristics, as primitives, poly or mesh, the result of export conversion will be made in a single geometry.

CONCLUSIONS OF EXPORTING GEOMETRY

In the particular case of the geometries found that all geometric objects that is imported into Unity is created and converted to polygons. Just for that, you cannot export geometries made by mathematical models such as primitives or splines or NURBS. You cannot export either Geometry Patch or composite geometry, or assets deformers.

All these types of geometry are converted to polygons when exported to FBX.

3.3.2. Texture Level

Procedure applied to each texture:

1. Review of standards using three-dimensional texture in real-time applications
 - Texture alpha transparency
 - Bump textures
 - Multichannel, multi-coordinate and multi-textures
2. Preparing for export
3. Export to FBX format
4. Review of problematic export
5. Imports into the Unity 3D engine
6. Review of the textures within the engine and the internal organization of folders
7. Compilation of useful information using FBX files

Any texture used in objects that will be used in real-time applications must be a power of 2 regardless of the engine used. This measurement refers to the width and height in

pixels and are usually measured values ranging from 32, 64, 128, 256, 512, 1024 and 2048. Textures higher or lower may exist, but only recommended in individual cases.

The formats supported by Unity are Photoshop™ (psd) *no layers*, no layers tiff, JPEG, PNG, GIF, BMP, TGA, IFF, PICT among others.

In SOLIDWORKS, 'appearance' defines the visual properties of a model, including color and texture. Appearances do not affect physical properties, which are defined by materials. Appearances allow you to add color and texture to your models. They can be used in a variety of ways to aid the design process, as well as communicate your design in a more realistic manner. SOLIDWORKS ships with approximately 490 appearances in a variety of categories. The categories include (but are not limited to) plastic, wood, metal, stone, fabric, etc. Even though the out-of-the-box appearance count is quite large, there are times when you will have to create custom appearances for your specific needs.

Normally, the mesh geometry of an object only gives a rough approximation of the shape while most of the fine detail is supplied by Textures. A texture is just a standard bitmap image that is applied over the mesh surface. Textures are applied to objects using Materials. Materials use specialized graphics programs called Shaders to render a texture on the mesh surface. Shaders can implement lighting and coloring effects to simulate shiny or bumpy surfaces among many other things. They can also make use of two or more textures at a time, combining them for even greater flexibility.

Mainly we study three kind of textures; transparency, bump mapping and multi-material.

ALPHA TRANSPARENCY

The standard method to create realistic objects is to use a partially transparent image mapped to a plane with texture alpha transparent. This texture will be associated to a shader, which can make mesh geometry partially or fully transparent by reading the alpha channel of the main texture. In the alpha, 0 (black) is completely transparent while 255 (white) is completely opaque. If your main texture does not have an alpha channel, the object will appear completely opaque. Alpha Transparency is achieved by integrating the alpha channel in the texture of diffusion and should be taken into account that the specularity is exported through the alpha channel of the diffuse texture, hence cannot live a texture with specularity with openly by alphas, so you must choose one.

BUMP

Bump mapping, is a technique in computer graphics for simulating bumps and wrinkles on the surface of an object. The relief type bump is achieved through a normal map and can be improved with heightmap and parallax mapping shader (kind of shaders available in UNITY library); all these maps should be defined from UNITY.

MULTI-MATERIAL

Not always want to map the entire mesh. Same happens when we need to add a second material to the mesh, it can happen that we want to paste an image that only occupies a few space. In this case we use multi-material. In the case of export multi-material, multichannel and multiple texture coordinates are lost. It is noteworthy that to handle additional channels for calculating ambient occlusion and lightmapping, these must be made within the engine UNITY.

CONCLUSION OF EXPORTING TEXTURE

UNITY retains all common types of texture mapped to standard materials, but not always assigned automatically. The unique texture is automatically assigned is it which contains the alpha channel; the other channels must be assigned manually into UNITY. That means that is more efficient import the Texture file directly to UNITY which will produce better results and do not require lot of work.

3.3.3. Material Level

Procedure applied to each type of material:

1. Review of standards of use of materials in three-dimensional real-time applications
 - Composite
 - Cubemapping
 - Procedural
 - Vertex Color
 - Multi / Sub-Object
 - Shaders
2. Preparing for export
3. Export to FBX format
4. Review of problematic export
5. Imports into the Unity 3D engine
6. Review of textures and materials within the engine
7. Preparation and correction of materials within the engine
8. Analysis of data and drawing conclusions

The material settings into Unity expose all properties from a material, allowing you to animate them. You can also use it to set custom shader properties that can't be accessed through the inspector. There is a close relationship between Materials and Shaders in UNITY. Shaders contain code that defines what kind of properties and assets to use. Materials allow you to adjust properties and assign assets.

A Material defines:

- Which textures to use for rendering.
- Which colors to use for rendering.
- Any other assets, such as a Cubemap that could be required by the shader for rendering.

COMPOSITE

Are materials made from two or more constituent materials.

No supporting the composite material, which happens is that this material and all their sub-materials are lost and replaced by a simple material.

CUBEMAPPING

A Cubemap is a collection of six square textures that represent the reflections on an environment. The six squares form the faces of an imaginary cube that surrounds an object; each face represents the view along the directions of the world axes (up, down, left, right, forward and back). Cubemaps are often used to capture reflections or “surroundings” of objects.

For cube mapping option the solutions, is to import through a separate cubemap faces. These are imported manually in UNITY and should generate a type cubemap map and then assign to each face corresponding texture.

PROCEDURAL

UNITY incorporates a new asset type known as Procedural Materials. These are essentially the same as standard Materials except that the textures they use can be generated at runtime rather than being predefined and stored.

The procedural materials cannot go directly to FBX-format and therefore must be pre-calculated as simple maps or through shaders built into the engine.

VERTEX COLOR

Vertex painting is the art of assigning a color for each vertex of the meshes to use in our shaders and they have a great number of uses such as tinting the color of a mesh. That does not have sense in SOLIDWORKS.

Does not support export of vertex colors, these cannot be converted to FBX-format and cannot be imported into Unity.

MULTI / SUB-OBJECT

Multi/ sub-object is referring to a mesh with different materials assigned.

Supports multi-materials with all available channels.

SHADERS

Unable to export shaders via the FBX-format is NOT possible to export and import shaders generated through SOLIDWORKS, 3DS Max or Maya. But this will not be a problem because unity has a powerful *set of built-in Shaders* that will help us to render our environment.

CONCLUSIONS OF EXPORTING MATERIAL

It was found that UNITY only supports the export of simple materials; properly this is because the interchange format, FBX, converts all materials in simple materials. The materials that make use of reflection, bump or transparency can keep the textures that define it considering that some appropriate modifications within the UNITY engine must be performed. Shaders are not allowed for the exportation, they should be implemented within UNITY.

3.3.4. Scale Level

The scale is critical because UNITY handles equivalence scale in centimeters. For example, 1 unit equate 0.01meters , to giving as a unit scale is 1 inch, it is essential that the models are exported in this way the FBX-format so we must always select the scale manually and choose inches at all times . Different case is the 3d application that models perform in 3d applications can handle the usual equivalence which is 1 unit equals one meter but always remember to export in centimeters.

3.3.5. Light & Shadow Level

Procedure applied:

1. Review of standards of use of lights in three-dimensional real-time applications. In SOLIDWORKS we can find these 3 kinds of lights:

- directional light
- Spot light
- Point light

2. Lighting of several primitive objects and a floor

3. Preparing to Export

4. Export to FBX format

5. Review of problematic export

6. Import into the Unity 3D engine

7. Data analysis and drawing conclusions lights

When we design a model in SOLIDWORKS, we have the possibilities to select various kinds of lights and project the shadow of the Model. When we make the export all

lights are turned into empty objects within Unity. So it does not make sense to study lights and shadows in SOLIDWORKS. It is necessary to create the light and shadow environment within UNITY, so we will study later the optimizers that this software offers us.

3.3.6. Camera Level

Procedure applied:

1. Review of standards use of dimensional cam for real-time applications
2. Creating cameras
3. Export to FBX format
4. Review of export problematic
5. Import into the Unity 3D engine
6. Collection of useful data for using FBX files Unity3D
7. Analysis of data and drawing conclusions cameras

Cameras are the devices that capture and display the world to the player. By customizing and manipulating cameras, you can make the presentation of your game truly unique. You can have an unlimited number of cameras in a scene. They can be set to render in any order, at any place on the screen, or only certain parts of the screen. This task has only sense into unity, because SOLIDWORKS camera is imported as an empty object.

