

# **DEVELOPMENT OF AN INTERACTIVE VIRTUAL ENVIRONMENT FOR TRAINING MANUFACTURING OPERATORS**

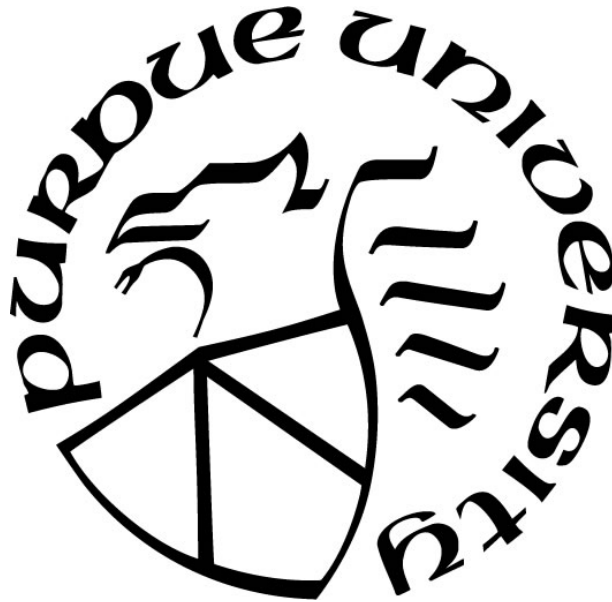
by  
**Aoya Sun**

**A Thesis**

*Submitted to the Faculty of Purdue University*

*In Partial Fulfillment of the Requirements for the degree of*

**Master of Science in Electrical and Computer Engineering**



Department of Electrical and Computer Engineering

Hammond, Indiana

December 2020

**THE PURDUE UNIVERSITY GRADUATE SCHOOL**  
**STATEMENT OF COMMITTEE APPROVAL**

**Dr. Donald Gray, Chair**

School of Engineering & Technology

**Dr. Chenn Zhou**

School of Engineering & Technology

**Dr. Xiaoli Yang**

School of Engineering & Technology

**Approved by:**

Dr. Xiaoli Yang

*Dedicated to my parents*

## **ACKNOWLEDGMENTS**

I wish to first thank my advisor, Dr. Donald Gray, for his scientific insights and guidance in this research. Besides inculcating technical expertise within me, he also took every opportunity to make me a well- rounded researcher. I have always marveled at his scientific and engineering expertise across various fields. I have learned work ethic, focus and dedication from the best. His passion for research and scientific truth will always inspire me.

I would like to acknowledge my other committee members, Dr. Chenn Zhou, and Dr. Xiaoli Yang, for their patience, inputs and feedbacks. I would like to thank the peers in my research group, for all the ideas and brainstorming with them.

In addition, I would also like to thank my mentor, John Moreland, for his valuable guidance throughout my studies. He provided me with the resources that I needed to choose the right direction and successfully complete my thesis.

Finally, I could not have completed this dissertation without the support of my parents for their wise counsel and sympathetic ear. They are always there for me.

# TABLE OF CONTENTS

LIST OF FIGURES .....	7
ABSTRACT.....	8
1. INTRODUCTION .....	9
1.1 Background to the topic .....	9
1.2 Virtual Training .....	10
1.2.1 Virtual Simulation .....	11
1.2.2 Interactive Environment .....	12
1.2.3 Advantages and Applications .....	12
1.3 Proposed Industrial Simulation Diagram.....	14
2. DEVELOPMENT OF INTERACTIVE FRAMEWORK .....	16
2.1 Framework and Implementation .....	16
2.2 Hardware and Software Tools .....	19
2.2.1 Hardware.....	19
2.2.2 360-degree Videos .....	20
2.2.3 Software .....	22
2.3 Simulator Software Development.....	23
2.3.1 Environmental Rendering and Shading .....	23
2.3.2 User Interface.....	24
2.3.3 Outline .....	25
2.3.4 OVRInput .....	27
2.3.5 Collision Detection .....	28
3. DEVELOPMENT OF ROUGHING MILL TRAINING SIMULATOR.....	30
3.1 Roughing Mill.....	30
3.2 Design Purpose .....	32
3.3 Roughing Mill Environment Creation .....	34
3.3.1 Data Collection .....	34
3.3.2 Modeling.....	34
3.3.3 360-degree Embedded Virtual View .....	38

3.4	Deformation and Curvature .....	40
3.5	Accessing Information by GUI Control.....	44
4.	RESULTS and DISCUSSION.....	45
4.1	Virtual Reality Platform.....	45
4.2	Feedback .....	46
4.3	Training Performance .....	46
4.4	Discussion .....	49
5.	CONCLUSIONS .....	52
	APPENDIX A. SCENARIOS TEMPLATE.....	54
	APPENDIX B. PESUDO CODE.....	55
	REFERENCES .....	59

## LIST OF FIGURES

Figure 1. Industrial Simulation Environment .....	15
Figure 2. The Frame Flow Chart.....	18
Figure 3. Oculus Go.....	19
Figure 4. Equirectangular Projection .....	21
Figure 5. The Shading Flow Chart.....	24
Figure 6. Unity UML Diagram .....	26
Figure 7.The Oculus Controller Virtual Map .....	27
Figure 8. The Roughing Mill Process .....	31
Figure 9. Roughing Mill Simulation Environment.....	32
Figure 10. Rolling Stands Models .....	35
Figure 11. The Slab Model with Heat Map .....	36
Figure 12. The Slab Segments Model.....	37
Figure 13. The Different Slab Models .....	38
Figure 14. 360-degree View .....	39
Figure 15. The Curved Slab Model with Gizmo.....	40
Figure 16. Bezier Curve.....	41
Figure 17. The Bent Slab Model.....	42
Figure 18. The Bent Slab Model Towards to Opposite Direction .....	43
Figure 19. The Monitors Tracking Slab.....	44
Figure 20. The Design Pattern .....	45
Figure 21. Slab Report UI.....	47
Figure 22. Main Menu UI .....	48

## **ABSTRACT**

The use of virtual reality technologies in education and training can provide an opportunity to reduce training costs while increasing the attractiveness and safety for trainees in the manufacturing industry. This thesis strives to show a virtual world which is based upon real world imagery can be embedded in a virtual reality environment and augmented with underlying real world mathematical models to show in real time the consequences of user selected parameters on a running manufacturing process.

An interactive virtual operator training environment is presented utilizing a VR headset in conjunction with a VR gaming engine which supports the headset and also embeds the behavior details of the underlying manufacturing process.

To demonstrate this approach an example is developed based on a steel industry roughing mill in which steel slabs are ultimately rolled into coils. This example shows the software application development process given as applied to an interactive training simulator intended for plant operator training which includes the use of the gaming engine Unity 3D, 360-degree video, 3Ds Max and several other software tools.

The resulting example is a controllable model of the steel plant roughing mill which can be used to enhanced the knowledge of plant operational personnel as to the effects various operating parameters have on the final rolled product while enhancing the understanding of this mill operation for new plant operators.



# **1. INTRODUCTION**

In recent times with the advancement of computer technology more viable opportunities exist for the creation of environments such as interactive virtual laboratory stands and training simulators. The use of virtual reality technologies in education and training can provide an opportunity to reduce training costs while increasing the attractiveness and safety for trainees in the manufacturing industry. In addition, the different uses of organic and virtual software development can enhance the educational training process. However, it should be noted that the interactive training software is generally a close approximation of an operational scenario and cannot be expected to completely replace an actual in person, on the job training scenario. In order to best convey necessary knowledge and abilities it is important that the users have the opportunity to interact with accurate representations of real objects. This thesis strives to show a virtual world which is based upon real world imagery can be embedded in a virtual reality environment and augmented with underlying real world mathematical models to show in real time the consequences of user selected parameters on a running manufacturing process.

## **1.1 Background to the topic**

Virtual reality (VR) is a rapidly evolving computer interface dedicated to fully immersing users in simulation, which can greatly strengthen the overall performance with strong impact and intuitive relationships between the computer and user interaction. As for the application usage, virtual reality has been successfully implemented some areas, such as rapid prototyping, manufacturing, scientific visualization, engineering and education.

More specifically, Virtual manufacturing (VM) is an emerging technology area that virtually creating environments refer to real manufacturing operations with a combination of knowledge and information-based models, simulations and possibly artificial intelligence. Many types of training activities can be integrated into one simulation-based training system by applying the use of the VM technologies. The result of this training can result in lower expenses and rapid development of scenarios not possible on the manufacturing floor. The technologies of Virtual Manufacturing and training techniques associated with manufacturing processing will be

reviewed. Practical applications of Virtual Manufacturing in the area of materials processing are found in a paper by Lee [1].

Virtual manufacturing systems can be defined as integrated computer models providing the functionalities of representing virtually the overall structure of a manufacturing process while mimicking their physical behavior simulation while in operation. Iwata [2] developed and defined a manufacturing system architecture with key functional components being scene modeling, process emulation / simulation and interface structure tying together the VR world and the associated system framework. This research focuses on the modeling and simulation architecture needed for the practical development of virtual manufacturing systems. A second approach to virtual manufacturing is found in Fumihiko's [3] paper. This approach, which emphasizes the modeling necessary for a given product and the associated manufacturing processes behavior to be replicated by computer as precisely as possible. Also discussed are the ability to make prediction for potential problems which may happen in product functional and manufacturing process before implementing the real one.

## **1.2 Virtual Training**

Virtual training is a broad area where the emulation of a real task is accomplished in a virtual environment, which could include haptic feedback or another non-force feedback mechanism. This can produce an effective tool for coaching operators through process tasks that may be too costly or risky to replicate in a real-world training environment. The time, expense and resources related to the development of process control skills in operators and maintenance personnel represent substantially an investment. As mentioned in the paper of Transfer of Training from Virtual Reality by Kozak [4], the important properties of virtual reality thought needed for trainees to be truly immersed in a 3D world providing a new approach to operator training. Despite environmental and task similarity, not all VR training environments are appropriate for motor skill task development required in the real world. One approach to enhance the operator training effectiveness is to increase the correspondence between the real-world task and the corresponding virtual training context. Lastly, it is noted that computer processing speed has a bearing on the operations such as rendering which reduces the responsiveness of the overall VR interface.

Virtual training has been shown in many cases to reduce operator error and increase speed and quality of task performance as compared to traditional training. Some of the training systems examined depict process training with either virtual reality or augmented reality being employed [5]. It is believed that what is needed is an improved approach to capture and represent the real world, in essence duplicating in virtual reality the exact real-world environment [6]. The ultimate goal of virtual training is to enhance the acquisition of knowledge obtained in a virtual environment and transfer that to the real-world environment.

### **1.2.1 Virtual Simulation**

Virtual simulation plays a part of role in the virtual training as the beginning step where virtual reality has an immersive effect between users and simulation environment [7]. In Zhong's [8] paper, a methodology is presented to create a 3D visualization of discrete event system. The connection between discrete event system and the 3D animation might be considered a novel approach as it converts discrete events into associated animation events used in the 3D animation. Another paper by Sanker [9] describes an experimentally proposed sorting station as the basis for a case study model of a real production system. By utilizing augmented virtual reality, this approach helps increase the execution speed of the simulated process in real time. This particular case examines a virtual product moving down a belt conveyor prior to the testing of the parts. Abidi [10] introduces a new simulation method that presents easily understood results of a simulation which shows the reader with the advantages of virtual reality usage in manufacturing processes type simulation. This is crucial importance because the users in manufacturing system usually are not familiar with the mathematical concepts used in discrete event process simulation. Cheung's [11] paper describes a simulation model used for the surface topography in a high precision diamond turning process where the kinematics of the turning process involving machining parameters and tool geometry are considered illustrating the relative motion between the process and the diamond. This is useful information regarding potential process modeling.

A strength of visualization is that it can become a critical component of simulation technology to enhance communication's results and allow trainees to get a better understanding of observing a process's behavior. Rohrer's [12] model uses 2D and 3D animation, model building time and debugging have been significantly improved. This paper discussed the elements of visualization to support simulation practitioners realize where to use animation and how to

improve the simulation modeling process. Winter's [13] team evaluated the suitability of the virtual reality environment for software requirements verification. Although higher costs are introduced during the formalization of requirements, this approach can be very helpful in developing high-result systems.

### **1.2.2 Interactive Environment**

Since virtual reality requires the use of both computer software and hardware to simulate a virtual environment for users a survey of current environments is in order. These can include flatscreen monitor projecting 3D images, "cave" environment as well as 3D headsets. The possible application areas of virtual training can be for safety training. Ultimate success would be seen in trainee's identifying and assessing construction safety risks. Moohyun's [14] research in real visualization and experience data to help further improvement of the effectiveness of firefighting training and enhance useful knowledge for less experienced users. Their research resulted in a real-time VR simulator utilizing a highly accuracy CFD simulation. This application included offline data conversion as well as real-time processing logic designed to handle a large amount of (CFD) data. A reduced real-time system has quantities that are pre-calculated based on the CFD data. Tesic's [15] research methodology is most suitable for interactive process and simulation applications that require high-precision object communicate with modeling. Examples presented were virtual assembly, mobile robot and manufacturing processes simulation that require a modeling to approach to embed error detection in those operations.

### **1.2.3 Advantages and Applications**

Virtual reality can be an effective tool for analyzing the flow and testing of new products, shortening development time and potentially reducing financial costs. Virtual reality can be considered as a natural extension of 3D computer graphics combined with advanced input and output hardware devices. This technology has recently evolved enough give it serious consideration for engineering applications. One aspect of virtual reality that may seriously affect design and manufacturing is assembly design. A paper proposes research work aimed at creating a virtual assembly design environment [16].

Different applications for virtual training include manual assembly tasks. Virtual training for manual assembly uses technology to improve worker performance in assembling mechanical components. In Gorecky [17] et.al., the author evaluated a virtual training system for manufacturing and found that its improved knowledge-delivery and skill transfer. Brough's [18] system allowed supervisors to create customized training and found that allowing users to change preferences helped training outcomes. The article is explaining a new approach for free-form shape design using a data glove [19].

Visual simulation of construction work is an effective means to convey the logic and internal work of the simulation model in a comprehensive and understandable way. This can use model verification and validation and support establish the credibility of simulation analysis. The inherent working nature of discrete event systems and the visualization operations required in a 3D virtual world faces many interesting challenges [20]. Liang [21] introduces a VR platform involving the use of a dynamic multi-agent programming language is introduced. The purpose of designing these tools is to show that multi-agent systems can be simulated in a virtual environment under dynamic properties for interactive prototyping situation.

By Masahiko's [22] definition, virtual manufacturing is the concept of performing manufacturing processes on computers and the real world. It becomes possible to estimate the manufacturing process in advance without using actual equipment, and hence, VM is expected to be implemented in many manufacturing applications. Virtual reality applications for use in manufacturing can include the operations of design, prototyping, machining, assembly, as well as the inspection, planning and training of operators. Current VR technology has been applied to real-world problems of customer experience, profitability, time to market and worker safety [23]. In Adams's [24] research, full use is made of advanced computer graphics, some systems containing audio feedback, but none provide mandatory prompt feedback to users.

In Langley's [25] paper, in order to be fully defined as an effective training tool in the automotive industry, evidence is needed to prove that these systems can provide effective and effective training for relevant users. The ability to quantify training success is very important and should be part of any training system. Nirit's [26] assumption that demanding tasks will require higher-level solutions such as strategic planning. VR and AR technology will focus on enhancing the awareness of these tasks, so it will have significant advantages compared to traditional training. As users become more exposed to VR and AR platforms, training over the years should become

more efficient. In this regard, the use of the AR platform is more intuitive, so the VR platform is expected to benefit more from the initial contact. As the literature research proves, the new platform for training can increase the proficiency of operators and reduce the error-prone performance in the real world, so the investment in the development of VR and AR training systems for this purpose is worthwhile.

In the past two decades, many attempts have been made to improve the training process by using virtual reality, but the use of these methods in industry is still not well accepted and not much documentation exists in the literature. Kamat [27] targets using case study methods to assess the actual benefits of virtual strategies employed in the oil and gas industry. In China, production methods have been continuously exploring in real-world production, constantly discovering and solving problems, fewer products but high manufacturing costs. The solution of the above problem is that the application of virtual technology in the simulation of manufacturing process of high-speed trains [28]. A real-time simulation graphic engine focusing on equipment manufacturing, can provide comprehensive technology support for industrial 3-dimensional data, accelerate applications of VR to all aspects of industry, cut down development costs of equipment manufacturing significantly, highly increase efficiency of equipment design, and promote development of equipment manufacturing [29]. In Trika's [30] paper, a computer aided design (cad) system is realized through the VR interface, which make the complicated design of mechanical parts easier.

### **1.3 Proposed Industrial Simulation Diagram**

Based on these above references, a virtual process training environment should contain three key features: appropriate functionality, effective human interaction, and operate as a real-time environment. Nowadays, VR technology has been applied to machinery simulations to ensure the proper operation and performance of equipment prototypes while it can be considered in the early stages of design. In order to understand the coupling of the process and verify the timing and process of equipment operation and the quality and performance of process equipment, this thesis proposes a VR framework integrating the modeling of the manufacturing environment along with the simulation of the manufacturing process in the VR world [31]. The manufacturing industry uses VR tools to extend product life cycle. Some other applications have been mechanical studies, assembly and maintenance training [32].

The use of Industrial Simulation Environment flow chart proposed in Figure 1 shows that it is a powerful methodology that can support the ability for processing methods of re-engineering the loop in order to achieve an enhanced performance when viewed from the perspective of the “user” in the loop. Building this proposed flow chart will require the use of user-based sources, model-based sources, and environmental-knowledge-based sources. The flowchart also depicts both input and output interfaces working simultaneously, a GUI based process environment and the underlying process model (Figure 1).

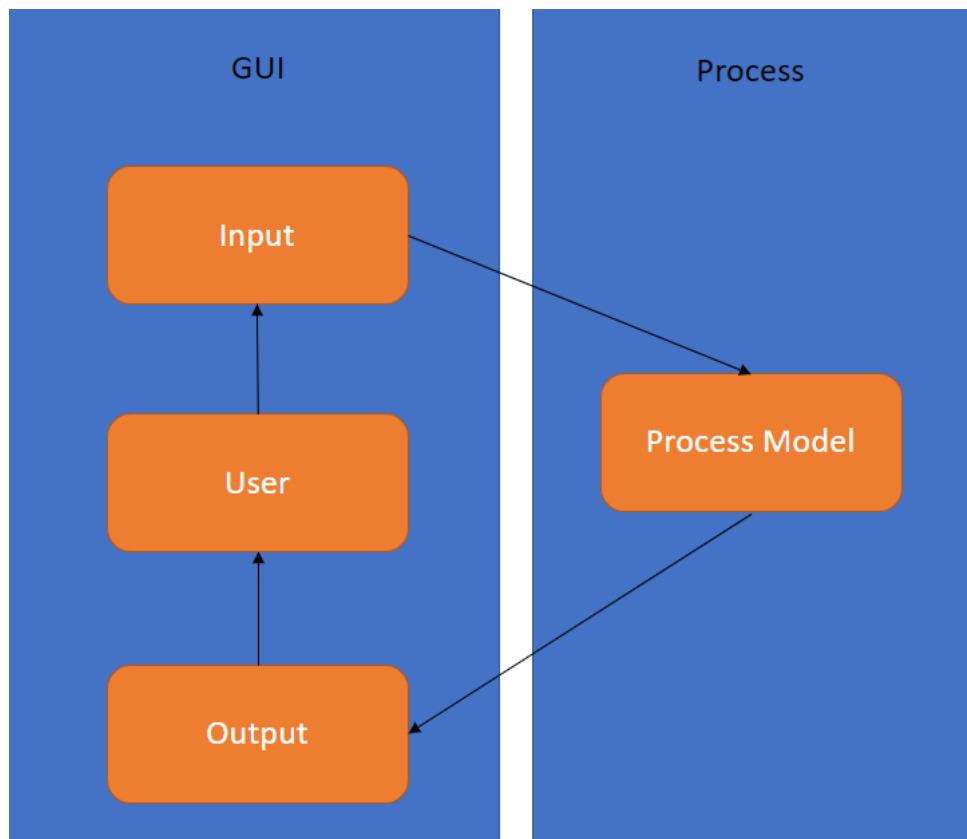


Figure 1. Industrial Simulation Environment

## **2. DEVELOPMENT OF INTERACTIVE FRAMEWORK**

This chapter intends to convey the technical issues encountered in creating an interactive operator training environment. It was decided that the most appropriate VR environment for this application was a VR headset. Next, a VR engine needed to be selected to support the headset and Unity 3D was chosen. To embed the intended system for training a modeling process module was developed and incorporated into a game engine. The game engine selected was again the Unity 3D and was used to create the virtual scenes. The virtual scenes are to be created by first-obtaining images of the intended operating environment which will then be stitched together to form the virtual 3D environment for training. Lastly, user input objects such as button and parameter knob images will be enabled in the UI system for users.

### **2.1 Framework and Implementation**

Virtual reality is the technology being immersed in an interactive three-dimensional computer-aided environment, where virtual models own a sense of space, which is called interactive three-dimensional surroundings. Three-dimensional computer-generated virtual environment. Through immersion, the user's observation view or an interaction part of the user's body (for example, hands) can be contained in a computer-generated space. Through existence, computer-aided models in the virtual environment have an obvious position in the three-dimensional space associated with the user.

Unity is a popular game engine with more than 700 million users. It is mainly being implemented with developing video games software for many devices such as mobile devices, Internet browsers, PC or Mac. In addition to creating games, Unity has recently been used to create videos, teaching materials, and neuroscientist applications. In addition to being used for traditional screen-based video game design. In addition, it uses C# as a dynamic scripting language in Visual Studio to control its various scripts and functions libraries. There are three scripting languages supported by Unity 3D: JavaScript, C# and the Python dialect called Boo. All of them are fairly smart and interoperable, and also support underlying .NET libraries with databases, regular expressions, XML, file access, and networking. Scripts are generally considered having limitation and running not fast. While in Unity3D compiling environment, native code are used to run the



scripts and run them almost in the same speed as C++. For those who like scripting languages, it is easy to get fast iteration time and easy to use.

To build a completely integrated training software, the development process will be divided into two parts. Process modeling, which is outside unity, will utilized 3Ds Max. Scripting for UI panel interactions and functionalities mainly done inside Unity 3D. The virtual environment and its interface should be suitable for actual operator tasks, which will be the main work within the unity. Strict performance constraints must be met to realize the benefits of virtual reality and conduct feedback analysis on it through its own development. The simulator software developed in the game engine is a key part of the whole framework.

In Figure 2, the starting from obtaining 360-degree video and all the material information has two branches connected with. A linear “waterfall” design process on the Figure 2 right branch has explained the process happened in Unity 3D while the right branch mainly happened in Solid works and 3Ds max. The material has been processed in different types files such as. FBX. The attaching information and data transmission will give the texture when files exported and import to the Unity 3D where the two branches will meet. Also, there will be a closed loop design for the processing optimization in Unity 3D.

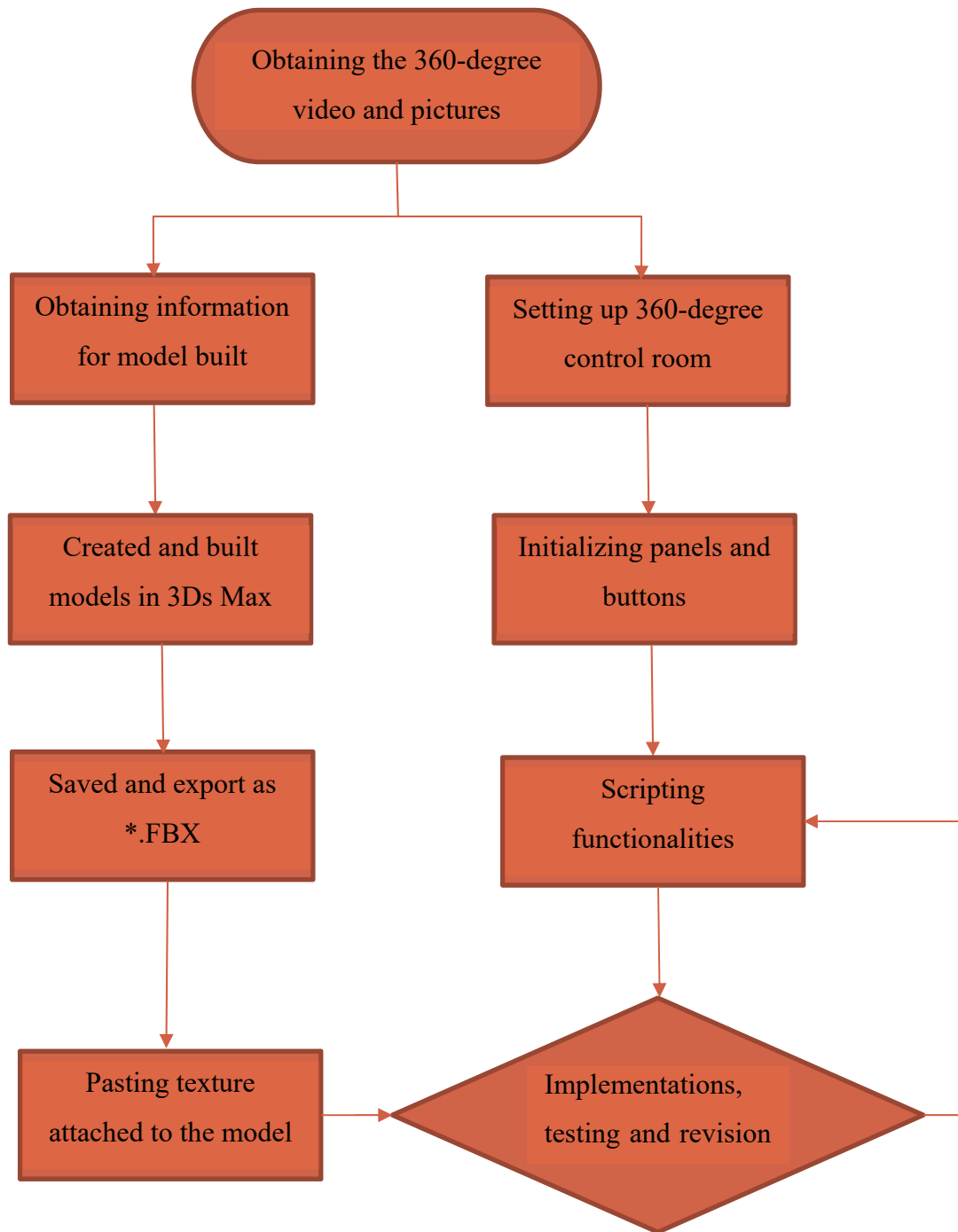


Figure 2. The Frame Flow Chart

## 2.2 Hardware and Software Tools

### 2.2.1 Hardware

The largest consideration for this training environment is that the VR user experience should be as easier as possible. Users must be easily able to master the equipment and simplify the software usage for them. Oculus Go was selected for the VR training. By implementing the inclusion of the Oculus library in Unity, users are able to observe a virtual feeling and sensing of their heads in the center of their world views. This is helpful for training experiments and can also be useful for future development. When wearing the Oculus Go, the users can move their heads rather moving their hands. The users will only see the smaller equirectangular view based on the cursor center location instead of the complete 360-degree view. The centerline view of the Oculus wearer also serves as the user's pointing functionality similar to that of a mouse or cursor with two handset buttons used for selection and confirmation.



Figure 3. Oculus Go

As for the oculus controller, since a specific control scheme will only be used as a specific controller type, a designated hardware device can be used. The controller buttons can correspond to the most updated user input through interaction. While the user can use a single one Oculus

Touch controllers, do the action such as put it down, and then lift up the controller. In this situation, once the user provides input, the Active controller will switch to the controller (Figure 3).

The users can see objects in the virtual reality system from any angle. As a respect of the adjustment of the control center, it can accomplish a real-time location movement. As for the user-center effects, some mirror reflection effects can be also detected. Except of the items mentioned above, users can also watch videos, listening horn sounds or instructions in the system, they can rotate around on the room floor or guided by tutorials' sounds. Therefore, the manufacturing training process in the modeling environment can be more mimic. This process model will also provide more functions of heat transform evaluation, force detection, alarm system information and so on, which are similar to 3D game tutorial.

### **2.2.2 360-degree Videos**

As an important component of the virtual reality technology that was utilized, 360-degree video provides users with a panoramic view of the complete environment and allows them to control their viewing through free direction while the video playback. The output device, in this the Oculus, often displays only the visible portion of a 360-degree video [33].

As in Figure 4, the spherical viewing space of omnidirectional video is showed on the left side while the Equirectangular projection (ERP) plane of the video is showed in different forms on the right side. (a) illustrates that the overall 360-degree visual view at a single time frame in time can only be seen by the observer partially. Figure 4(b) shows that the corresponding viewport of the ERP plane. This is a projection type used to map a part of the surface of a sphere to a flat image. Because the horizontal coordinate is only longitude and the vertical coordinate is only longitude, any transformation or scaling cannot be applied, so it is also called "non-projection". Cameras can sometimes scan and cover rotations of over 360 degrees, but the software usually assumes that the equiangular rectangular image covers 360 degrees horizontally and 180 degrees vertically. Isometric projection, as the default output format of a rotating (scanning) panoramic camera by using a fisheye lens-a 180° fisheye lens, can provide a complete sphere when rotated by 360° [34].

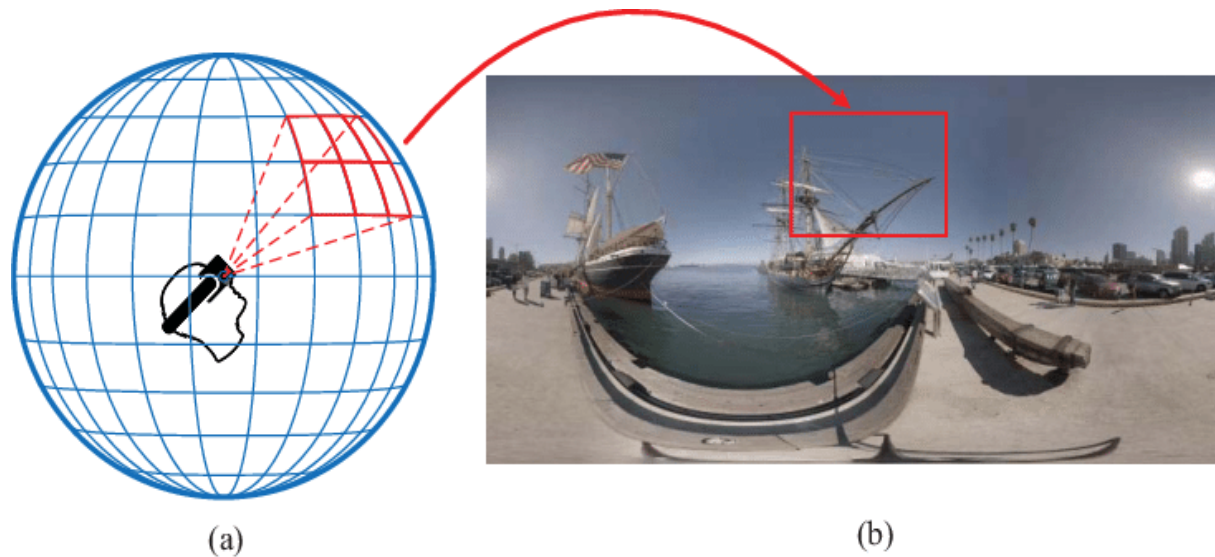


Figure 4. Equirectangular Projection

For the 360-degree video, the texture was processed with Photoshop for cutting several observation pieces which will then be added to the modeling and visualization parts of the simulation. Instead of creating all modeling environment, the real capture 360-degree pictures and the 360-degree video are overlapping each other. From the user experience side, this technique relies on the theory of Binocular vision. To achieve that is to use a texture mask with 'trick-effect' image and a parameter called alpha settled to 0 to show the transparency where world should be visible.

Then, this texture is attached to material with transparent-cutout shader and place material to plane and the plane should be placed in front of camera.