3.3.7. Animation Level

Procedure applied:

Simple animation

1. Creating simple animation
2. Export to FBX
3. Review of problematic export
4. Import into the Unity 3D engine
5. Collection of useful data for using FBX files Unity3D
6. Data analysis and drawing conclusions

We can find two kinds of animations: Simple animation and skeletal animation. Skeletal animation is not possible to create in SOLIDWORKS, so we must do it in Maya or 3DMax.

The IMPORT SETTINGS are dividing in 3 tabs; MODEL, RIG, ANIMATION [22]

- The Model tab has settings for importing the 3D model.
- The Rig tab has settings which will allow this 3D model to support animation.
- The Animations tab has settings for importing one or more Animation Clips from this model file.

MODEL TAB

Before an animation model can be used, it must first be imported into your project. Unity can import files in native format. To import an animation, just drag the file from the model to the Assets folder in your project. When you select the file in the Project View you can edit the Import Settings in the Inspector. The model file is placed in the Assets folder on your Unity project are automatically imported and stored as assets of Unity.

A model file may contain a 3D model, as it is a character, a building, or cabinet. The model is imported into multiple assets. In the Project View, imported main object is a pre-made model. Usually there are up to several Mesh objects that are referenced by the Pre-Made Model.

A file may also contain information model animation that can be used to animate this model or other models. The information of the animation is imported as one or more animation clips.

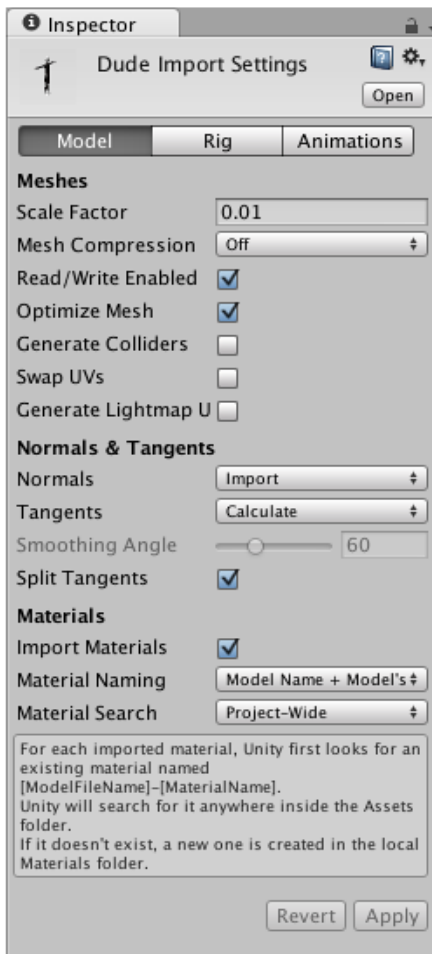


Figure 3.3.7.1 ModelTab

In the Figure 3.3.7.1 ModelTab we can notice some features of the importation.

IMPORT SETTINGS FOR MESHES

The Import Settings for a model file will be displayed on the Model tab inspector. These affect the mesh, normal and imported materials. The settings are applied by asset disk so if you need assets with different settings, make a duplicate file.

Some settings usually to be made are for example:

- Scale: This scale factor is used to compensate for differences in units between Unity and 3D modeling tool: This re-scale the entire file. Normally you can just set it to 1. Note that the physics engine is scaled as if 1 unit is equal to 1 meter. It is important that if you want to have a proper physical behavior, you should have the model successfully climbing the original modeling application. If this cannot be done, or you have no control over the modification of the mesh, the model scale can be set here.

- Generate colliders: this will generate a collision mesh that allows your model collide with other objects.

- Material Naming and Search: this is going to help automatically configure their materials and textures locate.

RIG TAB

The Rig tab allows you to assign or create a definition of your model skinned avatar imported for you to be encouraged

- If you have a humanoid character (i.e., two legs, two arms and a head) Humanoid and then choose "Create from this model ".An avatar will be created to fit as bone hierarchy.
- If you have a non-humanoid character e.g. a quadruped, or any animatable entity you want to use with Mecanim, choose Generic.
- Choose Legacy if you want to use the animation system

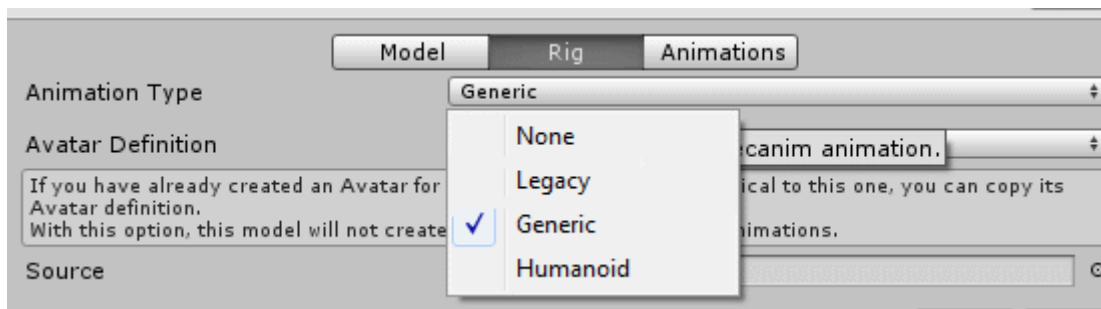


Figure 3.3.7.2 *RibTab*

In this figure 3.3.7.2 we can notice the animation types we should choose depending on our model which are:

- None: No animation present
- Legacy: Legacy animation system
- Generic: Generic Mecanim animation
- Humanoid: Humanoid Mecanim animation system

ANIMATION TAB

In this tab appears animation clips that we want to import.

The Animation Clips are small building blocks of animation in UNITY. They represent the individual parts of the movement, such as RunLeft (Running left), Jump (Jump), or Crawl (Crawling), and can be manipulated and combined in various ways to produce hot end results.

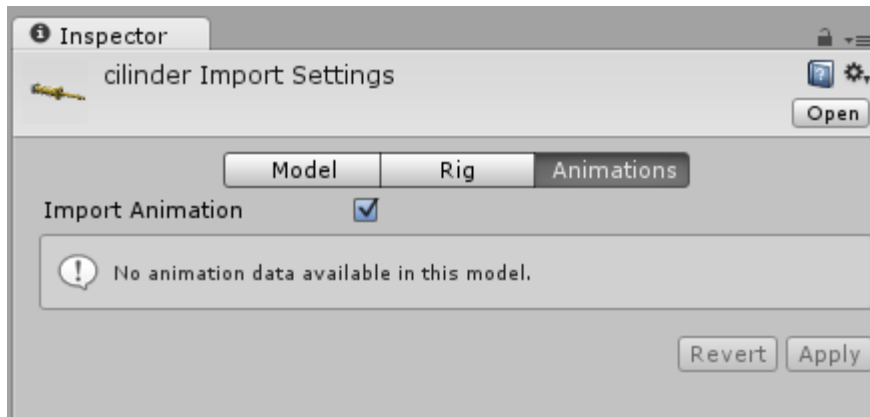


Figure 3.3.7.3 Animation Tab

As we can check out in the Figure 3.3.7.3 the animation import is not available for this model. So when we import animations from SOLIDWORKS, in the most of the cases, the animation will not available in UNITY. To solve this limitation we have to create this animation within unity.

CONCLUSION OF EXPORTING ANIMATION

If we use a program that UNITY 3d support we can carry out with the simple animation. But when we use SOLIDWORKS, which is a program not used to create videogames, we will find some limitations. Our aim is not to import a humanoid animation, because we design architectural buildings or industrial machines or geometries that usually, they do not need more than simple animation clips. So if we have problems with the availability of this clip models in UNITY import settings, we could create it within unity with the assistance of animation. We could do it in a simple and easy way. For SOLIDWORKS files, the most important thing to import is the geometries and the models, meshes and precision that we need to achieve.