VR panorama is a static picture or dynamic picture collected by a 360° panoramic image collector. Each picture contains 360° azimuth environmental information. After wearing VR glasses, the viewer's body rotates the angle of view, thus bringing the immersion the feeling of the environment. VR panoramic shooting is shooting in 360° lens mode, which can restore the real space 1:1, and the immersive scene-oriented perspective can improve the user experience. VR panoramic shooting can choose different equipment according to different shooting needs, from lightweight Insta 360 panoramic camera, GoPro, to professional-grade SLR camera, the operation difficulty is also different. Novices with zero basics can start with a panoramic camera, and then improve step by step to master more complex equipment.

### 2.2.3 Software

By using "build settings", Unity3D provides seven formats for output production: Web Player, Web Player Stream, OS X dashboard widgets, Windows independent, OS X universal binary independent, OS X PPC independent, and OS X Intel independent. If the "build settings" is selected, an executable file will appear which can be stored in the user's computer. Therefore, the virtual reality system can be applied with various purposes and even many operating processes. The project generates the system through the "build settings" in the file. Other types of systems can also be generated, such as Web Player. After the files are modified, their status will be updated immediately. The operation of assets moved and renamed is very elegant. For refreshing the system, submissions and graphical models' comparison are all showed in the Unity editor [35].

Virtual reality systems can be highly vivid and interactive. Through the comprehensive personalized operation mode, users can select the way to browse and enter in software environment according to their own wishes, and give full play to their imagination, without having to use the designated keys on the keyboard to influence others. The computer rendered scene is continuously updated at 60 times per second, and the user will be able to make inputs naturally and immediately immersing the user in the virtual scene to explore and observe spontaneously. It will be shown that Unity 3D can be used not only for game development, but also for processing information about the operation and performing some simple data analysis.

Visual Studio brings powerful features to C# programmers. Use IntelliSense to write code quickly and accurately. Visual Studio can bring a high-quality debugging experience to the Unity game engine, quickly identifying problems by debugging Unity game in Visual Studio-set break points and evaluating variables and complex expressions. In addition, using the implementation of MonoBehaviours and fast MonoBehaviours wizards, users can quickly create Unity script methods in Visual Studio. Projects can be browsed in the same way as using Unity Project Explorer in Unity.

3Ds Max provides animators, modelers, and visual effects artists with a variety of tools to create eye-catching effects, realistic 3D characters, and stunning environments for movies, TV, and games. 3Ds Max 3D modeling and rendering software allows users to create magnificent worlds in games, stunning scenes in design visualizations, and fascinating virtual reality (VR) experiences.

Blender, as an open source cross-platform comprehensive 3D animation production software, provides a series of animation short film production solutions, from modeling, animation,

material, rendering to audio processing and video editing. Blender has a variety of user interfaces that can be easily used in different work environments through built-in advanced movie and TV solutions, such as green screen keying, camera back tracking.

SolidWorks is a set of mechanical design automation software, using the familiar Windows graphical interface, easy to operate, easy to learn and use, and is widely used in the fields of machinery, automobiles and aviation. There are three major modules in SolidWorks 2014, namely parts, assembly and engineering drawings. The "parts" module also includes small modules such as sketch design, part design, surface design, sheet metal design and mold.

## **2.3 Simulator Software Development**

### **2.3.1 Environmental Rendering and Shading**

Following the modeling creation, the shading flow happen after model being exported from 3Ds max. Depending on which type of data is running in Solidworks, models may attach with materials information as initial status or may only have the polygon which will need a material later in Unity. For adding more function to meet the different needs in the modeling process, a definition called shader support the scripting and editing ability of the 3D models.

The material of each model defines how light reacts to the surface of the model and is the basic element for making realistic visual effects. Shaders are programs designed in detail to run on the GPU. In the end, it is the reason to draw the triangles of the 3D model. When the model is being created, using a shader to define its surface properties. The Unity Standard Shader is a built-in shader with a comprehensive set of features, which allow users to create plausible materials easily, and contains advanced lighting models called Physically Based Shading (PBS) (Figure 5).

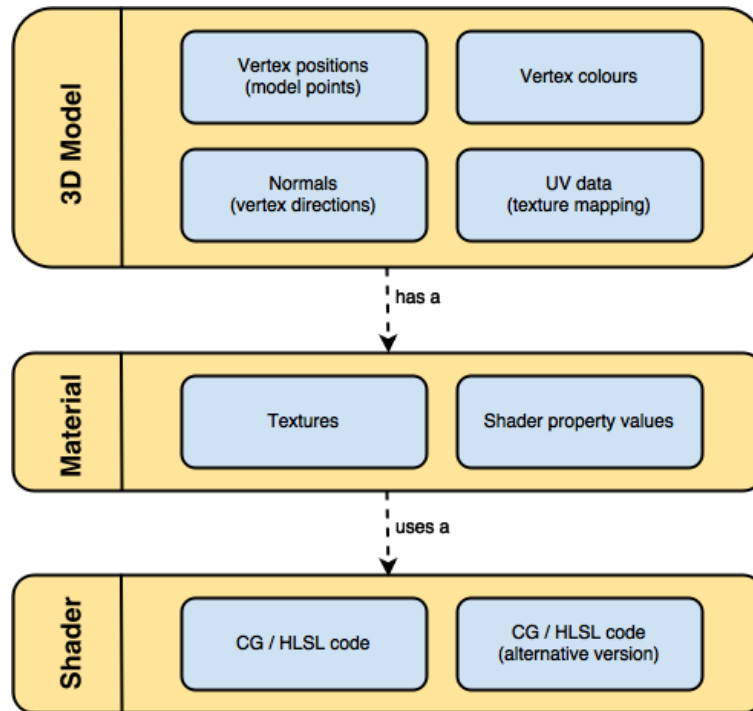


Figure 5. The Shading Flow Chart

A detail texture is a pattern that is faded in gradually on a mesh as the camera gets close. This can be used to simulate dirt, weathering or other similar detail on a surface without adding to rendering overhead when the camera is too far away to see the difference. Unity import settings for the detail texture. Once the image has been saved to the Unity project, it can be selected to see its import settings in the Inspector. The aim of using this type of textures is for extra realism close-up [36].

### 2.3.2 User Interface

The user interface (UI), as the part of human-computer interaction (HCI), and provides data transmission in software and hardware, displays panels, buttons and monitor screens, which can be considered as a common technology used in recent years. In the real life, the buttons and panels from the control table can be considered as an origin from user experience. The device appearance, content presented and time for user to response are the main elements in the user interface [37].



These above terms all explain the concept of human-computer interaction which can be considered as the field of study focusing on the innovation of computer technology and the interaction among users and all types of technology design. HCI studies areas including UCD (User Centered Design), UI design and UX design are examples of this area. There were few user interfaces except for a few buttons on the operator console in early times. However, the Graphical User Interface (GUI) appeared as an integral component of contemporary computer software. Windows, buttons, scroll bars, pull-down menus and icons are some elements in GUI. After a short period, the increasing application needs of multimedia leads to the GUI. Sound, voice, motion video and virtual reality are rapidly applied to the GUI for various areas [38].

### **2.3.3 Outline**

In this study, creating a training simulator to mimic a manufacturing process control is the goal. Hence, the model is the initial game object should be dealing with. The communication from UI panels and UI buttons to the model process plays the important role during the whole programming process. The application programming interface (API) and/or language has a highly standard in design. Hiding the underlying complexity of the system should be also important in the design view. So, C # as the main scripting language and Visual Studio as the editing platform communicate between unity and programming logic.

In order to accomplish some specific functions in the scenes, the functions can be applied by the programs in the scripting platform of Unity3D. Among many options for editing software, Visual Studio can edit the scenes in Unity3D and can also compiled by adding the UnityEngine.dll.

On the one hand, the development system should have the user-friendly configurations and be able to adapt to many hardware configurations. As for the programmer side, the proposed design patterns and the whole scripts structure should be dynamic. In different situations, the structure only needs to be changed by API names or any details information.

On the other hand, the functionality requirements support the logic of different tasks. And the connection and relationship between these tasks guide the scripting. Panels for users to click buttons need to be separate scripts control while the functions will be easy to realize. Some general functions can be stated as changing model size through different dimensions. The model changes also happen differently and simultaneously. In some cases, the changes are multitasking because

not only one factor is impacting on the model. The characteristics of the model should be conditional update based on user's inputs.

Starting process of simulator begins with the menu associated with the main scene. Main menu separates tutorial, about and practice mode while each scenario to perform tasks individually and allows users to check results or restart the practice in the parallel process. The main menu should be made convenient and clear. All the important reference material for learning the process, equipment, and knowledgebase technology of manufacturing operation is supported by the simulation system systematically for get familiar of working process for new operators as training process.

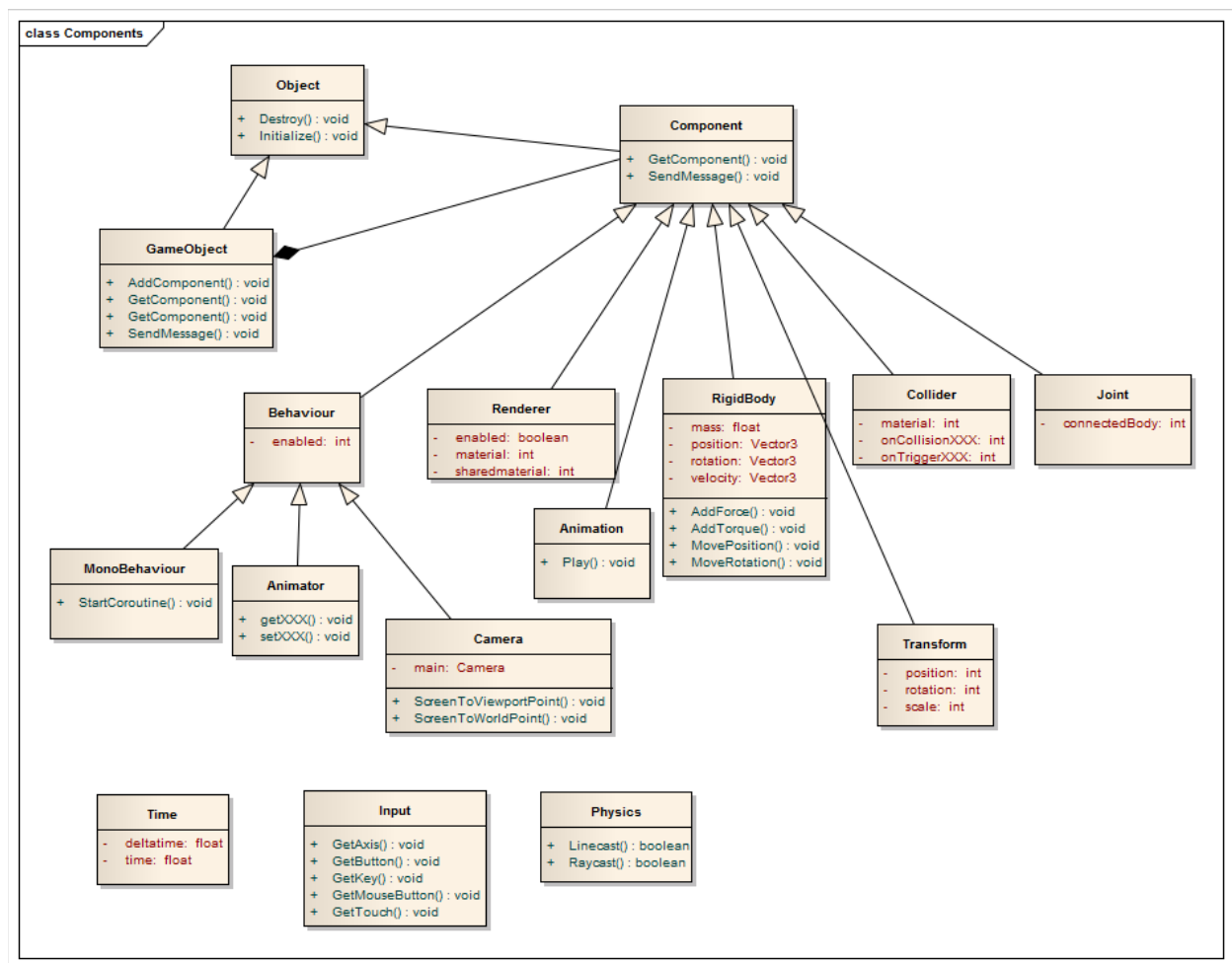


Figure 6. Unity UML Diagram

The mechanics module was developed into game engine Unity 3D to create the virtual scenes. There are some components (Figure 6) used as each steps of building environment, such as GameObjects and objects. MonoBehaviour will be the mainly used in the unity project with modeling renderer. Rigidbody attached to the GameObjects, Collider, Joint and Transform are components also utilized for various functionalities. Some scripts were developed to control the camera and the other scripts control operation parameters of the manufacturing. Each script can be considered as a code block with an executable individual file and perform a unique task, such as the logic control for a rotation or a movement of the Game Objects [39].

### 2.3.4 OVRInput

It has been specified in this thesis that the user is required to wear a VR headset. Since the hardware is chosen as Oculus go, OVRInput package is applied in the unity scene files. OVRInput is used to expose a unified input API for multiple controller types, such as buttons, thumbsticks, triggers, and accumulated recorded touch data which also support the Oculus Go and Oculus Go controllers. Ads for using keyboard and mouse on the PC, it is recommended to use the UnityEngine.Input scripting API [40].

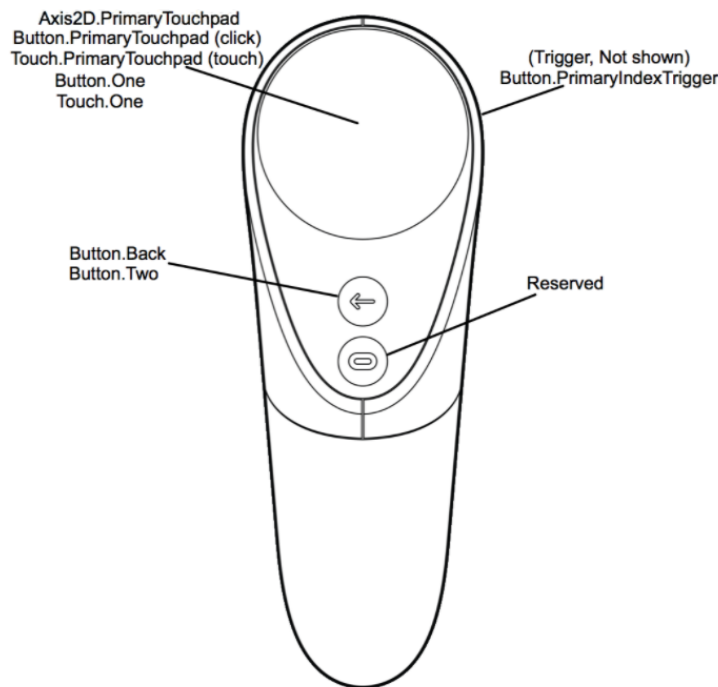


Figure 7. The Oculus Controller Virtual Map

In addition to the traditional gamepad buttons, the Oculus Touch controller also has a capacitive-sensitive control surface that detects. In some situations, the user's finger or thumb makes physical contact (touch) or they are contacting closely (NearTouch) or the user's finger approaches the control and approaches the control, "NearTouch", as a Boolean variable, will report as true before the user makes physical interaction [41]. When the user interacts with environment, the touch of the control will be reported as true. When the users press the controller's index trigger, the button of the control will reflect in true status. To correctly track user contact with the controller and ensure various control schemes, these different states can show information by passing variable value. In this way, several different states of the user's interaction with a specific control interface can be detected. Similarly, if the user using index finger is completely losing contact from the control surface, "NearTouch" of the control can be reported as false (Figure 7).

### **2.3.5 Collision Detection**

Collision detection was applied in this research to mimic the real operation process. To make sure the accuracy of the simulation, some packages from Unity 3D are used to establish the real scene. This is the original idea of the first approach with the previous model. Simulating the operation process through virtual system need following the specific working regulations and operation process of the specific equipment. The designed manufacturing operation control system did not first take factors of gravity and friction impacts in consideration for the first design model. Hence, the simulation process causes some defects, which are hard to deal with by only editing the scripts.

However, as moving forward to the third approach that instead of using manually scripting control the simulation process, there is a better choice and tool that can be used to mimic the manufacturing operation process in the real world, especially for the side guides' and edger's collision detection.