3.4. Evaluation of success

When importing SOLIDWORKS files, the software automatically excludes construction history, lines. That could be a problem if we have not defined correctly all of the parts, and assembly we want to work later with in UNITY. For example, if we want to animate a geometry part of a CAD-model in UNITY, and this part has not been sectioned previously in SOLIDWORKS, will be impossible carry it out within UNITY. Because the program will select the whole body instead of the specific part we want to animate or work with. So for this reason is important keep in mind which kind of operations we want to implement later in UNITY. Otherwise there are programs that could help us to modification SOLIDWORKS files according to our target. *Anark Core* [4] is one alternative for this purpose.

Regarding to Geometric Level; Primitives, Splines, Patch, deforming geometry, composed geometry, Nurbs, are not support by FBX-format and all this geometries are converted to polygons (mesh). In the case of polygons, they are converted into triangles.

In Texture level; alpha transparency, bump, multi-material work in the following way. UNITY retains all common types of texture mapped to standard materials, but not always assigned automatically. The unique texture is automatically assigned is it which contains the alpha channel.

Referring to material level; it was found that UNITY only supports the export of simple materials; properly this is because the interchange format, FBX, converts all materials in simple materials. To export models, they should have applied a standard material and can use dissemination textures and some kind of transparency for alphas. For reflection maps, may be included in the FBX export, but should be allocated within the engine later.

In the fields of Scale, light and shadows, it must always be selected the scale manually and choose inches all times into UNITY. Unable to import cameras and lights from FBX files, they must be created directly in UNITY.

As for the simple animations, could potentially be imported without any problems because FBX-format supports animations, SOLIDWORKS animations are not supported into UNITY. That will not be a huge problem because in this project, just simple animations are required, which could be created easily into UNITY. For skeletal animation, various requirements are required.

In the 3 prototypes-files checked we can notice that the geometry exportation is successfully reached, which is the vital part in the project because from this starting point we can adequate the model into UNITY to desired characteristics, using the tools this engine offers.

4. NURB LIMITATIONS

4.1. Differences between Nurbs and Polygons

Mesh polygons (hard surface modeling) and NURBS are two ways to face the same task: create 3D surfaces.

NURBS provide mathematical exact and continuously smooth representation of geometry. It is easier to create organic surfaces or surfaces requiring complicated curves. However, they have the problem that they are built with a mathematical model, which is not possible to render easily. Furthermore, there will be problems in the time we tried to convert Nurbs surface to polygon mesh.

To find an optimal method between mesh polygons and Nurbs that equates the advantages and disadvantages, Subdivision Surface Modeling method is implemented. Modeling Subdivision surface is a method of representing a smoothed surface, "refined" from a simpler polygon mesh.

One of the many advantages of incorporating this method of construction is that, in principle, still starting from a polygonal modeling: they still use the same tools that already existed for classic polygonal modeling (bridge, bevel, extrude ...). But on the other hand, how to use them and work processes change radically: we use a completely different mindset and methodology in the approach to modeling. In fact it is very important to set aside many of the procedures that we could learn modeling with "rigid" polygons or NURBS surfaces.

To better understand how the system works subdivision surfaces we will compare it with two other systems work: the classic polygonal modeling commonly known as "hard surface modeling" and NURBS modeling.

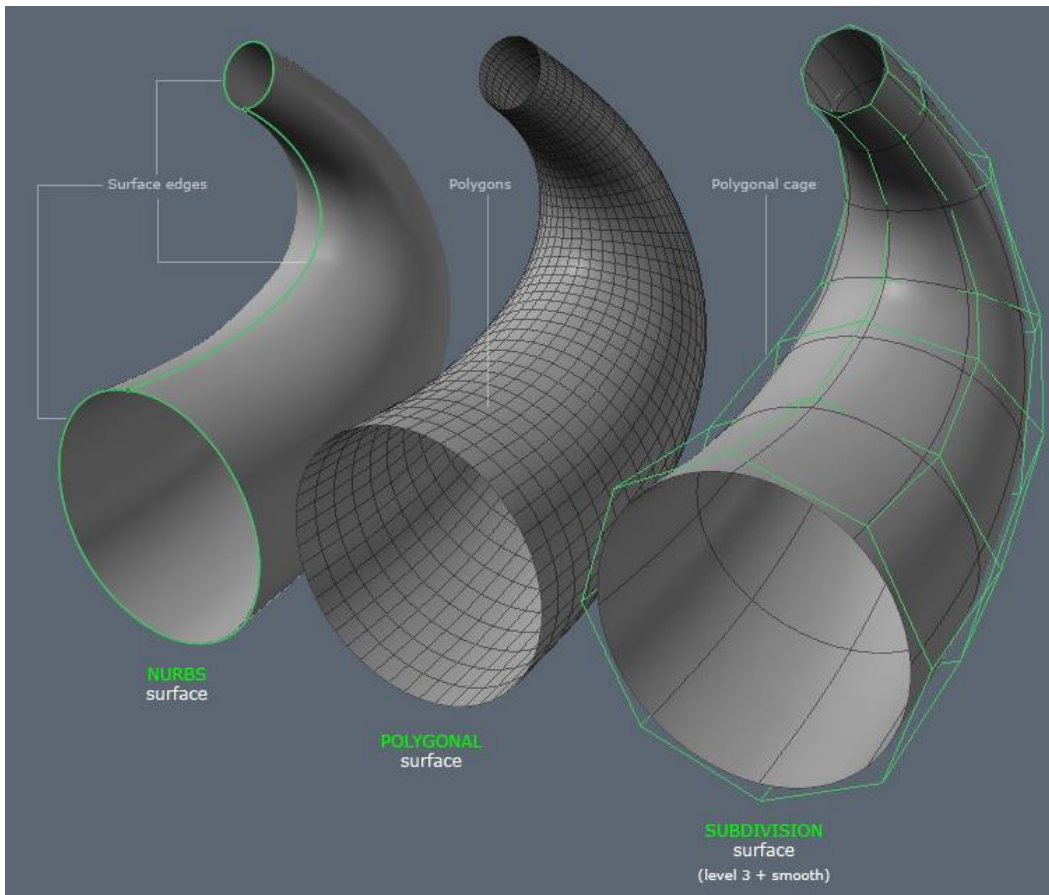


Figure 4.1 Surface Systems [23]

Here you can see three equal objects modeled following three systems:

- In the first, have been used two circular profiles and an extra profile that defines the route to generate a NURBS surface by typical operation "sweep". There aren't really polygons, although usually all render engines must convert these surfaces may be rendered triangles. The great advantage of these surfaces is that they are defined mathematically: however much we approach always looks perfect, which means that exact and continuously smooth representation of geometry is generated.
- In the second we see the same surface defined by polygons, quads, in what would be a classic polygonal surface (hard surface). That object, this distance is perfectly displayed, but if we get too would reach the polygons become evident, especially in the contour of the surface. That means that we loss accurate, so is convenient to work with enough number of polygons depending on the accuracy we want to reach.
- In the third case the same but have other surface defined by subdivision. We see a simple polygon mesh, coarse (green) which is subdivided several times to generate a much denser and refined surface.

Not always the methods of creating a classic polygonal modeling and another subdivision are equivalent. They can become very different. Actually we can model

almost any object with any of these 3 methods, but it is clear that certain objects, certain geometries, are more simple and affordable with one system rather than another. There is not a tool that is best for all. NURBS modeling is ideal for industrial DESIGN and Hard Surface modeling is perfect for modeling ARCHITECTURE. In general it could say that the subdivision modeling is ideal for creating ORGANIC SURFACE. But organic not only in a narrow sense (a head or human body or any animal) but also in a broader sense: any object with a lot of more or less complex curved surfaces such as a car, a plane or a hair dryer. Although NURBS modeling is particularly suitable for such objects DESIGNING, as already mentioned, the truth is that modeling them after a few reference images (where the design work is no longer needed) is often more practical and fluid through a system of subdivision surfaces.

4.2. Limitation, Main problem

Why polygons are used in 3D graphics?

To understand this we have to consider the differences between GPU & CPU [24]

The CPU (central processing unit) has often been called the brain of the PC. However, increasingly, the brain is being reinforced by another part of the PC, the GPU (graphics processing unit), which is its soul.

CPU cores consist of only a few nucleus with a large amount of cache memory that can handle a few threads of software at once. On the other hand, GPU is composed of hundreds of nucleus that can handle thousands of data simultaneously.