The Unity 3D / NVIDIA PhysX physics engine is a popular physics engine used by Unity developers. This engine supports the physical collision by using rigid body and simulates the effects from gravity in scenes. Users can gain an immersive view and sensing in this type of virtual environment through physical engines.

There are five basic colliders can be applied to the model. Among them, box collider is mainly being used for collisions between cubes. While there also others like Sphere, Capsule,

Mesh and Wheel. Two types of collision detection exist in Unity3D. One is rigid body collision and the other is a ray collision-based method. In order to detect the stage of collision and occurrence, Rigid body collisions are able to perform `OnCollisionEnter`, `OnCollisionExit`, and `OnCollisionStay` functions. In Unity 3D, there three types of colliders: Static collider, Rigid body collider, Kinematic rigid body collider.

After knowing which of collision engine to use, there are two main methods to detect collisions for options: use a collider or an event trigger. In the collider process, the trigger is separated into three modules for collision detection. They are an entry trigger, an exit trigger and the stay trigger. When two triggers are generated the same time, the `OnTriggerEnter` function will be called. When the two models stay on the trigger, the `OnTriggerStay` function will be invoked, and when they leave the `OnTriggerOut` function will be called [42].

### **3. DEVELOPMENT OF ROUGHING MILL TRAINING SIMULATOR**

As an example of what is possible with this approach an interactive virtual simulator entitled "Roughing Mill Simulator" has been designed and realized by using Unity3D, SolidWorks, Blender and 3Ds Max.

#### **3.1 Roughing Mill**

The interactive virtual simulator "Roughing Mill Simulator" presents a scenario in which a new operator could be trained to work a roughing mill process while positioned in a control room. The roughing mill usually are made of several roughing stands. While the slab is passing through these rolling stand, it is repeated hot rolling five to seven times to achieve the optimized thickness requirement. The roughing mill also consists of edger rolls to functioned and roll slab's edge and center the location. The precise control of the roll gap can better control the inventory in each pass and avoid the excessive load on the roll and the drive system due to the incorrect roll gap. Faster and more precise gap settings can perform better and increase mill output.

A programmable logic controller (PLC)-based automation system is managing all these functions of Hardware Security Module (HSM) [43]. There are three levels of automation to accomplish precise control of roll gaps. It is possible to better control the inventory in each pass and avoid excessive load on the roll and drive system due to incorrect roll gap. Faster and precision gap setting, screws, side guides and force meters settings can be desired as it increases mill output (Figure 8).

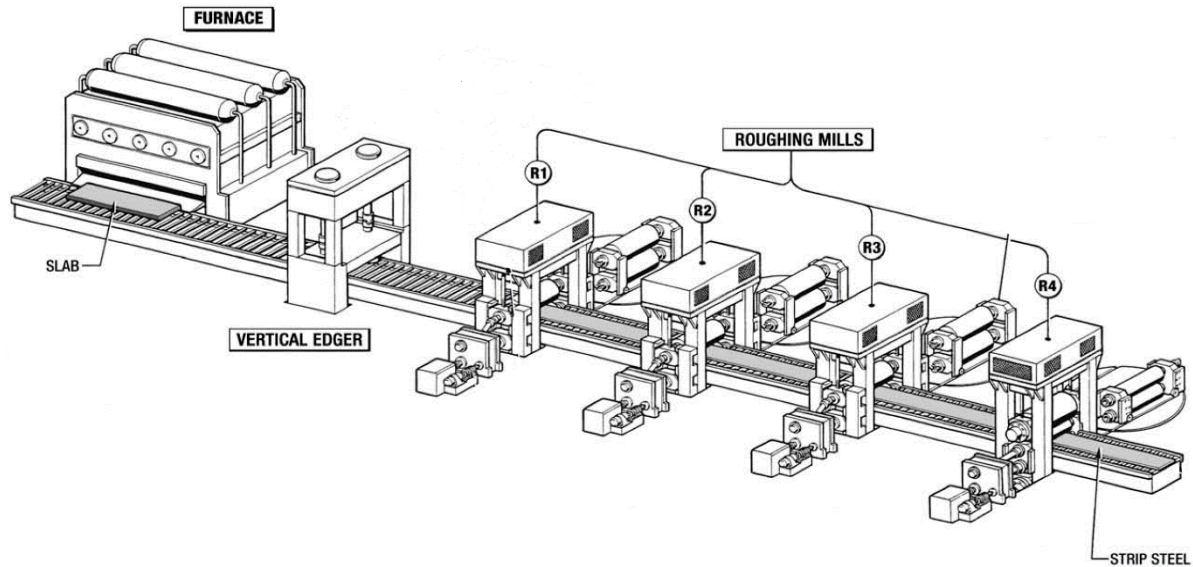


Figure 8. The Roughing Mill Process

Five major parts in the hot strip rolling process are introduced below: Reheating, Rough rolling, Finish rolling, Cooling, Coiling [44]. The automatic instrument control (AGC) system supports the returning signal of the instrument as output to give the feedback control and evaluate the exit thickness of the strip [45].

A physical machine was designed to train students in troubleshooting for automotive engineering. In order to provide an innovative solution for optimizing learning effectiveness, Center for Innovation through Visualization and Simulation (CIVS) is helping to build a virtual simulator for this machine so that users can operate it online and provide convenience and safety. The simulator includes the machine itself, a main cabinet and a PLC interface which has been fully modeled in 3D space. Different troubleshooting scenarios were added and Logic for the machine was added based on the behavior of the real one. Next, users can use the human machine interface to operate the machine provided with tools for troubleshooting and protection. The outcome is a web-based 3D Virtual Simulator providing students with unlimited opportunities for practice and supports instructional delivery curriculum to improve educational outcomes based on the physical Instructional Manufacturing System Simulator, with fault-based scenarios [46].

### 3.2 Design Purpose

The goal of the example project is to assist the collaboration between PNW CIVS and a steel mill in the midwest to develop a training software that both incorporated virtual hands-on training for roughing mill operations and visualization of previous CFD results to broaden understanding of physical phenomena that happen during the roughing mill processes for process operators.

The technology used in the process is aimed at improving understanding of roughing mill operations and enhancing the understanding of how new grades of steel might impact mill performance. As a research tool for a steel mill in the midwest for training roughing mill operators and providing better understanding of how to deal with abnormal slabs conditions, this software aims at reducing fatalities and injuries by developing extensive safety training software to improve the effectiveness of safety training and employee training engagement. Also, this software made an Interactive 3D environment based on real incidents in multiple technology platforms which allows employees to experience virtual hazardous environment.

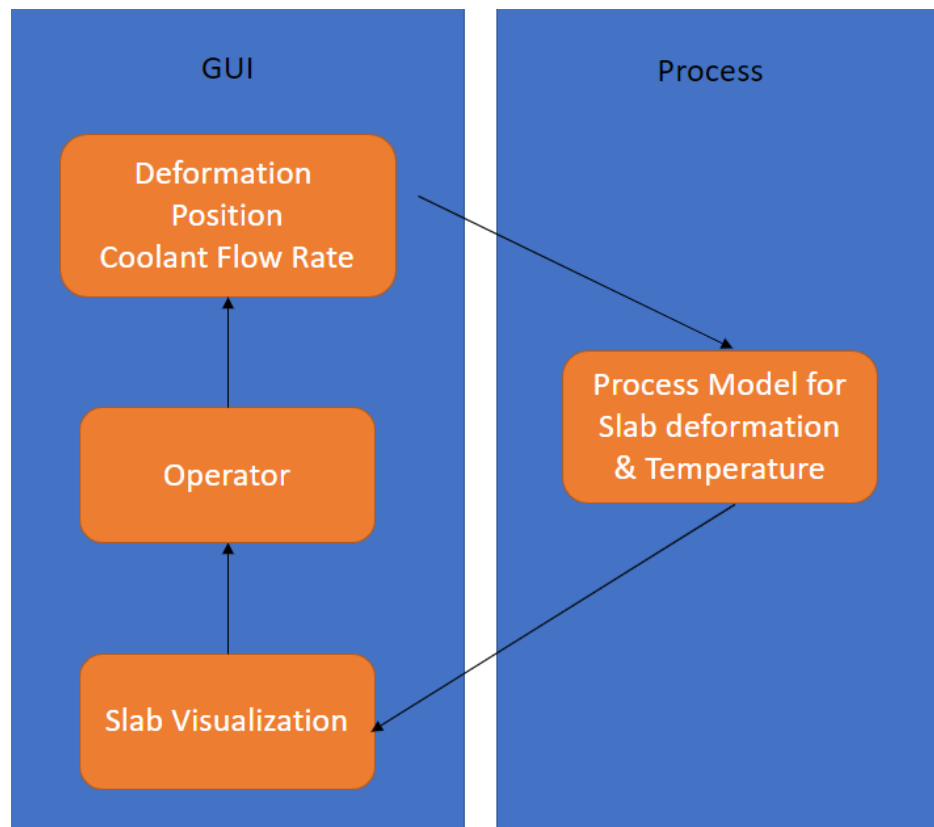


Figure 9. Roughing Mill Simulation Environment



Visual implementations are taken into steps to enhance understanding of mill operation for new operators. Based on the demands of training software usage, the user interface to accomplish the interaction are designed as user-friendly and easy used by new users. In order to accomplish the goal of designing a controllable model of the steel manufacture roughing mill and enhance knowledge of effects various operating parameters have on the slab hot rolling. The basic step will start developing a self-contained software application for interactive roughing mill training simulator for use on PC which can be also considered as a VR software of the steel manufacture roughing mill.

The cooling process is simulated when the slab is stuck by the rolling stands and having trouble to go through. Sprays continue working on cooling slab till the temperature of the slab go down very far and the color of the slab surface become darken. While the training model will be designed as basic control panels, changing the slab deformation in different dimensions when the user side inputs differently. Slab's physical performance in temperature changing and deformation will be main model reflections. In order to track the camera, control operation process of the slab, and to observe the roughing mill control process, modification of scripts is necessary. Each script in an individual file can be considered as a code block. Each block performs several identical tasks, such as changing the rotation and movement of the slab. The above designed graph allows to control the correctness of the mill process.

Process knowledge plays as an important role in considering for industries similar like the flow chart used in this study. Model process knowledge refers to process route information of a model, which may change according to the input manufacturing data from operator control. In Roughing Mill Simulation Environment (Figure 9), GUI side is dealing with the process knowledge controlled by operator. The deformation position and coolant flow rate of slab will be the input parameters passed through the Process side, where the model of slab deformation and temperature will be updated by simulation. The simulation changes will be transferred back to GUI side for slab visualization as output. Once the operator saw the result of slab through monitors, the process knowledge following the rules will be processed by operators. The operation manuals will guide the operator to behave in the right direction and the close loop stay in good status while it may have some possibilities for miscellaneous mistakes result in loop cyclic interrupts.

This training software aims at gaining knowledge-based background control of the key workflow of technology in roughing mill rolling process. All tasks are created by variants

according to the authorized user. In short, in this software program, it can be used not only in office training, but also remotely wherever the operators are. Use oculus go to provide feedback and training courses remotely.

### **3.3 Roughing Mill Environment Creation**

The preparation before unity modification is mainly operated in 3Ds Max and Blender. The reference scenes to be used came from real-time pictures and video recording for the training simulator software.

#### **3.3.1 Data Collection**

There are several data sources in the study. Data for modeling rolling stands and tables come from typical plant mill process blueprints. Since steel roughing rolling mill stand is an necessary part of work stands, it is key to read the blueprints and know the exact size of each parts width, length and height. The distances between each rolling stands are also very critical since the calculation will be done by scripting the slab movement speed taking reference from the data.

#### **3.3.2 Modeling**

Modeling is an important piece and the beginning part of the visual reality system, which will represent a specific geometry in the virtual process environment. To ensure the virtual system running smoothly, the proper modeling tools are chosen as 3Ds Max and **Blender**. The virtual reality environment includes plane, lights, safety signs, stands, control room and so on. The precise models arranged location, roughing mill process environment and models of relatively sizes are important for accomplish good quality GameObjects of scenes. A 3D model is essentially a collection of 3D coordinates called vertices. They are connected to form a triangle. Each vertex can contain information such as the color, the direction it points (called normal) and some coordinates (called UV data) to map the texture to.

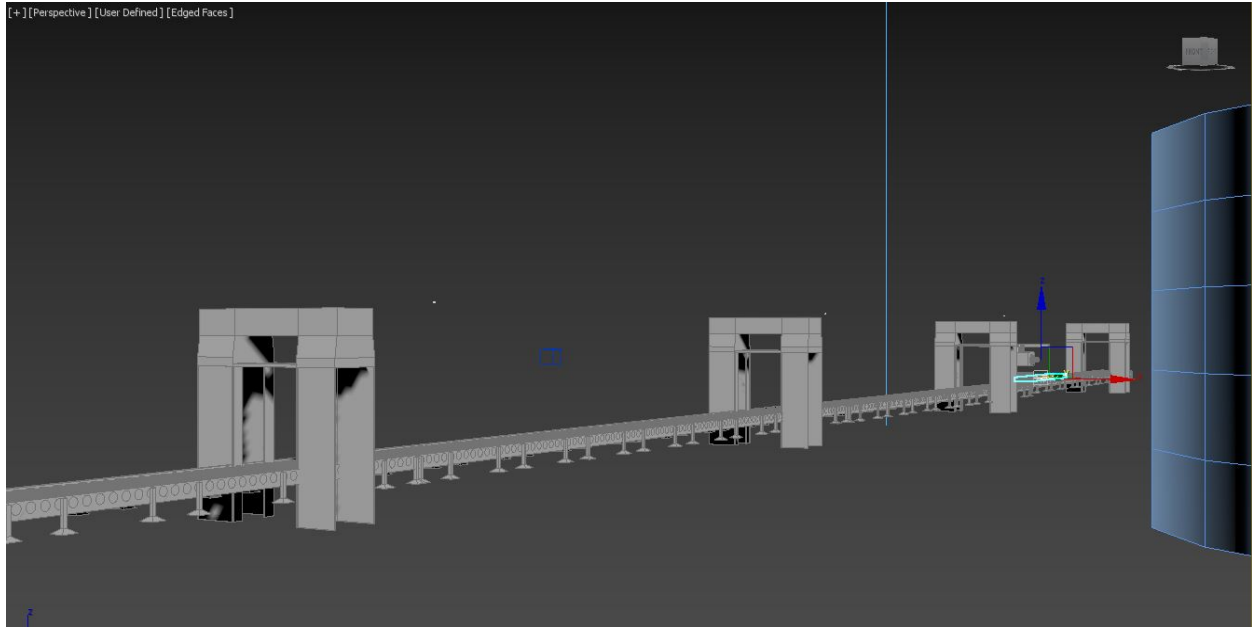


Figure 10. Rolling Stands Models

The figure above shows the ratio of distance between each rolling stands. In three dimensions view, the real distance will be reflecting right data through the straight angle while the visible view will be changed by the rotation of main view (Figure 10). In the developer view, the programming and modeling environment creation make the computer hardware "invisible" to the user instead of telling every detail.

3D models are important in the virtual environment where virtual modeling tool, 3Ds Max, is implemented to build the models. The original data of the models take the reference from Roughing Mill Process blueprint maps. Attaching textures on models to perform the mimic appearance of the slab is also vital since the rendering and shading process before exporting the slab. Textures are obtained by pictures from different resources of slab's surface.

As for slab modeling, it starts originally a simple cube box in 3Ds Max. Many implementations are taken into unity and controlled by scripts. The strategy changed because of simulation results not in an expected and usable status. Then, the modeling part of slab deformation is raised up which also be tried out in Blender and 3Ds Max. More testing illustrations will be stated in the later chapter. The idea of deformation meeting the requirements from scripts control separates the slab into pieces evenly. These segments rotate in a certain degree will create a curve

with accumulated degrees angle. However, the performance of curvature is the key issue needed be solved and optimized. The ideal of curvature finally will be similar as a visible smoothly bend.

The scenario zero is programmed as all basic functions with default correct settings, which means if the user doesn't do anything, the slab should smoothly finish the roughing mill process and become a standard slab in the end. The module of the heat map (Figure 11) is allowed to describe the slab in different conditions with temperature changing. The users consistently change the inputs of side guides and screws values to maintain the main process of the roughing mill and meanwhile learning how to operate the roughing mill process and get familiar with the control operation during the slab status updating.

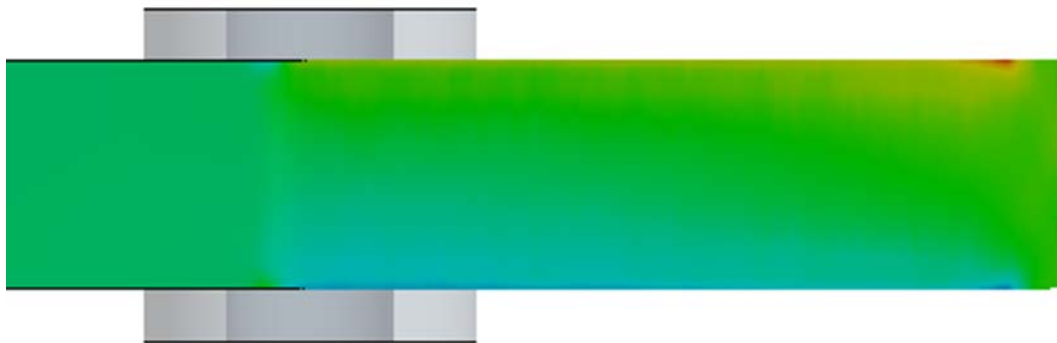


Figure 11. The Slab Model with Heat Map

As it is obvious that models cannot be rendered without a material, materials consider as wrappers contain a shader and store the data information for its properties. As for the slab, the lava materials are attached to it to mimic a realistically heating slab (Figure 11). The results from simulation side also is tried for wrapping out the whole slab model. By observing its color mapping, the slab can be in a real roughing mill process when passing through the rolling stand.

There are mainly three approaches for slab modeling to show the best performance of curvature. The first approach is designing and creating bone structures for the slab. The view from top to bottom for the slab is a rectangle, which is expected to see a bend into different degrees. Three groups of bones are connected to each other called right, center and left. Each group will have the same number of joints connections between bones nearby.

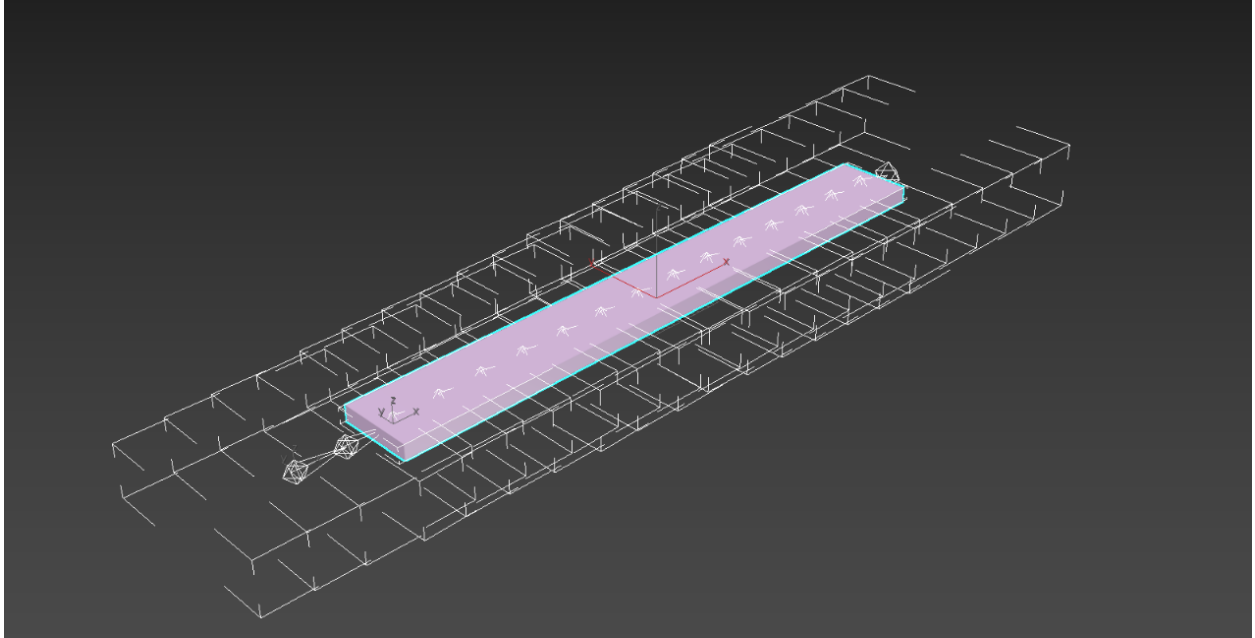


Figure 12. The Slab Segments Model

The second approach will be a straight cube box in 3Ds max wrapping with vertexes as a map. The vertices can be calculated and changed the whole slab deformation. A mesh, as geometric model of a three-dimensional object, where the basic shape is composed of points or vertices connected by edges, is used in this deformation. In 3Ds Max, the mesh can be edited by transforming, replacing and deleting various elements or objects (vertices, edges, and polygons) which can be also applied by various changes using modifiers. The render-able surface of a mesh object is composed of faces or polygons connecting vertices and edges. There are some examples of mesh objects in 3Ds Max include primitives such as Sphere and Teapot, as well as "editable mesh" and "editable polygon" objects. A good example is illustrated as a flower branch blowing by freeze and bending as a slightly “C” shape.

The final approach for the current project is modeling box segments (Figure 12). The design of these is focusing on each segment’s connections, which also can be considered as inheritance relationship among segments. The first segment will be the parent of the second while the third will be the child of second segment. This arrangement happens in 3dx Max when the slab is first designed, also can be viewed as a group of small slab segments with a big texture wrapping all together. In this way, the vertices are connected, and it can barely be observed the gap between each part since the vertex map has covered all of them. The design idea offers a flexibility for latter

scripting control at the same time. In the long term, modeling is the fundamental start of this study which gives various room for developing different functionalities on the slab.

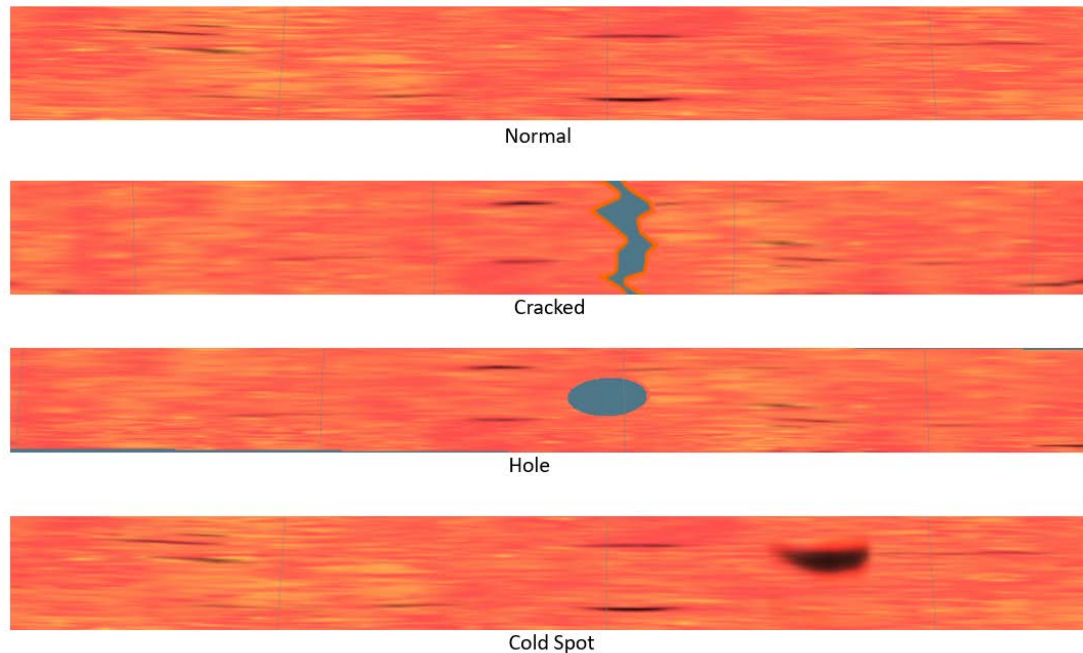


Figure 13. The Different Slab Models

Slabs can be in various conditions for different cases. In the scenarios, the slab initial status is different with each other. Hence, the modeling of different types of slabs will show the variety of model performance, such as Cracks, Holes and Cold spot, etc. (Figure 13). More details of scenarios' samples can be found in Appendix A.

### 3.3.3 360-degree Embedded Virtual View

Once the environment model is completed, the saved 3D models are exported in the format .fbx to maintain compatibility with Unity3D. Other elements such as animations, scripts, textures and sounds are stored in the associated asset files found in a Unity3D project. During creating the scene, these are stored as GmeObjects in Unity 3D. The designed structure kept its style after exporting from 3Ds Max and it's also user-friendly and can be modified with the saved .fbx files.

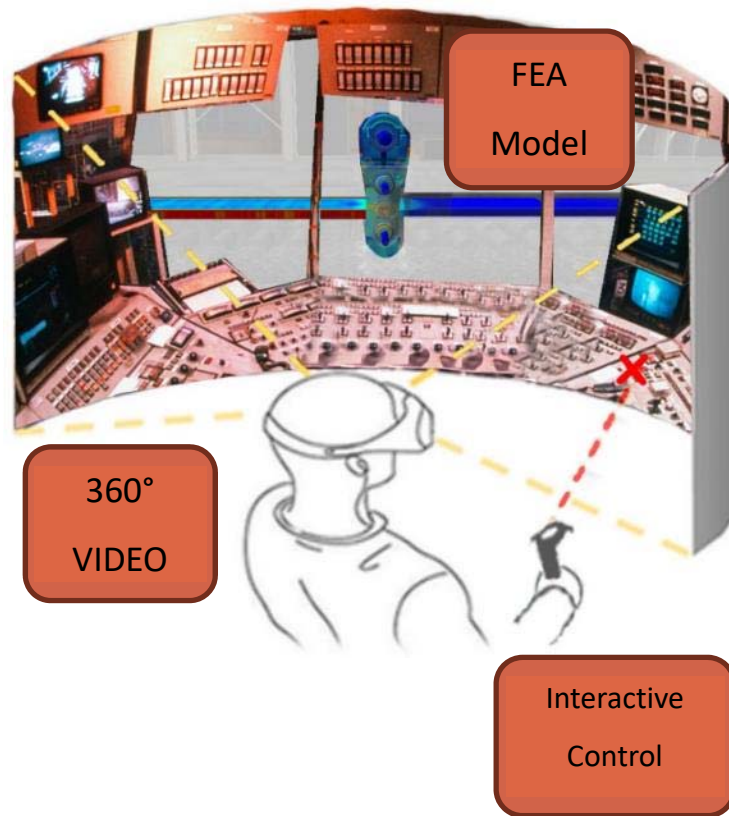


Figure 14. 360-degree View

One of the most important functions of VR is to transport and completely immerse the viewer in a new experience or environment. Similar to the traditional narrative and the immersive physical environment, the immersive technology makes people feel on the spot. However, immersive technology may have unique capabilities that can help users feel that they are in the environment and experience the situation from the perspective of an actor. There is evidence that immersive technology can elicit emotional responses from users, such as empathy and affect behavior changes. Compared with text news describing the same event, the use of 360-degree video storytelling and reporting can produce higher empathy, immersion, and participation. In addition, compared with traditional videos, 360-degree videos attract more positive emotions. Following with the layered Virtual Environment, the models exported to Unity where the user can see the view for simulation and process running. The 360-degree video works as a surrounding texture overlapping the control room while the window cutting off as embedded view for the software setting (Figure 14).

### 3.4 Deformation and Curvature

To move the slab into different visible deformation, Segments are involved and placed in an array with available movement in x, y, z three axis/dimensions (Figure 15). As mentioned in modeling part, the texture attached to the slab can be also edited in unity by different settings options. Brightness will be not only reflecting on the light object in the scene but also in the adjustment of the material texture. And the other characteristic element will be the modeling part in 3Ds Max when the slab be designed.

As the critical part in the whole model, slab deformation is the hardest topic being discussed many times and have tried a lot of experiments on it. The combination of programing and modeling make the slab movement and deformation vivid like a real-world simulation. Hence, there are also three approaches of the slab programing part.

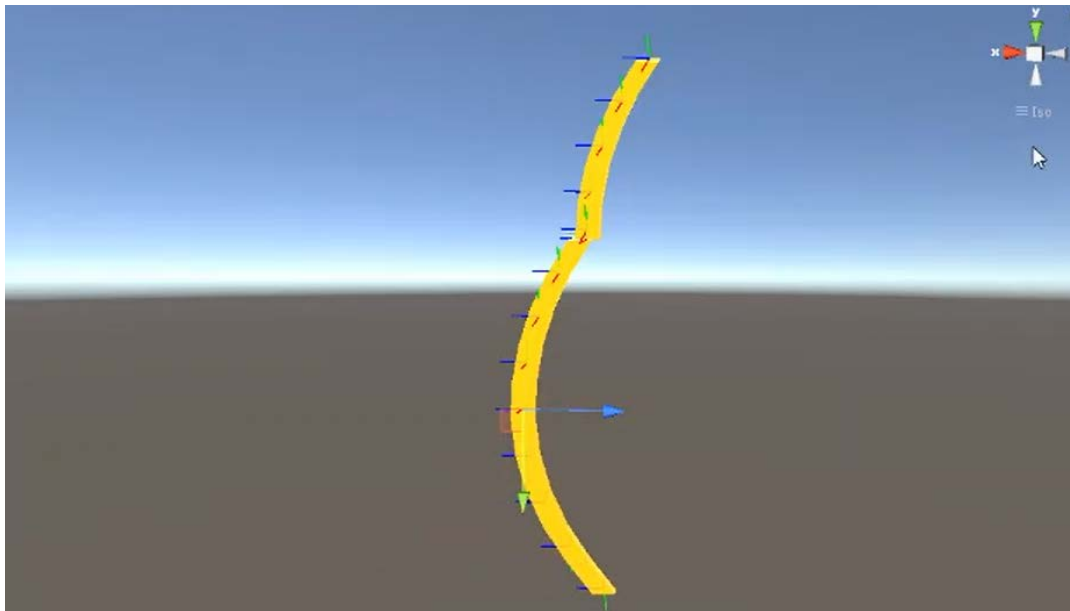


Figure 15. The Curved Slab Model with Gizmo

The first approach will stay with the modeling part in Blender that each joint is scripting by some degrees while the option chosen differently through the UI panel. Simply, it shows the straightforward relationship between buttons and slab different status. The expectation hasn't set up a linear or higher level- nonlinear deformation happening on the slab during the roughing mill process.



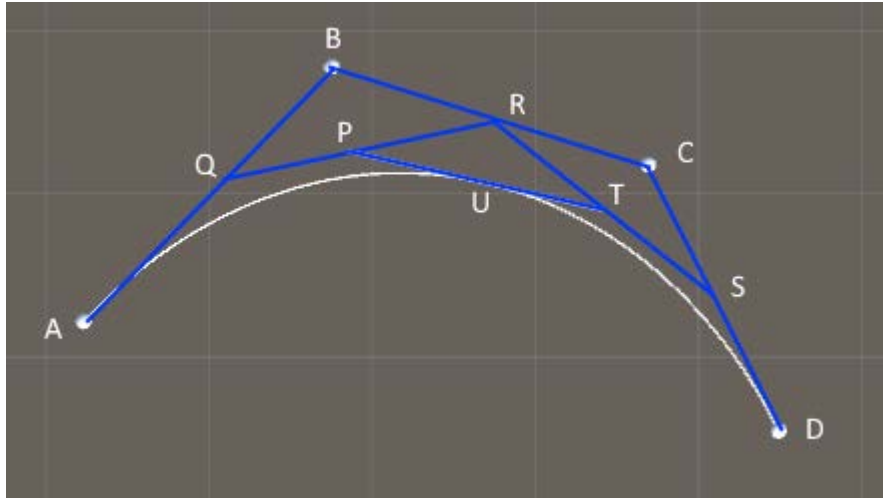


Figure 16. Bezier Curve

The second approach begins to bend in some degrees to cause a smooth curvature. By inspiring from an asset, as simple bend deformer is utilized and attached to the slab model meshes for testing and performing (Figure 16). As the bellowed screenshot showed, the slab actually is organized by three segments/boxes for making more than one curve. Not only the slab curvature is looking for “c” shape but also a “s” shape or even more curvatures are needed. However, comparing with the real roughing mill process, this deformation is not enough while the slab is going through each stand. The Bezier curve, as the most basic spline, is applied to design streamlined cars, a case first invented this curve definition. The goal of this case is to build roads in the Skylines of the game cities in Photoshop. The original idea is that by using two end points and two control points to draw a curve which forms the final spline. The curve also can be interpolated linearly in different layers with De Casteljaun's algorithm [47].