The main point is that Nurbs are controlled by the CPU and Polygons are controlled by GPU, and today, the GPU can operate faster and more cost-efficient than microprocessors in a wide range of sectors.

Thanks to advances in graphics processing technology, today's video cards can render millions of polygons per second. This allows for several highly detailed 3D models to be displayed on the screen at one time. Polygonal modeling is well suited to scanline rendering and is therefore the method of choice for real-time computer graphics

The last conversion of a Nurbs to a mesh is a triangle

As realistic as 3D graphics may appear, they are typically comprised of flat, multi-sided polygons. These polygons are placed together to create a three-dimensional mesh, which produces a 3D image. While most polygons are triangles (which have the fewest possible sides), they can also be rectangles, hexagons, or other shapes. Colors and textures can be mapped onto these polygons, giving the final image a realistic appearance.

We will concentrate on triangles mostly because of one main reason: every object can be split into triangles but a triangle cannot be split into anything else than triangles. Because of this, drawing triangles is a lot simpler than drawing polygons of higher order; less things to deal with. This is why those triangles are so commonly used in computer graphics.

Because polygons are flat surfaces, they can only estimate curved shapes, which many objects have. Therefore, smaller polygons can more accurately represent curved objects. Of course, using smaller polygons also means more polygons are required to create the object. So the more polygons a 3D model has, the more smooth and realistic it will look.

Rendering Problem

NURBS need much power of mathematical calculation of curves, while the polygons not. To get a detailed polygon mesh you need many polygons, but to make a detailed mesh NURBS only takes several checkpoints and much power calculation. Depending on our necessities one or other model will be chosen.

NURBS surfaces are complex to render. To do this task, several programs are used to convert polygon meshes, simplifying them. This simplification is a problem and not really easy to carry out.

Now, here's the problem. How to render these surfaces? Rendering NURBS directly can be done, but it's slow, especially for anything approximating "real-time" graphics. So what applications do is create polygon meshes fitted to the surfaces and render that. Meshes are fast to render because they are a finite number of points to arrange in a certain order with flat areas between them, and the math involved with rendering flat things is relatively simple.

Rendering engines use "tricks" to make meshes appear to be smooth despite being composed of an often remarkably small number of flat polygons. We can adjust settings so that meshes look smoother, but there is always a compromise between speed and quality. If we have the horsepower to make the polygons smaller than the pixels in the image, we wouldn't have any hassles.

4.3. Alternatives and Solutions

After an overview, and studying advantages and drawbacks of both surface modeling, Nurbs and mesh polygons is necessary to focus on the way to convert Nurbs Surfaces to polygons mesh, which is used in 3D graphics. There are many programs that carry on with the task to convert a mathematical NURB model into polygons. Notice that in this project we only have made tests with Simlab 3d Plug-in [2.4 Interchange format].

- Simlab 3d Plug-in [18], the interchange format we are using to develop our project. For our models, it is working properly, but in case you need specific quality of the meshes you should use 'nPower software'.

- Autodesk Maya LT [25], you can convert a NURBS surface to a polygon mesh whether it was created in Maya LT or imported from another 3D application. Trimmed surfaces are also converted in the process. So the way to act here, would be exporting your model to Maya, and then make the transformation there.

- nPower Software, power translation [26], is a revolutionary SOLIDWORKS plugin product that facilitates the rendering of native SOLIDWORKS models and assemblies through its new Power SolidWorksToMax translator. 'nPower SolidWorksToMax' greatly improves the work flow efficiency for design visualization; Now with Power SolidWorksToMax, SOLIDWORKS users can import their high quality models directly into Autodesk 3ds Max for high quality rendering, without imprecise polygonal models and without intermediate formats, and without loss of precision. An important feature is that this Program can import Nurbs without converting them into polygons. Notice that we have not tested this program.

5. OPTIMIZERS IN UNITY

5.1. Optimizers and their use in Unity

Once the model has been imported into UNITY, is time to implement the possibilities UNITY provides users to improve the Initial SOLIDWORKS design file. Here we study the amount of optimizers UNITY provide us to develop our model.

Review of standards optimization for real-time scenes within Unity

A. Visibility Culling

Cut what is not seen, there are four cases:

- Culling Occlusion
- Viewfrustum culling
- Culling backface
- Contribution Culling

B. Layer

C. Per-layer Culling

D. Camera

E. PHYSICS system

F. Animations

G. Baked texture and Shadows

H. Lights

I. Shaders

A. VISIBILITY CULLING

Defining the concept of visibility culling, consist of cut what does not see, to avoid excessively the objects overdrawing and therefore the optimization of the model.

There are four possible cases: [27]

- Occlusion Culling - Hidden Faces behind other objects
- View-frustum culling - which is outside the cone of vision
- Backface Culling - faces that the observer doesn't see.
- Contribution Culling - Those faces who are too small to contribute to the image

Based on the experimental use of FBX-format, UNITY found that you cannot export any Visibility culling so this must be done within UNITY.

EACH TEST WAS PERFORMED IN UNITY FINDING THAT:

Occlusion culling

Occlusion Culling is a feature that disables rendering of objects when they are not currently seen by the camera because they are obscured (occluded) by other objects.

It is possible to optimize Occlusion Culling within UNITY, we must define geometry as static and then in the window Occlusion Culling, modify it later to create some values occlusion areas, and then calculate them and save them in a file independent.

View-frustum culling

The view frustum is the volume that contains everything that is potentially visible on the screen. This volume is defined according to the camera's settings, and when using a perspective projection takes the shape of a truncated pyramid. View frustum culling is built into every camera unity and values can be manually changed by the user defining where the cone projection starts and where it is cut by the cutting plane.

To perform efficiently in UNITY, frustum culling objects must be separate, can these be hierarchical but should be recognized for UNITY as separate objects, therefore it is advisable to separate geometries that occupy much space on drives. For example, two buildings those are distant.

Culling backface

Back-face culling determines whether a polygon of a graphical object is visible. It is a step in the graphical pipeline that tests whether the points in the polygon appear in clockwise or counter-clockwise order when projected onto the screen. If the user has specified that front-facing polygons have a clockwise winding, but the polygon projected on the screen has a counter-clockwise winding then it has been rotated to face away from the camera and will not be drawn.

The process makes rendering objects quicker and more efficient by reducing the number of polygons for the program to draw.

All UNITY shaders, except those used for vegetation, have active backface culling test so the calculation of the reverse side of the normal polygons are not represented. To change this optimization must be edited or created a shader that contains a null test for face culling.

Contribution Culling

There is not direct implementation of this algorithm in UNITY, it is necessary to program it.

B. LAYER

Layers are most commonly used by Cameras to render only a part of the scene, and by Lights to illuminate only parts of the scene. Once a layer is created, game objects can be assigned to them by choosing the desired layer under the options box next to Layer under that game object's inspector window. This way, you can group objects in common layers for later use and manipulation. Keep in mind what layers are and how to create and modify them for when I talk about a few other layer features later in the paper.

C. PER-LAYER CULLING

Your camera will not render game objects beyond the camera's clipping plane in Unity. There is a way, through UNITY scripting, to have certain layers set to a shorter clipping plane.

D. CAMERA

When creating a new scene, by default, there is only one camera game object labeled *Main Camera*. To create or add another camera, first create an empty game object by going to: Game Object->Create Empty. Then select the newly created empty object and add the camera component: Components->Rendering->Camera.

Unity's camera comes with a host of functionality inside its Guide.

E. PHYSICS SYSTEM

Meshes

Materials are used in conjunction with mesh renderers, particle systems and other rendering components used in UNITY. They play an essential part in defining how your object is displayed.

Meshes make up a large part of your 3d worlds. Aside from some asset store plugins, UNITY does not include modelling tools. UNITY does however have great interactivity with most 3d modelling packages. UNITY supports triangulated or quadrangulated polygon meshes. Nurbs, nurms, subdivide surfaces must be converted to polygons.

RigidBody

Rigid bodies enable your GameObjects to act under the control of physics. The Rigidbody can receive forces and torque to make your objects move in a realistic way. Any GameObject must contain a Rigidbody to be influenced by gravity, act under added forces via scripting, or interact with other objects through the NVIDIA PhysX physics engine.

Boxes collision & mesh collider

The Mesh Collider builds its collision representation from the Mesh attached to the GameObject, and reads the properties of the attached Transform to set its position and scale correctly. You can define the invisible geometry as collision boxes within UNITY, these boxes can be exported from modeling programs, although it is also possible to assign collision geometry directly without create a modeling software, this will need to select the object and then define the type of collision as required. If bit complex then choose the cash even if more accuracy is required shall be chosen sphere and so on until you reach a high level of detail of the geometry of collision.

The real-time physics will be implemented directly in UNITY, it is not possible to export models with physical assigned from SOLIDWORKS.

F. ANIMATIONS

The Animation View in UNITY allows you to create and modify Animation Clips directly inside UNITY. It is designed to act as a powerful and straightforward alternative to external 3D animation programs. In addition to animating movement, the editor also allows you to animate variables of materials and components and augment your Animation Clips with Animation Events, functions that are called at specified points along the timeline.

G. BAKED TEXTURE AND SHADOWS

When light falls on an object, shadows are created that highlights the three dimensional form. These shadows will also highlight bumps and other impurities on the surface. Simulating the way light falls on a 3D subject (or Avatar) may make it appear more natural and realistic. It is possible to use rendering algorithms on a computer to simulate shading, shadows and bumps on the surface (or texture) of a 3D object. This process is commonly referred to as 'baking' the texture.

Texture baking, allow you to create texture maps based on an object's appearance in the rendered scene. [28]You can export the FBX model with baked maps but these must be in complete map or alternatively have separate lightmap for this is used in 3d, which will not be possible in SOLIDWORKS. Moreover, it is possible to generate maps lights inside UNITY giving it a plus in the pro version because this version can pre-compute global illumination and automatically generate maps defining quality and the complexity and detail of automatic geometric projection, so it is not necessary to pre-calculate from the modeling program.

H. LIGHTS

In order to calculate the shading of a 3D object, UNITY needs to know the intensity, direction and color of the light that falls on it. There are different kinds of lights: Point Lights, Spot Lights, and Directional Lights.

Furthermore, we should evaluate the Global Illumination option in certain cases.

Global Illumination (GI) is a system that models how light is bounced off of surfaces onto other surfaces (indirect light) rather than being limited to just the light that hits a surface directly from a light source (direct light). Modelling indirect lighting allows for effects that make the virtual world seem more realistic and connected, since objects affect each other's appearance. [29]

In the following Figure we can see the different Light settings and how to create shadows, with the features of Shadow type, Strength and resolution.

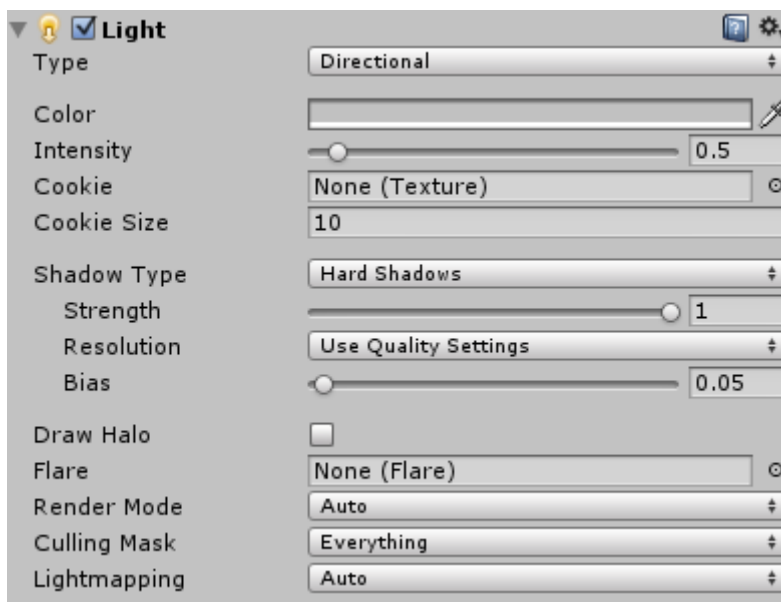


Figure 5.1 Light Settings

I. SHADERS

The properties that a Material's inspector displays are determined by the Shader that the Material uses. A shader is a specialized kind of graphical program that determines how texture and lighting information are combined to generate the pixels of the rendered object onscreen.

Shaders define both how an object looks by itself (its material properties) and how it reacts to the light. Because lighting calculations must be built into the shader, and there are many possible light-shadow types, writing quality shaders that "just work" would be an involved task. To make it easier, UNITY has Surface Shaders, where all the lighting, shadowing, lightmapping, are taken care of automatically.

There is a close relationship between Materials and Shaders in UNITY. Shaders contain code that defines what kind of properties and assets to use. Materials allow you to adjust properties and assign assets.

A set of built-in Shaders are installed with the UNITY editor. The main ones used for texturing game objects fall into the following categories: [30]

- Normal: For opaque textured objects.
- Transparent: For partly transparent objects. The texture’s alpha channel defines the level of transparency.
- Transparent Cutout: For objects that have only fully opaque and fully transparent areas, like fences.
- Self-Illuminated: For objects that have light emitting parts.
- Reflective: For opaque textured objects that reflect an environment Cubemap.

In the Figure 5.1.2 we can see the different kind of shaders can be easily implemented in UNITY.

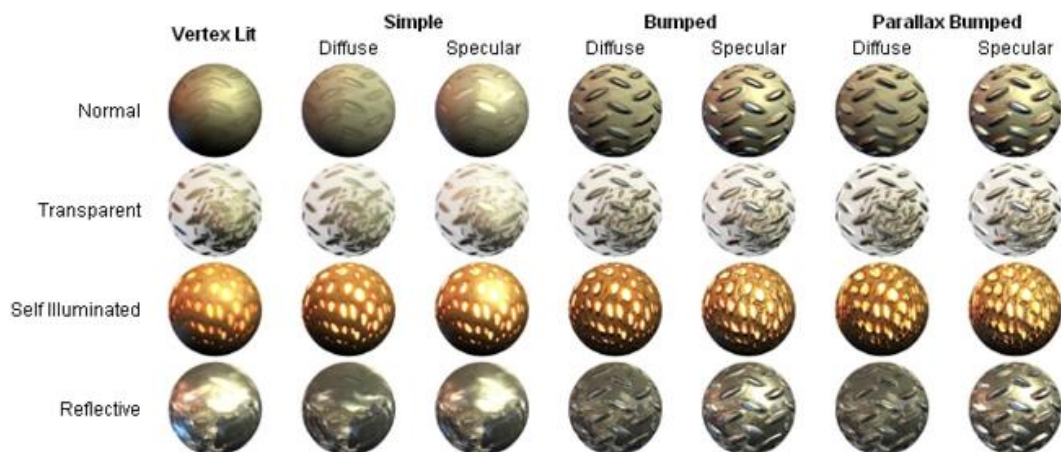


Figure 5.1.2 Shaders

J. LEVEL OF DETAIL

Level of Detail (LOD) allows multiple meshes to attach to a game object and provides the ability to switch between meshes the object uses based on camera distance. This can be beneficial for complex game objects that are really far away from the camera. The LOD can automatically simplify the mesh to compensate. [31]

You cannot export levels of detail from SOLIDWORKS to UNITY using FBX-format, so the use of UNILOD [33] within unity is recommended. This application is acquired from Asset Store to modification the meshes. This extension of unity allows managing

dynamic levels of detail in the pro version automatically, however in the User version the automatic simplification of meshes is not possible.

5.2. Asset Store

This is a Store where you can find lots of Assets and applications already created in UNITY and uploaded for be used by developers.

Standard Assets are the set of tools and prefabs to use as a building block in creating our design or even just to get our kickstarted in prototyping. So if we need a First or Third Person controller, a Car, a Plane or 3D character, look no further. It also includes useful tools like Touch control setups.

In this store you can find a big amount of applications that can help to create our design within UNITY. Some of them are free, and some not. The main assets are: 3D models (Characters, Environments, Props, Vegetation, Vehicles, Other); Animations (Bipedal, Other); Audio (ambient, music, Sound Fx, Other); Complete project (Tutorials, Packs, Unity Demos); Editor Extensions (Game toolkits); Particle Systems; Scripting; Services; Shaders; Textures & Materials; Unity Essentials.