Also, the slab movement need to be considered when the bend is happening. Then, the degree of the curvature will be accumulated by each frame or second. The bending algorithm is well-designed with lerp application. Lerp is representing the idea of linear interpolation. While in Unity, Lerp can be used for finding a desired degree calculation result between two known points positions. As the points' x and values are known, the Lerp can run the calculations through the library to give an angle degree as results. It's needed to be awarded that Lerp are essentially mis-used to make a smooth-damp by using Time.deltaTime and the start position can be reassigned.

The whole Lerp functions work as below process: a demo example script was attached on a GameObject for testing purpose and the axis's values can be kept tracking. From the game view, the GameObject could be seem “pause” for a while. The truth is that changes in the GameObject's position can be observed slightly jumping. Performance will be maintained when these changes of tiny differentiation being separated to meet the presetting float values. Lerp for smooth dampening can be applied to this scene for slab curving. Also, the nature curvature cannot live without the math computational process.

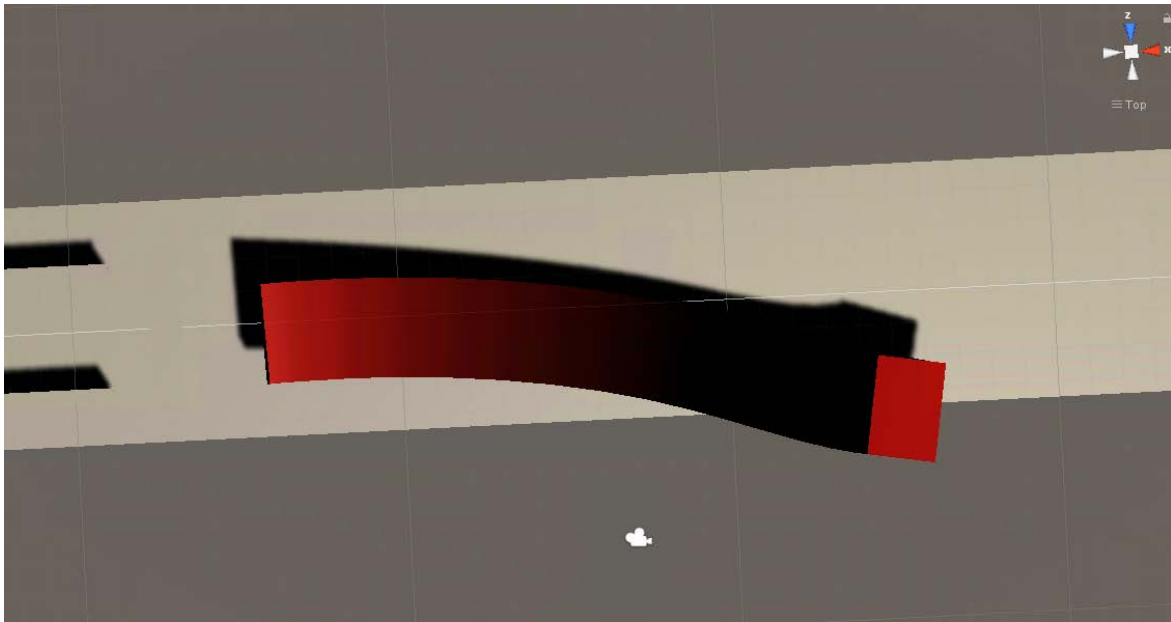


Figure 17. The Bent Slab Model

The last approach appears when the requirements of other functionalities are raised up. For screws functions, the slab should curve towards one side than other (Figure 17). The differential values between the default values and the current values after the user's operations control the slab bending degree. The bigger difference the more curved bend will happen. Rotation of each segments become accumulated to create the simulation of slab bending vividly. It is very easy to observe that the first segment of the slab model which also is the beginning end curves the most angle. And this script combining with the new model give them enough flexibility for changing during the movement process.

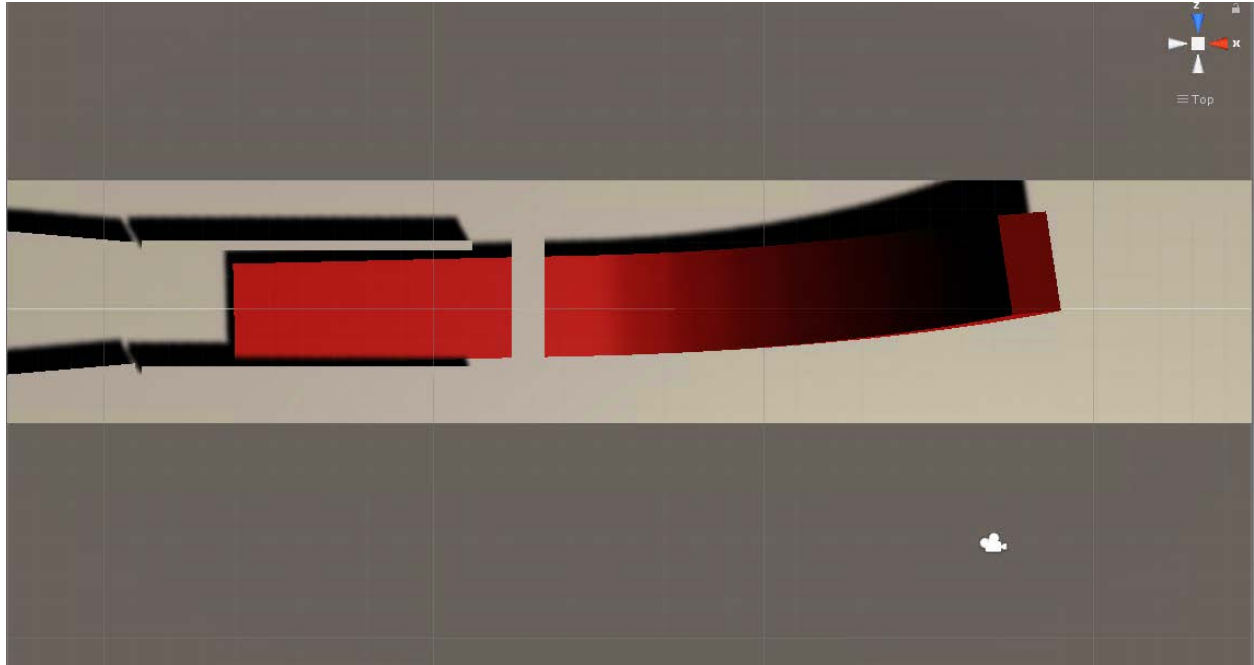


Figure 18. The Bent Slab Model Towards to Opposite Direction

For example, the slab showed above perform as curvature step by step towards to an opposite direction (Figure 18). This difference rotation and deformation is the result of different user's input. Screw values are changed through user interface and data is passed by scripts to calculating the new deformation and the direction of the rotation or curvature will depends on screws' values with positive value or negative.

Beyond the screw values and functions, there are many other factors that cause different impacts on slab. Edger's functionality focuses on the slab size changing. Due to the slab segments structure, the parent-child inheritance structure needs a recursion for gradually size changing by the scripted logic. This simultaneous process needs more interaction from UI and communication between different scripting block for several tasks.

Slab movement is pushed by force and the speed can be adjusted indirectly through the physical process. Table gives the reference speed through the friction theory to push the slab moving forward while the thrust parameter also give it a forwarding force. Since the side guides are also set to rigid body. When the collision happens, slab's speed will be simulated as a friction collision and go down step by step till it stop. This designed process and simulation give the flexibility of slab movement and interaction between the simulation process and user interphase

control. The experimental case is that once the user readjusts the side guides value to the maximum values, the slab can go through and continue moving forward again.

### 3.5 Accessing Information by GUI Control

In this study example, buttons, panels and monitor pop up panels with data table are the main GUI. Scripts can be attached to these interactable object in Unity 3D and functionalities can be added by programming in C#. Some examples can be that a button with the function working as start the training process while the logic scripts for this button may be assigned initial values to the model slab. After clicking on start button, a new scene opens based upon the roughing mill process in the GUI and the slab starts moving.

When some monitor panels and buttons are interacted with in the GUI (Fig. 3), it may pop up new panels with new options and buttons. This embedded style, a one-dimensional (1D) or two-dimensional (2D) monitors 'panels-based scene will be selected. At the same time, GameObjects and scripts linked with the specific kind of task will be functional running.

In order to customize the appearance of the scene, the user can choose different scenes with different input slab models. Additionally, various types of objects such as rolling stands, rollers, etc, can be included to provide a more immersive experience for the user with some being available in the Unity Asset Store [48]. More details of these available codes can be found in Appendix B (Figure 19).

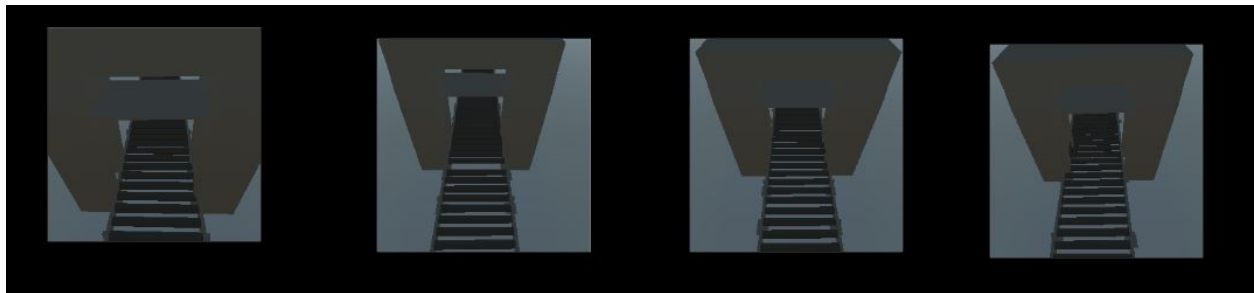


Figure 19. The Monitors Tracking Slab

## 4. RESULTS AND DISCUSSION

The goal of this example in the study is to generate a virtual process system following the proposed diagram. For the future usage and practical training, the virtual reality system can be generated the android standalone. The project us built in the hardware, virtual reality headsets. Oculus go is selected in this study and the first testing happens through the mirror view of the oculus go. During the running software on the hardware, there more elements needed to be considered due to the performance issue.

### 4.1 Virtual Reality Platform

The hierarchical approach of Roughing Mill information system is developed next, and the design process is built by several patterns: slab modeling, control UI panels, backend logic scripts, performance and reflections. Figure 20 shows the details of the patterns in the project.

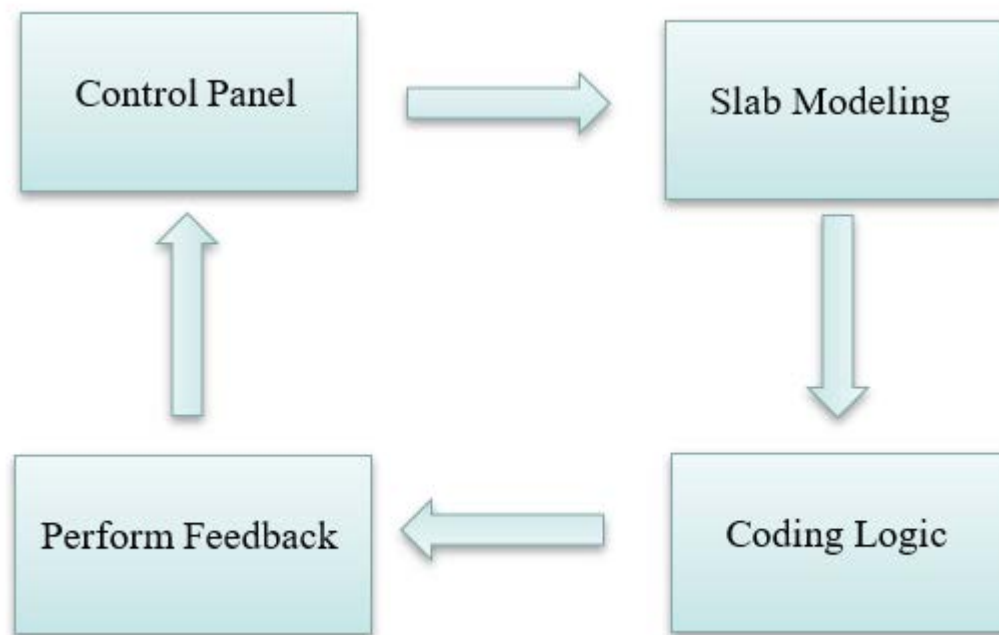


Figure 20. The Design Pattern

In the virtual reality system, the project is implemented by a view of 360-degree control room. Control panels are created and shows how GUI work with the modeling process part. As for

slab modeling preparation, 3Ds Max and Blender are used to create slabs of different supported functionalities in various scenarios. While the software is running, coding scripts will compile to give the calculation results. This updated process will pass through performance and give users feedback in real-time in order to have user's new inputs entering the control panel. The closed loop circulation will be accomplished by the above steps one by one.

## **4.2 Feedback**

Both cases are tested for success and failures. And the slab speed is also rechecked by the real recording materials. Specially for the collision detection part, the testing and revision is mainly to solve the issues caused by collision. Some gap between the models are not obvious. However, it will cause the slab falling from the table and continuing falling or flying due to the thrust and friction disappear. The mass and gravity also need to meet the requirements for the physical phenomenon.

## **4.3 Training Performance**

The virtual process system should be recording performance data. The data can not only be updated to the monitor's panels but also can be stored in the back-end scripts to give a performance evaluation for users and for developers to optimize software and hardware configurations, and then bottlenecks could be found. The user may enhance and optimize their performance through viewing the performance data being collected. For example, the slab straightness as the priority standard rule to make a judgement whether it is straight enough for the slab after a roughing mill operation process (Figure 21). Furthermore, the log files which can store the user's action during the complete scenario will be stored for the report card. Once the user clicks on the action's buttons, it will pop up a panel listing all the related actions has been done in the past training scenarios. By implementing this performance showing panel, the designed diagram with workflow can be optimized by users' feedback and reinforce learning while seeing their action history.

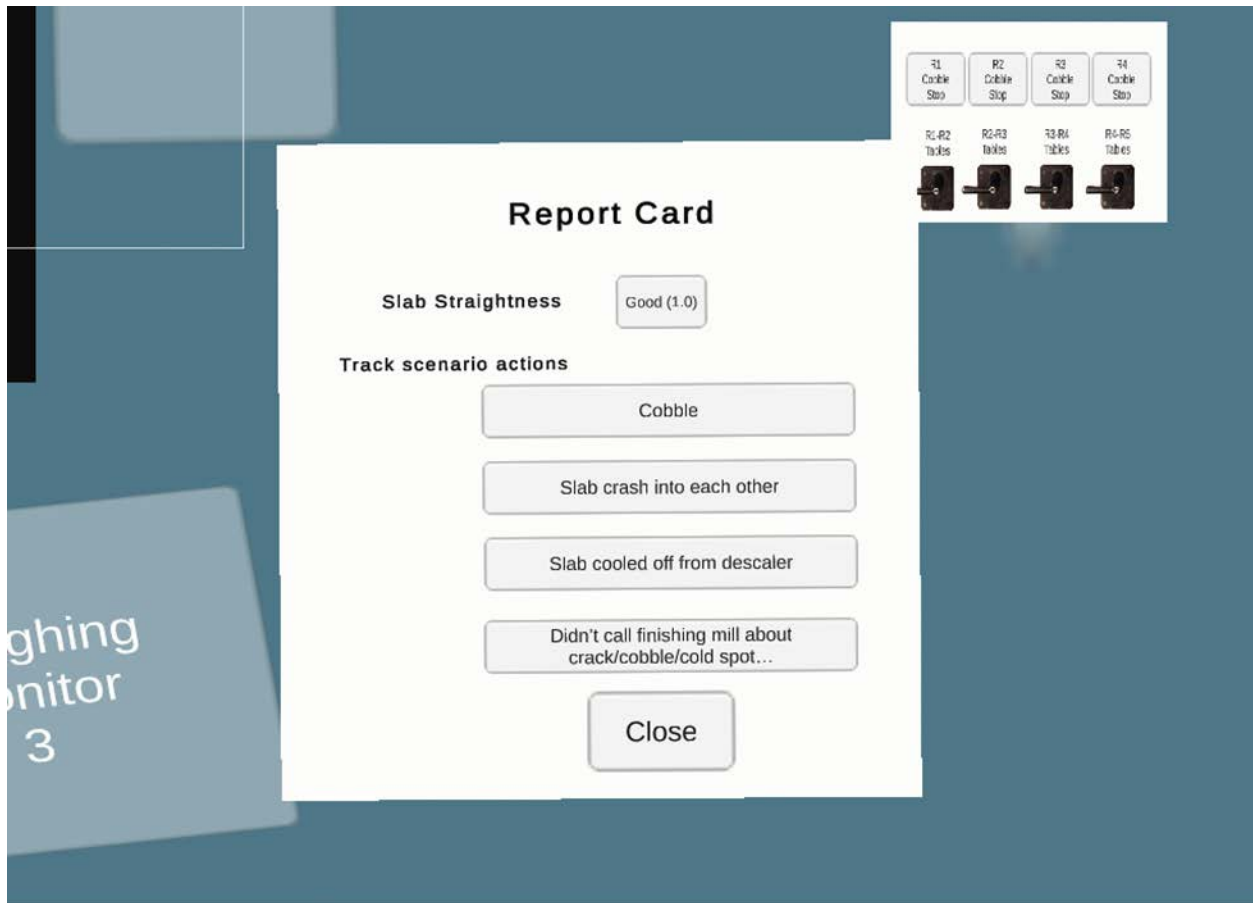


Figure 21. Slab Report UI

Compared with traditional training process, the report card can be considered as a score card or a paper writing sheet with all answers from operators. This traditional situation, also similar to the non-immersive systems, computer screens are running a software with electronic questions and static pictures [49]. Users easily get bored of this traditional system and the performance of the real training results will affect the work efficiency of operators when they actually work in the real control room environment. Hence, the immersive idea has been brought to the surface. Virtual reality will be the technology for the new implementation. Since the virtual process will enhance the operator's sense and ability for mastering knowledge, more details are designed for finish the workflow and optimized by applying more features from software development side.

In this study example, manufacturing process training software give the performance report for users and help enhance and optimize the performance during the immersive training. The results in the example and this study follows the training performance for future upgrade design version.

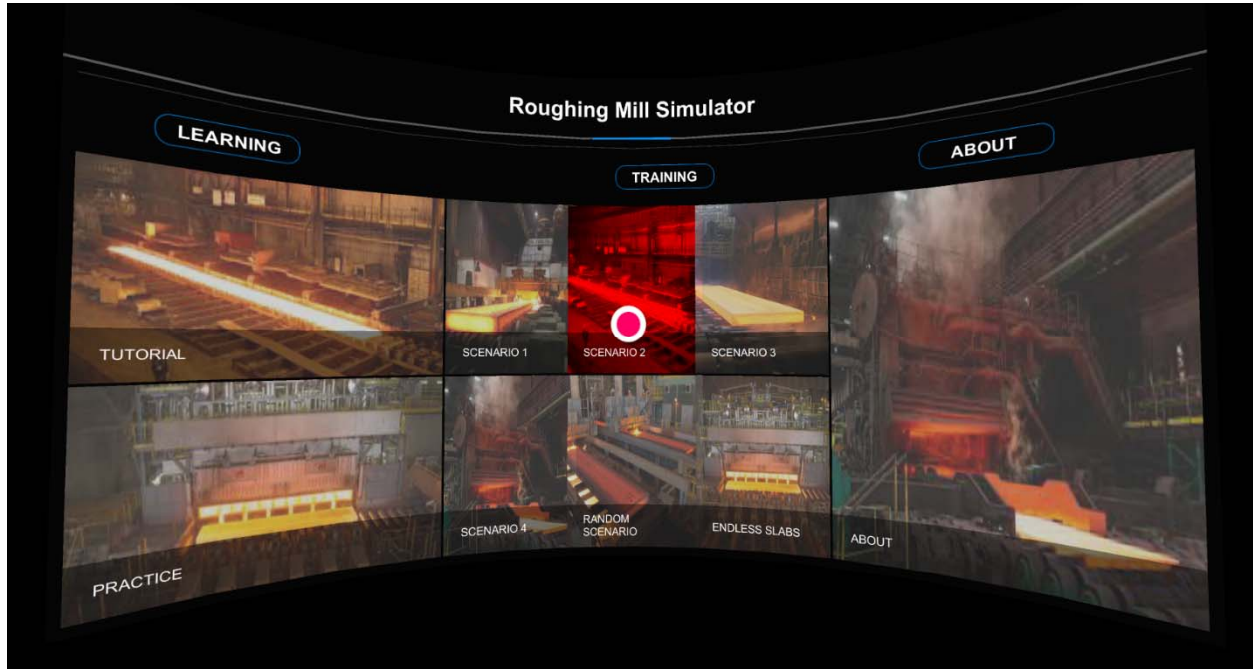


Figure 22. Main Menu UI

It is a UI design in starting this training software. The user can start training here to check on tutorial about how to use this software (Figure 22); the practice pane gives a new user the ability to get familiar with this new training environment and all the functions utilized by the VR equipment. The real training part will allow for the choice of these scenarios each one having different topic areas and problems. There is a system of scenes consisting of practical scenarios, tutorials, control practice for checking the gained knowledge. The subsection of practical scenarios consists of several input modules on the main menu for the main topics of this roughing mill module.

Slab variations will appear in the different scenarios as inputs. Right now, there are four types of them. Normal, Cracked, Hole and Cold spot. These different conditions will cause slab to perform various through the roughing mill process. And the use, as new operator will learn how to deal with these situations while using this training software. The prototype of the draft version about control room, also which is used this as the software UI panel later. And the design idea about pop-up functional is also easy for user to figure out more details about each control panel parts. The operation user made will be shown in the software scenario like parameter changing and caused the outputs differently.



In this application, the user interaction with the VR environment is high. This project also in a form needed for better performance and interaction experience.

#### **4.4 Discussion**

The performance limitations of virtual reality are divided into two categories: visual display limit the display frame rate required to provide the environment the impact of immersion and existence; and interactivity constraints the waiting time from the input provided by the user to the response provided by the system (Visual or otherwise) necessary for the user to react in fast and effective speed while controlling the operation process. The visual display has the limitation refers to the frame rate of the running software, and the interactivity constraints refer to lag in the system. One constraint may be satisfied, while another constraint it may fail. Failure to meet hardware display performances constraints will result in hallucination failure a sense of immersion and presence. Failure to meet the interactivity constraints will result in users cannot precisely control objects in the environment through direct interactive with GUI. These constraints have two components: the bottom-line component. According to human factors research, this will apply to all virtual environments; and a more rigorous components based on this theory will be applied to objects with fast moving objects sampling.

While besides to see the impact on a mobile platform, here's some of the conclusions that came to while looking at the project/statistics: the software is actually very hard on CPU single thread performance, specifically the world thread. There are about 170 instances of TextMeshPro Assets that are processed every frame and they take about .01 seconds to process, which not the main cause of the problems with frame rate, are still somewhat significant. TextMeshPro shaders are already set to be the most optimal for mobile VR platforms, so there are no real ways to optimize, except for possibly using default text assets. There is a huge bottleneck in the performance that being noted in the CPU department is specifically the script for the rollers. When on, each roller contributes about .02 seconds (total of about .12 seconds) per frame calculation, which is one of the primary contributors to the slow frame rates. Turning them off yields an almost 75-80% uplift in performance on PC, which may be even better on lower-clocked cores on mobile. The GPU utilization is remarkably low for the most part, but any additional loading may decrease overall performance even lower because of thermal limits, just something should be considered for future work [50].

If there was a way to make the rollers operate as subroutines that could go to other threads, that would probably help a lot too, in addition to potentially getting rid of them. Then, the model approach evaluated and improved into the third approach. The slab is remodeled for an intension to use the physical engine in unity. This solution solves partially the low FPS problem.

To provide a stable development system, an API to replace the device at runtime, and an abstraction that prevents the application from noticing changes in the foundation are needed. The choice of hardware mainly depends on the selected VR application attribute level, such as authenticity, immersion, and interactivity refer to the goals originally intended for application usage. In order to support more involved user perceptions, the hardware and software device and programming system will need the higher complexity. The most common sense involved in current VR applications is sight and hearing. Since tactile stimuli (touch, temperature, etc.) are still stunted, users can only use their hands to manipulate the controls.

Performance constraints based on research results of human factors are most suitable for an environment where the object will not move around unless the user manipulates the object. The results of human impacts research tell us that the frame rate of visual display must be more than about ten frames per second, so that visual changes the image acts as a continuous motion rather than a series of still images can perceived. It should be noted that this is not the discreteness of the frame cannot detect the required frame rate, but this constraint requires the interpretation of visual changes as continuous cognitive effects movement. The features of interactive constraints are more flexible: a delay of only 30 milliseconds severely reduces the performance of trackers task. However, humans can easily adapt by slowing down movement in the environment. While if the lag time is close to half seconds, direct interaction becomes very slow and basically unusable. It is shown that the lag time is as high as 0.1 seconds is tolerable, as long as the user interact with objects that only move below the user control during past experience.

Also, this study is mainly to develop a training software while the simulation side are simplified by using the unity physical engine. For future development, it is a very challenging topic to develop the simulator out of unity such as Ansys. The real-time data simulation and communication in unity will provide more concrete results and slab deform and heat transformation. APIs between simulation software and unity are necessary for deployment. Cloud GPU services or huge datasets cloud services will be the key issue to figure out. The theory for this foreseen development is similar to a digital twin.

The future step for this example project can be integrated digital twin which constantly studies and optimizes itself from multiple sources in order to represent close to real-time status, working conditions or location. The manufacturing operation process as a learning system will use data to transmit the operation status information to give the feedback learning from its own and can also from professional experts, such as operators with years industry knowledge. As for the data usage, the digital twin also integrates historical data on past equipment or data files and optimized into its digital model. Twins are also used to update the operation process in various industrial fields.

## 5. CONCLUSIONS

This thesis ultimately generates a virtual reality system based on Unity3D for a roughing mill process as an applied example. This method for interactive virtual process that can be applied to many manufacturing processes not limited to this specific case. After selecting the software and hardware, a VR platform with Oculus go or some steel industry specific equipment for operators to have engaging experiences with real-time physiological status showed in-game.

The interactive virtual environment "Roughing Mill" is innovative creation based on the field of mechanical engineering and operation training. It supports some flexibility for users when they in the operation mode. This designed virtual process system not only provides an immersive environment but also gives the arability for random customized options rather than traditional static training computer screen or paper testing. It must be noted this paradigm is not the only virtual simulator which can be used at the factory for the training new operators. This design diagram and patterns can be split and re-engineering for any knowledge-based system. The process model and GUI will be able to be applied to many different situations and cases. As for its training purpose, the trainee can be educated further from a normal learning experience while coaches or professional manages can save their time by input their new knowledge to update the training scenarios. The process related to education and learning can be easily applied by changing the details environment and parameter setup. What's more, the new training method and process example shows new operators can master skills such as those required by the roughing mill in detail quickly without big material costs.

In this thesis, a workflow diagram is introduced which combines a software application framework with a software engineering approach to virtual process development. The architecture of the framework training system is patterns including few components based on GUI and model processing. The purpose is to provide future possibilities to expand this diagram for more application and industries usage. [51].

The innovations for this research start from the beginning status 360-degree video and whole panorama view for users. Back-up video is playing when users can operate through clicking on the buttons. Next, physic engine is used in the simulator as modeling and scripting for controlling the process. Providing the feedback is also a training purpose to check the slab deformation performance. For this research, it presents a user-friendly, quick-to-learn interface to

the users and the controller usage is simple functional with only two active buttons. Performance is one of this project top priorities. The designed workflow can be believed and support the virtual process system. The design process in this training simulator is partially based on the “waterfall” model where the design view meet requirements and make it through implantation. Analyzing the performance and testing the functionalities at each step of design ensure the efficient implementation process.

As a consequence of this framework’s modular nature, it is felt that it can be easily adapted and customized for new VR industrial training applications, which can be more portable and scalable. Initially, this study envision is some applications in the steel industry domain using this framework. Furthermore, as for this example project have more upgrade room which can be integrated with Internet of Things, artificial intelligence and, machine learning along with software analytic capability to create living digital simulation models and similar optimized implementations could also be applied to other manufacturing industry applications.

## APPENDIX A. SCENARIOS TEMPLATE

- Scenario 1 (40" Tin Mill slab "wedge" shape)
  - Background
    - #1 caster makes wide slabs that have a belly
    - Wide slab is cut down center to become two slit slabs
    - Slit slabs each have a thick edge and a thin edge
    - Slabs are oriented the same
    - Start of shift, text from last operator – "Everything went good today. You've got 40" tin slit coming next."
  - Objective
    - Induce camber (out, toward pulpit) to remove some of the wedge before finishing mill
  - Actions
    - Flatten out the wedge at HSB, R1, R2 (induces camber toward pulpit)
    - Jog at R3 (and R4, R5 if needed) to remove camber and straighten the bar back out (Front Down)
      - Choices (HSB,R1,R2 - .05, .10, .15; R3 - .01, .03, **.05**, .08; R4,R5 - .01, **.03**, .05)
    - If slab gets stuck (R1, R2, R3, R4)
      - turn off descender (red button)
      - Let out side guides
    - When slab starts moving again
      - turn descender back on (green button)
      - Reset side guides to reference
      - Change side guides reference to 10" (Roughing Mill Center Monitor)
    - If operator didn't reduce camber by time it reaches R4, let out E4 to prevent cobble, and jog R4 & R5
    - If cobble, press R4 Cobble Stop (stops R4 & R5), stop all tables (turn pistol grips down)
  - Consequences
    - If slab is stuck in descender water too long, slab will cool off
    - If operator doesn't straighten bar at R3 & R4, slab hits south edge of E4 and cobbles between R4 & R5
    - If we get a cobble, and operator doesn't stop tables, the next slab will still come and crash into the cobble
  - Behaviors
    - Force meters (R1-R5) show bite as slab passes through

## APPENDIX B. PESUDO CODE

### Roughing Mill Simulator Specification (11-25-2019)

- Variables
  - slab[n]
    - segmentVertex[n] (Bone structure, 10 bones)
      - X
      - Y
      - Z
    - Temperature
    - Texture
      - Brightness
      - Cracks
      - Holes
  - Rollers
    - Side Guides
      - HSB, R1, R2, R3, R4, R5
      - Value (float)
    - sideGuideReference
      - HSB, R1, R2, R3, R4, R5
    - screwValue
      - HSB, R1, R2, R3, R4, R5
      - Value (float)
    - screwsReference [default = SEE VIDEO]
    - Spray
      - HSB, R1, R2, R3 (R4? R5?)
      - Value (Boolean)
  - Tables
    - rollerSpeed (maybe boolean)
  - Emergency Stop
    - Value (Boolean)
      - Controls Tables.rollerSpeed
  - Time
  - Monitors
    - Text values [different for each monitor... multiple rows]
    - Texture
  - USER CONTROLS
    - SideGuideButtons
    - ScrewButtons
    - EmergencyStop

- TablePistolGrips
- Horn
- ...