In this project we are going to focus on some applications that will help us to reach our target.

5.2.1. Mega Fiers

A complete Mesh deformation, animation and Morphing system including over 50 modifiers such as Bend, Twist, FFD, Displace, Taper. They can be stacked in any combination and any number of modifiers can be applied to a mesh to achieve complex results. Objects can stretch, squash, morph or be bent over or deformed, whatever your imagination allows. The morph system is the most advanced available for UNITY. Recent additions to the suite include Point Cache animation support; physics based rope system and a dynamic water ripple and floating object system. MegaFiers is written in C# and all source code is included, also it makes use of multiple CPU's for amazing performance and works on all platforms and with the free version of Unity.

Also included is an advanced spline system which also allows for path following or conversion of splines to meshes, a dynamic hose system and a system for tracked vehicles, and an advanced mesh wrapping system for applying clothing etc. to deforming objects. [21]

That could complete the Geometrical level, in the field of deforming Geometry.

5.2.2. Unilod

This extension of unity allows managing dynamic levels of detail in the pro version automatically. This application will create mesh that has reduced triangles, vertices, normals and Coordinates UV(letters "U" and "V" denote the axes of the 2D texture). No other 3D applications are required for this less-detailed-mesh operation; UNITY supported the entire creation of LOD(level of Detail) mesh. It is old fashion way reducing polygons, but since it is not hardware dependent approach, this algorithm works on any platforms and still many games use this kind of approach.

This editor script works creating series of meshes in a similar way as mipmap generation algorithm. Mipmaps are pre-calculated, optimized sequence of textures that accompany a main texture, each of which is a progressively lower resolution representation of the same image. The height and width of each image, or level, in the mipmap is a power of two smaller than the previous level. Mipmaps do not have to be square. They are intended to increase rendering speed and reduce aliasing artifacts. [32]

The main Features of this app are the following ones: [33]

- Automatic mesh simplification. It's only achieved in Windows with Unity Pro version.
- Generate meshes with reduced polygon count automatically
- Easily create LOD prefabs that change quality levels or that disappear at a certain distance.
- Group objects together into a single mesh and texture to reduce draw calls. Draw calls are total of meshes drawn after the batching process where batching is the process where the rendering engine tries to combine multiple objects in a single draw call to reduce the load on the CPU.

That could complete the Optimizers in UNITY, in the field of Level of Detail (LOD).

5.3. Manager Settings

Unity's Settings Managers are available from the menu Edit > Project Settings. The settings affect overall aspects of Unity's functionality, such as Graphics, Physics and the details of the published player. The various managers are described in detail in this section.

QUALITY SETTINGS

UNITY has extra render settings for games found in: Edit->Project Settings->Quality menu (Figure5.3.1). They are customizable render settings that can be modified for

individual needs. Unity has helpful online documentation for explaining what the Quality Settings are and how to modify these settings through Unity's scripting API.

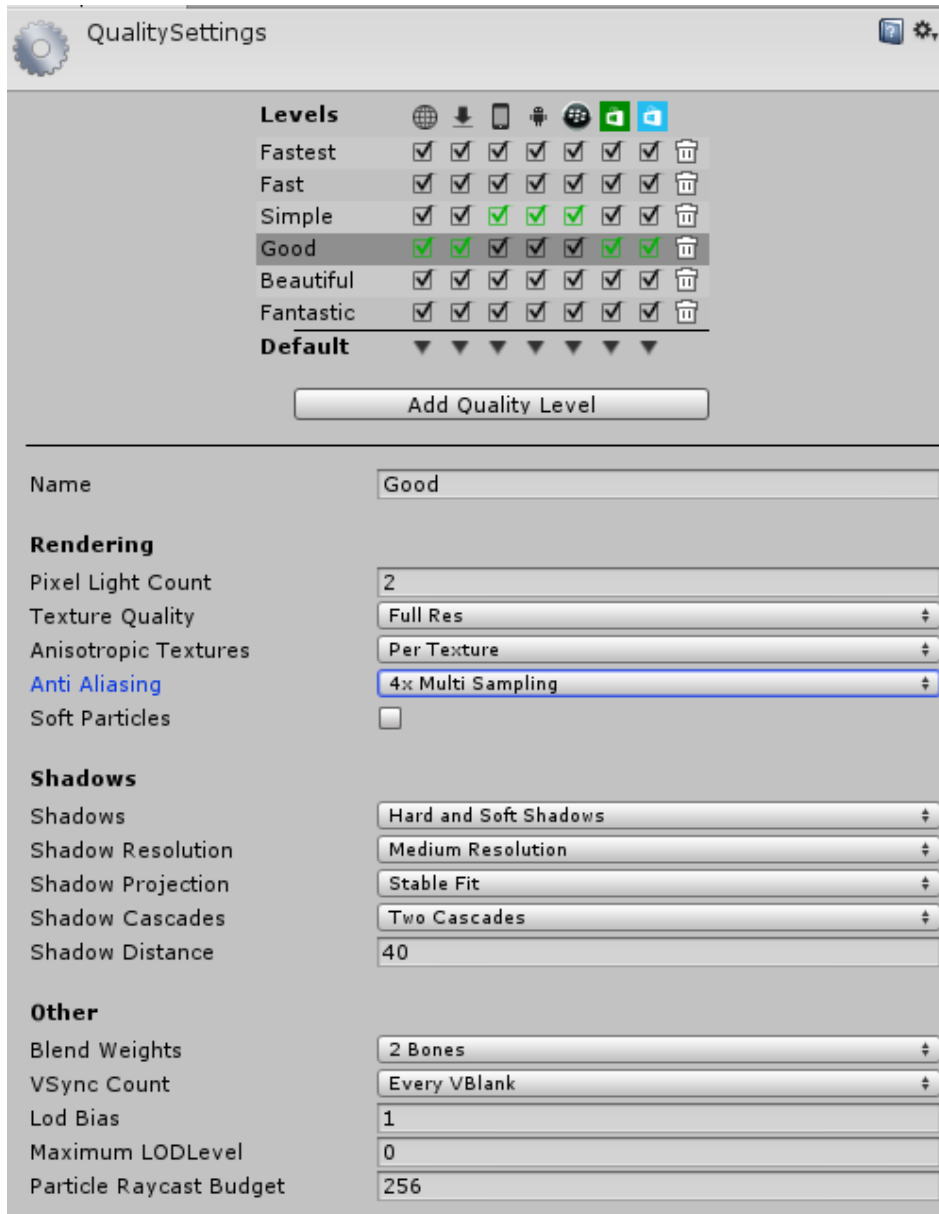


Figure 5.3.1 Quality settings

Find attached in the Table5.3 the followings features of the quality settings in UNITY.

[34]

Property:	Function:
Rendering	
Pixel Light	The maximum number of pixel lights when Forward Rendering is used.

Property:	Function:
	Unity supports different Rendering Paths. You should choose which one you use depending on your game content and target platform/hardware. Different rendering paths have different performance characteristics that mostly affect Lights and Shadows. Forward Rendering is the traditional rendering path. It supports all the typical Unity graphics features (normal maps, per-pixel lights, shadows etc.).
Texture Quality	This lets you choose whether to display textures at maximum resolution or at a fraction of this (lower resolution has less processing overhead). The options are Full Res, Half Res, Quarter Res and Eighth Res.
Anisotropic Textures	This enables if and how anisotropic textures will be used. The options are <i>Disabled</i> , <i>Per Texture</i> and <i>Forced On</i> (i.e., always enabled). Anisotropic filtering (abbreviated AF) is a method of enhancing the image quality of textures on surfaces of computer graphics that are at oblique viewing angles with respect to the camera where the projection of the texture.
AntiAliasing	The options are 2x, 4x and 8x multi-sampling. Antialiasing is the removal of too high frequency information to be represented. The most important value to modify is Anti-aliasing. The anti-aliasing must be increased to compensate for the stereo rendering, which reduces the effective horizontal resolution by 50%. An anti-aliasing value of 4X or higher is ideal. However, if necessary, you can adjust to suit your application needs.
Shadows	
Shadows	This determines which type of shadows should be used. The available options are <i>Hard and Soft Shadows</i> , <i>Hard Shadows Only</i> and <i>Disable Shadows</i> .
Shadow resolution	Shadows can be rendered at several different resolutions: Low, Medium, High and Very High. The higher the resolution, the greater the processing overhead.
Shadow Projection	There are two different methods for projecting shadows from a directional light. Close Fit renders higher resolution shadows but they can sometimes wobble slightly if the camera moves. Stable Fit renders lower resolution shadows but they don't wobble with camera movements.
Shadow	The number of shadow cascades can be set to zero, two or four. A

Property:	Function:
Cascades	higher number of cascades gives better quality but at the expense of processing overhead.
Shadow Distance	The maximum distance from camera at which shadows will be visible. Shadows that fall beyond this distance will not be rendered.
Other	
Blend Weights	The number of bones that can affect a given vertex during an animation. The available options are one, two or four bones.
VSync Count	Screen tearing is a visual artifact in video display where a display device shows information from two or more frames in a single screen draw. Vertical synchronization (VSync) is an option in most systems; where in the video card is prevented from doing anything visible to the display memory until after the monitor finishes its current refresh cycle. Rendering can be synchronised with the refresh rate of the display device to avoid "tearing" artifacts .You can choose to synchronise with every vertical blank (VBlank), every second vertical blank or not to synchronise at all.
Maximum LOD Level	The highest LOD (level of detail) that will be used by the game. See note below for more Information.

Table 5.3 *Quality settings features*

BUILD SETTINGS

Generating a game PC version is done easily and quickly. You reach "File> Build Setting". After these steps, the screen below appears and where it can be seen that unity gives us an option to compile the game for several PC, iOS, Android, Xbox 360, etc. platforms although in most cases is prepaid. In the case of PC is completely free and just select this option and click "Build" then a folder called "PC" is generated. With this option, once the game is burn we can run it within Oculus Rift Device.

Find attached in the Figure 5.3.2 the Build settings of our design. [35]

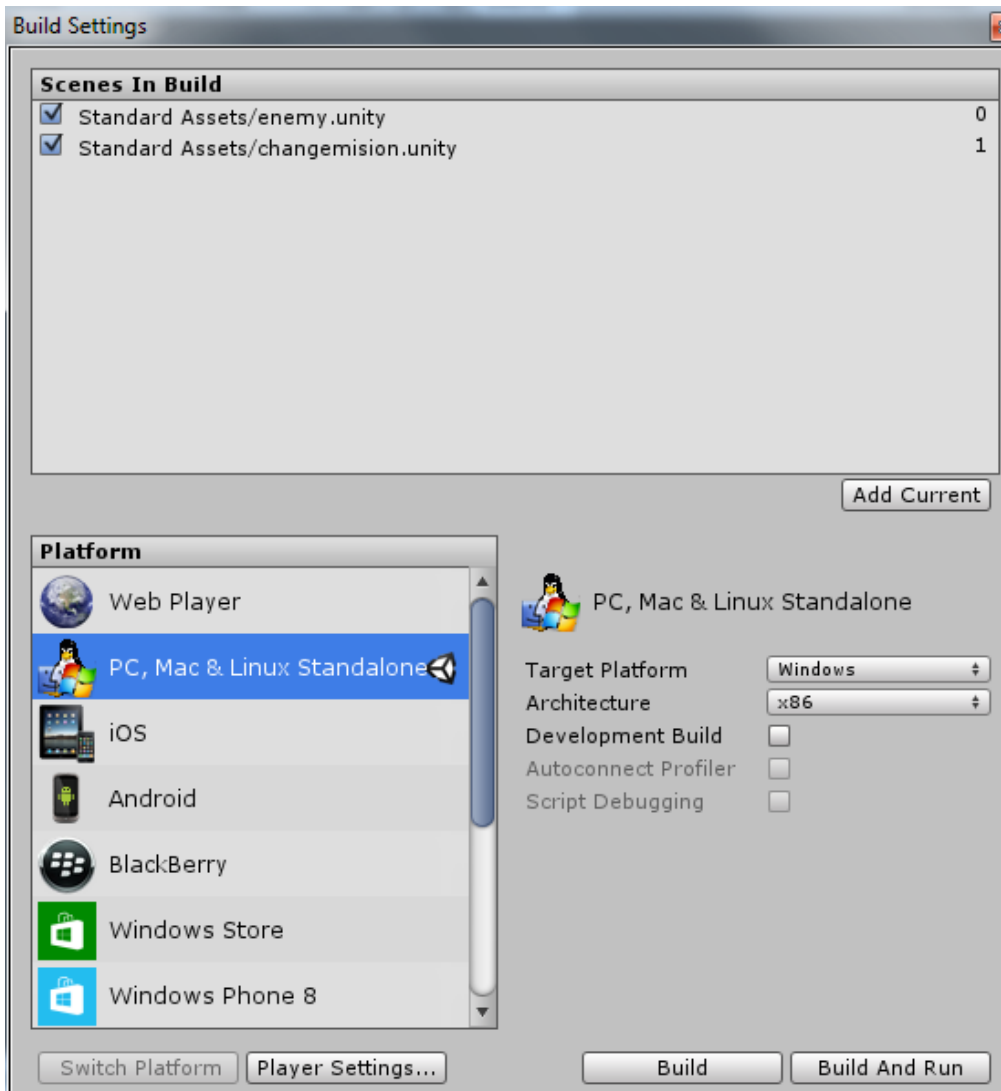


Figure 5.3.2 *Build Settings*

6. CONCLUSIONS AND PERSPECTIVES

A new paradigm for 3D immersive project review has been developed by using a VR environment. It has been tested successfully for feasibility on real cases and is naturally adapted to a variety of situations involving industrial designs. This work provides a proof-of-concept of a new way of using 3D models data and interactive simulation in a widespread industrial procedure. This work explores how companies could improve significantly the examination of designs, providing an efficient and better understanding of the industrial model, including non-technical consumers.

In this document, we have used different models, carefully chosen for a given conversion application, checking a wide modeling features, such as geometries, textures, animations, etc. In the future, it will be interesting to consider almost all industrial possible representations models with the aim to standardize the process for every kind of industrial models. This task must include implementation of another industrial interchange-format programs and continuous evaluation of conclusions.

This project presents the effects that the projection of 3D models in Head-mounted-display could reach, finding the most impressive thing about the Oculus Rift isn't even the hardware, but the complete and useful platform for virtual reality Oculus Rift has created. The SDK (software development kit) allows us to easily integrate Oculus Rift into our project, with a non-wide knowledge about VR. And it's even easier for us using UNITY through its efficient virtual environment. With regards to UNITY, some optimizers limitations are found due to on occasion the professional-UNITY version is required. This professional suite of tools gives support to all platforms and allows achieving a high rendering power.

This thesis has investigated how VR can be implemented in the industry design process. The possibilities implementing new methods and framework for the use of VR are endless and there are many aspects of this research that can serve as a basis for further research and development.

REFERENCES

- [1] P.Fillatreau, J.Y. Fourquet, R. Le Bolloc'H, S. Cailhol, A. Datas, B. Pue, Using virtual reality and 3D industrial numerical models for immersive interactive checklists, *Computers in Industry*, Elsevier, 2013, vol. 64, pp. 1253-1262.
- [2] Tainen, T., Ellman, A., Kaapu, T., 2011: Three Frames for Studying Users in Virtual Environments: Case of Simulated Mobile Machines. *Proceedings of 15th IEEE/ACM International Symposium on Distributed Simulation and Real Time Applications*. 4-7.9.2011, Salford UK. pp 115-119.

Kaapu, T., Tiainen, T., Ellman, A., 2014: User Interpretations of Virtual Prototypes: Physical Place Matters. *Scandinavian Journal of Information Systems*, 2013, 25(2), pp 3–24.
- [3] G. Morin, Habilitation à Diriger les Recherches in computer science. 3D models for..., Toulouse, 2014, M.sc thesis pp 1-12.
- [4] Anark Corporation, Anark core from, <http://www.anark.com/core/>.
- [5] Autodesk® DirectConnect Users's guide, version 2015,from, http://download.autodesk.com/us/support/files/maya_2015_assets_checklist/directconnect_2015_help.pdf
- [6] D. Systemes, support SOLIDWORKS web help,solidworks fundamentals, 2015 edition,from, http://help.solidworks.com/2014/english/SolidWorks/sldworks/c_open_document.htm?id=c1527a491a6a4366b85c43a00217c2b6#Pg0
- [7] D. Systemes, support SOLIDWORKS web help, Import/Export, Importing/ exporting SOLIDWORKS documents, 2015 edition, from, http://help.solidworks.com/2015/English/SolidWorks/sldworks/c_Importing-Exporting_SOLIDWORKS_Documents.htm?id=b1d9d01860434208a66a1790fb255cfb#Pg0
- [8] D. Systemes, Support SOLIDWORKS web Help,import/export, file types, 2015 edition, from, http://help.solidworks.com/2014/English/SolidWorks/sldworks/c_File_Types.htm?id=b630fd37f8fe428b94ee844550a054aa#Pg0
- [9] D. systemes, Support SOLIDWORKS web help,Assemblies,Component suppression states, 2015 edition,from, http://help.solidworks.com/2015/English/SolidWorks/sldworks/c_Component_Suppression_States_SWassy.htm?id=bb4d3eee5e6046dd84b9de4b1aca73df#Pg0
- [10] O. Mazaný, Articulated 3D human model and its animation for testing and learning algorithms of multi-camera systems, Czech Technical University,Prague,2007,M.sc thesis, pp 16-34, from

<https://cyber.felk.cvut.cz/research/thesis/papers/19.pdf>

- [11] Unity 3D,Unity documentation manual,graphics, How do i import models from my 3D app?, 2015 edition,from
<http://docs.unity3d.com/Manual/HOWTO-importObject.html>
- [12] Blender extensions,from
http://wiki.blender.org/index.php/Extensions:2.6/Py/Scripts/Import-Export/Wavefront_OBJ
- [13] V. H. Franco Serrano, Compatibilidad del formato FBX,Departamento de Realidad virtual, UNAM ,Mexico,2009,M.sc thesis,from
<https://docs.google.com/viewer?a=v&pid=sites&srcid=ZGVmYXVsdGRvbWFpbnydnVuYWw18Z3g6MTZiNTJINzg2NTYxZWnkMQ,UNAM,Departamento de Realidad virtual>
- [14] Oculus, Oculus developer Guide,Overwiev SDK, version 0.5.0, March 2015.
- [15] Oculus, Oculus developer Guide, display device management-pag 40, version 0.5.0, from,
<https://developer.oculus.com/documentation/>, March 2015.
- [16] Oculus, Oculus developer guide,inizialitation and sensor enumeration pag 10,March 2015:
<https://developer.oculus.com/documentation/>, version 0.5.0.
- [17] Oculus, Oculus Unity integration Guide,prefabs,pag 10, version 0.5.0, March 2015.
- [18] Simlab-soft, Simlab 3D plugins, simlab FBX exporter for Solidworks, 2014 Simulation Lab Software, from
http://www.simlab-soft.com/3d-plugins/SolidWorks/fbx_exporter_for_SolidWorks-main.aspx, 2014 Simulation Lab Software.
- [19] Autodesk, FBX converter, 2013.3. <http://www.autodesk.com/products/fbx/overview>
- [20] Unity Asset Store, Rage Spline 15, from,
https://www.assetstore.unity3d.com/en/?gclid=CNLo9dzV_MMCFSYUwwodNEAAuw&gclid=CjwKEAjw876oBRCYr86w6KGfpgSJAAClidwYUnTTci6EN9Od99qnlToiosgD7QM6iHthxA79HO6vRoCHM3w_wcB#!/content/555.
- [21] Unity Asset Store,Mega Fiers16, from,
https://www.assetstore.unity3d.com/en/?gclid=CNLo9dzV_MMCFSYUwwodNEAAuw&gclid=CjwKEAjw876oBRCYr86w6KGfpgSJAAClidwYUnTTci6EN9Od99qnlToiosgD7QM6iHthxA79HO6vRoCHM3w_wcB#!/content/644.
- [22] Unity 3D,Unity documentation manual,animations, importing models, 2015 edition,from
<http://docs.unity3d.com/Manual/class-FBXImporter.html>

- [23] C. Vila, Introducción al modelado por subdivisión, Zaragoza, Spain, 2014, from http://www.etereaestudios.com/training_img/subd_tips/introduccion.htm.
- [24] K. Krewell, Nvidia blog, What's the Difference Between a CPU and a GPU?, 2009, from <http://blogs.nvidia.com/blog/2009/12/16/whats-the-difference-between-a-cpu-and-a-gpu/>
- [25] Autodesk, Maya LT, 2015 edition, from, <http://knowledge.autodesk.com/support/maya-lt/learn-explore/caas/CloudHelp/cloudhelp/2015/ENU/MayaLT/files/Polygon-selection-and-creation-Convert-NURBS-surfaces-to-a-polygon-mesh-htm.html>, 2015.
- [26] nPower Software, Power translation basic for 3Dmax documentation, <http://www.npowersoftware.com/NewPowerTranslatorsOverview.html>.
- [27] Unity 3D , Unity documentation Manual, graphics, graphics overview, camera, 2015 edition, from, <http://docs.unity3d.com/Manual/OcclusionCulling.html>
- [28] Unity 3D , Unity documentation Manual, graphics, lightmapping, 2015 edition, from <http://docs.unity3d.com/es/current/Manual/Lightmapping.html>
- [29] Unity 3D , Unity documentation Manual, graphics, lighting, global illumination, 2015 edition, from, <http://docs.unity3d.com/Manual/GIIntro.html>
- [30] Unity 3D , Unity documentation Manual, graphics, graphics overview, shaders, 2015 edition, from, <http://docs.unity3d.com/Manual/Shader.html>
- [31] Unity 3D , Unity documentation Manual, graphics, graphics overview, Advanced Rendering Features, LOD, 2015 edition, from, <http://docs.unity3d.com/Manual/LevelOfDetail.html>
- [32] macroseed projects, http://www.macroseed.com/Projects/Unity/Level_of_Detail_System
- [33] Unity forums, UniLOD BETA, Level of detail and streaming support, 2008, from, <http://forum.unity3d.com/threads/unilod-beta-level-of-detail-and-streaming-support-update.42524/>
- [34] Unity 3D , Unity documentation Manual, editor, editor reference, settings manager, quality settings, 2015 edition, from, <http://docs.unity3d.com/Manual/class-QualitySettings.html>
- [35] I. Auzani, Manual de creacion de videojuego con Unity 3D, Carlos III University, Madrid, Spain, 2012, M.sc thesis, pp 132, from, http://www.academia.edu/5171832/MANUAL_DE_CREACION_DE_VIDEOJUEGO_CON_UNITY_3D

APPENDIX

Figure 1.2 <i>Graphics Pipeline</i>	3
Figure 2.1 <i>SOLIDWORKS native extensions</i>	7
Table 2.1 <i>Import & Export SOLIDWORKS Formats</i>	9
Table 2.2.1 <i>Unity 3D package support</i>	15
Figure 2.2.1 <i>Import Inspector</i>	16
Table 2.2.2 <i>Comparison of FBX and OBJ formats properties</i>	18
Figure 2.3.1 <i>Oculus Rift HMD</i>	19
Figure 2.3.1.1 <i>Rift display mode dialog Box</i>	20
Figure 2.3.1.2 <i>Extended display mode</i>	21
Figure 2.3.1.3 <i>Head tracking-sensors</i>	22
Figure 2.3.2.1 <i>Prefabs OVRCameraRig, expanded in the editor</i>	23
Figure 2.3.2.2 <i>Prefabs OVRPlayerControler, extended in the editor</i>	24
Figure 2.4 <i>Interchange-format used in Solidworks</i>	25
Figure 3.2.1 <i>C) Pipe system</i>	28
Figure 3.2.2 <i>A) Crane</i>	28
Figure 3.2.1 <i>B) Hydraulic Cylinder</i>	28
Figure 3.3.7.1 <i>ModelTab</i>	38
Figure 3.3.7.2 <i>RibTab</i>	39
Figure 3.3.7.3 <i>AnimationTab</i>	40
Figure 4.1 <i>Surface Sitems</i>	43
Figure 5.1 <i>Light Settings</i>	51
Figure 5.1.2 <i>Shaders</i>	52
Figure 5.3.1 <i>Quality settings</i>	55
Table 5.3 <i>Quality settings features</i>	57
Figure 5.3.2 <i>Build Settings</i>	58