#### Scenario 0 (Normal Slab)

- User can do nothing, and slab will make it through in good shape
- slab
  - segment
    - length=60"
    - width=60"
    - thickness=4"
  - temperature=2200
  - Texture
    - Brightness=1
    - Cracks=0
    - Holes=0
- Roller
  - Sideguide=sideGuideReference
  - screwValue=screwsReference
  - Spray=[0 when no slab present, changes to 1 when slab is present]
  - [WHEN SLAB PASSES THROUGH ROLLER, AFFECT THE SLAB BY THESE VALUES]
    - Slab.texture.brightness-=0.01
    - Slab.temperature-=1
    - Slab.segment.thickness-=screwValue
    - Slab.segment.angle+=0;



### Scenario 1 (slanted slab)

- If user does nothing, and slab will stuck in different place
  - HSB
  - R1
  - R2
  - R3
  - R4
  - R5
- USER CONTROLS
  - SideGuideButtons
  - ScrewButtons
  - EmergencyStop
  - TablePistolGrips
  - Horn

### Initial Status:

- slab
  - segment
    - length=60"
    - width=60"
    - thickness=4"
  - temperature=2200
  - Texture
    - Brightness=1
    - Cracks=0
    - Holes=0
- Roller
  - Sideguide=sideGuideReference
  - screwValue=screwsReference
  - Spray=[0 when no slab present, changes to 1 when slab is present]
  - [WHEN SLAB PASSES THROUGH ROLLER, AFFECT THE SLAB BY THESE VALUES]
    - Slab.texture.brightness-=0.01
    - Slab.temperature-=1
    - Slab.segment.thickness-=screwValue
    - Slab.segment.angle+=0;

## Scenario 2 (cracked slab)

- If user does nothing, and the scenario will fail
  - May happen like stuck as Scenario 1
- USER CONTROLS
  - SideGuideButtons
  - ScrewButtons
  - EmergencyStop
  - TablePistolGrips
  - Horn

## Initial Status:

- slab
  - segment
    - length=60"
    - width=60"
    - thickness=4"
  - temperature=2200
  - Texture
    - Brightness=1
    - Cracks=0[changing]
    - Holes=0[changing]
- Roller
  - Sideguide=sideGuideReference
  - screwValue=screwsReference
  - Spray=[0 when no slab present, changes to 1 when slab is present]
  - [WHEN SLAB PASSES THROUGH ROLLER, AFFECT THE SLAB BY THESE VALUES]
    - Slab.texture.brightness-=0.01
    - Slab.temperature-=1
    - Slab.segment.thickness-=screwValue
    - Slab.segment.angle+=0;

## REFERENCES

- [1] Lee, W.B. "Applications of Virtual Manufacturing in Materials Processing." *Journal of Materials Processing Technology*. 113.1-3 (2001): 416-23. Web.
- [2] K. Iwata, M. Onosato, K. Teramoto, S. Osaki, A Modelling and Simulation Architecture for Virtual Manufacturing Systems, *CIRP Annals*, Volume 44, Issue 1, 1995, Pages 399 402, ISSN 0007-8506, [https://doi.org/10.1016/S0007-8506\(07\)62350-6](https://doi.org/10.1016/S0007-8506(07)62350-6).
- [3] Fumihiko Kimura, Product and Process Modelling as a Kernel for Virtual Manufacturing Environment, *CIRP Annals*, Volume 42, Issue 1, 1993, Pages 147 150, ISSN 0007-8506, [https://doi.org/10.1016/S0007-8506\(07\)62413-5](https://doi.org/10.1016/S0007-8506(07)62413-5).
- [4] J. J. KOZAK, P. A. HANCOCK, E. J. ARTHUR & S. T. CHRYSLER (1993) Transfer of training from virtual reality, *Ergonomics*, 36:7, 777-784, DOI: 10.1080/00140139308967941
- [5] M. W. Rohrer, "Seeing is believing: the importance of visualization in manufacturing simulation," 2000 Winter Simulation Conference Proceedings (Cat. No.00CH37165), Orlando, FL, USA, 2000, pp. 1211-1216 vol.2, doi: 10.1109/WSC.2000.899087.
- [6] Bennis, F., Chablat, D., & Dépincé, P. (2007). Virtual reality: A human centered tool for improving Manufacturing. *arXiv preprint arXiv:0707.3563*.
- [7] Abidi, M.-A. "Simulation of Manufacturing Processes via Virtual Reality." *Robotics, Automation, and Control in Industrial and Service Settings* (2015): 142-78. Web.
- [8] Yongmin Zhong and B. Shirinzadeh, "Analysis, conversion and visualization of discrete simulation results," *Proceedings. Eighth International Conference on Information Visualisation*, 2004. IV 2004., London, UK, 2004, pp. 118-123, doi: 10.1109/IV.2004.1320133.
- [9] Sankar Jayaram, Hugh I Connacher, Kevin W Lyons, Virtual assembly using virtual reality techniques, *Computer-Aided Design*, Volume 29, Issue 8, 1997, Pages 575-584, ISSN 0010-4485, [https://doi.org/10.1016/S0010-4485\(96\)00094-2](https://doi.org/10.1016/S0010-4485(96)00094-2).
- [10] S. Darmoul, M. H. Abidi, A. Ahmad, A. M. Al-Ahmari, S. M. Darwish and H. M. A. Hussein, "Virtual reality for manufacturing: A robotic cell case study," 2015 International Conference on Industrial Engineering and Operations Management (IEOM), Dubai, 2015, pp. 1-7, doi: 10.1109/IEOM.2015.7093880.

- [11] Cheung CF, Lee WB. Modelling and simulation of surface topography in ultra-precision diamond turning. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*. 2000;214(6):463-480. doi:10.1243/0954405001517775
- [12] M. W. Rohrer, "Seeing is believing: the importance of visualization in manufacturing simulation," 2000 Winter Simulation Conference Proceedings (Cat. No.00CH37165), Orlando, FL, USA, 2000, pp. 1211-1216 vol.2, doi: 10.1109/WSC.2000.899087.
- [13] V. Winter, D. Desovski and B. Cukic, "Virtual environment modeling for requirements validation of high consequence systems," *Proceedings Fifth IEEE International Symposium on Requirements Engineering*, Toronto, Ontario, Canada, 2001, pp. 23-30, doi: 10.1109/ISRE.2001.948540.
- [14] Moohyun Cha, Soonhung Han, Jaikyung Lee, Byungil Choi, A virtual reality based fire training simulator integrated with fire dynamics data, *Fire Safety Journal*, Volume 50, 2012, Pages 12-24, ISSN 0379-7112, <https://doi.org/10.1016/j.firesaf.2012.01.004>.
- [15] T.S. Mujber, T. Szecsi, M.S.J. Hashmi, Virtual reality applications in manufacturing process simulation, *Journal of Materials Processing Technology*, Volumes 155–156, 2004, Pages 1834-1838, ISSN 0924-0136, <https://doi.org/10.1016/j.jmatprotec.2004.04.401>.
- [16] Zdenek Tůma, Jiri Tůma, Radek Knoflíček, Petr Blecha, Frantisek Bradáč, The Process Simulation Using by Virtual Reality, *Procedia Engineering*, Volume 69, 2014, Pages 1015-1020, ISSN 1877-7058, <https://doi.org/10.1016/j.proeng.2014.03.084>.
- [17] Gorecky, D., Khamis, M., & Mura, K. (2017). Introduction and establishment of virtual training in the factory of the future. *International Journal of Computer Integrated Manufacturing*, 30(1), 182-190.
- [18] Brough, J. E., M. Schwartz, S. K. Gupta, D. K. Anand, R. Kavetsky, and R. Pettersen. 2007. "Towards the Development of a Virtual Environment-Based Training System for Mechanical Assembly Operations." *Virtual Reality* 11: 189–206. doi:10.1007/s10055-007-0076-4.
- [19] S. Darmoul, M. H. Abidi, A. Ahmad, A. M. Al-Ahmari, S. M. Darwish and H. M. A. Hussein, "Virtual reality for manufacturing: A robotic cell case study," 2015 International Conference on Industrial Engineering and Operations Management (IEOM), Dubai, 2015, pp. 1-7, doi: 10.1109/IEOM.2015.7093880.

- [20] Chevaillier, P., Harrouet, F., Reignier, P., & Tisseau, J. (2000). Virtual reality and multi-agent systems for manufacturing system interactive prototyping. *International Journal of Design and Innovation Research*, 2(1), 90-101.
- [21] Liang Hu, Zhenyu Liu, Jianrong Tan, A VR simulation framework integrated with multisource CAE analysis data for mechanical equipment working process, *Computers in Industry*, Volume 97, 2018, Pages 85-96, ISSN 0166-3615, <https://doi.org/10.1016/j.compind.2018.01.009>.
- [22] Masahiko Onosato, Kazuaki Iwata, Development of a Virtual Manufacturing System by Integrating Product Models and Factory Models, *CIRP Annals*, Volume 42, Issue 1, 1993, Pages 475-478, ISSN 0007-8506, [https://doi.org/10.1016/S0007-8506\(07\)62489-5](https://doi.org/10.1016/S0007-8506(07)62489-5).
- [23] Chen, P. (2003). Virtual reality in design and manufacturing (Doctoral dissertation).
- [24] Adams, R. J., D. Klowden, and B. Hannaford. 2001. "Virtual Training for a Manual Assembly Task." *Haptics-e 2* (2): 1–7.
- [25] Langley, A., Lawson, G., Hermawati, S., D'cruz, M., Apold, J., Arlt, F., & Mura, K. (2016). Establishing the usability of a virtual training system for assembly operations within the automotive industry. *Human Factors and Ergonomics in Manufacturing & Service Industries*, 26(6), 667-679.
- [26] Nirit Gavish, Teresa Gutiérrez, Sabine Webel, Jorge Rodríguez, Matteo Peveri, Uli Bockholt & Franco Tecchia (2015) Evaluating virtual reality and augmented reality training for industrial maintenance and assembly tasks, *Interactive Learning Environments*, 23:6, 778-798, DOI: 10.1080/10494820.2013.815221
- [27] V. R. Kamat and J. C. Martinez, "Enabling smooth and scalable dynamic 3D visualization of discrete-event construction simulations," *Proceeding of the 2001 Winter Simulation Conference* (Cat. No.01CH37304), Arlington, VA, USA, 2001, pp. 1528-1533 vol.2, doi: 10.1109/WSC.2001.977481.
- [28] Wang, W., Wu, B., Hu, Y. M., Li, M. Y., Liu, Q. Y., & He, C. (2012). Virtual Simulation Applications in Manufacturing Process of High-Speed Trains. *Advanced Materials Research*, 622–623, 575–580. <https://doi.org/10.4028/www.scientific.net/amr.622-623.575>

- [29] Pascarelli C., Lazoi M., Papadia G., Galli V., Piarulli L. (2018) CAD-VR Integration as a Tool for Industrial Assembly Processes Validation: A Practical Application. In: De Paolis L., Bourdot P. (eds) *Augmented Reality, Virtual Reality, and Computer Graphics. AVR 2018. Lecture Notes in Computer Science*, vol 10851. Springer, Cham.  
[https://doi.org/10.1007/978-3-319-95282-6\\_32](https://doi.org/10.1007/978-3-319-95282-6_32)
- [30] S.N. Trika, P Banerjee, R.L. Kashyap, Virtual reality interfaces for feature-based computer-aided design systems, *Computer-Aided Design*, Volume 29, Issue 8, 1997, Pages 565-574, ISSN 0010-4485, [https://doi.org/10.1016/S0010-4485\(96\)00092-9](https://doi.org/10.1016/S0010-4485(96)00092-9).
- [31] Franck Ganier, Charlotte Hoareau & Jacques Tisseau (2014) Evaluation of procedural learning transfer from a virtual environment to a real situation: a case study on tank maintenance training, *Ergonomics*, 57:6, 828-843, DOI: 10.1080/00140139.2014.899628
- [32] N. S. S. Hamid, F. A. Aziz and A. Azizi, "Virtual reality applications in manufacturing system," 2014 Science and Information Conference, London, 2014, pp. 1034-1037, doi: 10.1109/SAI.2014.6918317.
- [33] Ching-Ling Fan, Jean Lee, Wen-Chih Lo, Chun-Ying Huang, Kuan-Ta Chen, and Cheng-Hsin Hsu. 2017. Fixation Prediction for 360° Video Streaming in Head-Mounted Virtual Reality. In *Proceedings of the 27th Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV'17)*. Association for Computing Machinery, New York, NY, USA, 67–72. DOI:<https://doi.org/10.1145/3083165.3083180>
- [34] Zhang, Wei & Zou, Wenjie & Yang, Fuzheng & L  v  que, Lucie & Liu, Hantao. (2019). The Effect of Spatio-temporal Inconsistency on the Subjective Quality Evaluation of Omnidirectional Videos. 4055-4059. 10.1109/ICASSP.2019.8682221.
- [35] Zulkiewicz, B. A., Boudewyns, V., Gupta, C., Kirschenbaum, A., & Lewis, M. A. (2020). Using 360-Degree Video as a Research Stimulus in Digital Health Studies: Lessons Learned. *JMIR serious games*, 8(1), e15422. <https://doi.org/10.2196/15422>
- [36] C. G. Coogan and B. He, "Brain-Computer Interface Control in a Virtual Reality Environment and Applications for the Internet of Things," in *IEEE Access*, vol. 6, pp. 10840-10849, 2018, doi: 10.1109/ACCESS.2018.2809453.
- [37] "The Shading Flow Chart" [Online]. Available:  
<https://learn.unity.com/tutorial/rendering-and-shading#5c7f8528edbc2a002053b539>

- [38] Z. Qu, X. Li, P. Guo, S. Song and X. Sun, "Virtual Maintenance System for A Certain Equipment based on Unity3D," 2019 IEEE International Conference on Power, Intelligent Computing and Systems (ICPICS), Shenyang, China, 2019, pp. 424-427, doi: 10.1109/ICPICS47731.2019.8942536. <https://ieeexplore.ieee.org/document/8942536>
- [39] "Unity UML diagram." [Online].  
Available:[https://blog.csdn.net/TW\\_345/article/details/49452451](https://blog.csdn.net/TW_345/article/details/49452451)
- [40] L.V. Radionova, A.D. Chernyshev, R.A. Lisovskiy, Interactive Educational System – Virtual Simulator "Sheet Rolling", Procedia Engineering, Volume 206, 2017, Pages 512-518, ISSN 1877-7058, <https://doi.org/10.1016/j.proeng.2017.10.509>. [Online]. Available: <https://scriptable.com/blog/oculus-go-unity-setup-quick-start>
- [41] Y. Wang, K. Ijaz and R. A. Calvo, "A software application framework for developing immersive virtual reality experiences in health domain," 2017 IEEE Life Sciences Conference (LSC), Sydney, NSW, 2017, pp. 37-30, doi: 10.1109/LSC.2017.8268135.
- [42] Y. Liu, Y. Tang and Q. Sun, "SYVR Virtual Equipment Manufacturing Simulation," 2018 International Conference on Virtual Reality and Visualization (ICVRV), Qingdao, China, 2018, pp. 139-139, doi: 10.1109/ICVRV.2018.00046.
- [43] "Rolling Stands Models." [Online]. Available:  
[https://www.forcecontrol.com/images/downloads/AppBulletins/AB-07-01A%20\(Roughing\)%20Hot%20Strip%20Rolling.pdf](https://www.forcecontrol.com/images/downloads/AppBulletins/AB-07-01A%20(Roughing)%20Hot%20Strip%20Rolling.pdf)
- [44] J.G. Lenard, ... L. Cser, in Mathematical and Physical Simulation of the Properties of Hot Rolled Products, 1999
- [45] "Bezier curve." [Online]. Available:  
<https://www.habrador.com/tutorials/interpolation/2-bezier-curve/>
- [46] AMTEC CIVS [Online]. Available: Research Areas and Projects - Center for Innovation through Visualization and Simulation (CIVS) (pnw.edu)
- [47] Manufacturing of advanced high-strength steels (AHSS) M.-C. Theyssier, in Welding and Joining of Advanced High Strength Steels (AHSS), 2015  
<https://www.sciencedirect.com/topics/engineering/roughing-mill>
- [48] L.V. Radionova, A.D. Chernyshev, R.A. Lisovskiy, Interactive Educational System – Virtual Simulator "Sheet Rolling", Procedia Engineering, Volume 206, 2017, Pages 512-518, ISSN 1877-7058, <https://doi.org/10.1016/j.proeng.2017.10.509>.

- [49] S. Wang, Z. Mao, C. Zeng, H. Gong, S. Li and B. Chen, "A new method of virtual reality based on Unity3D," 2010 18th International Conference on Geoinformatics, Beijing, 2010, pp. 1-5.  
URL: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5567608&isnumber=5567473>
- [50] Andaluz V.H. et al. (2017) Training of Tannery Processes Through Virtual Reality. In: De Paolis L., Bourdot P., Mongelli A. (eds) Augmented Reality, Virtual Reality, and Computer Graphics. AVR 2017. Lecture Notes in Computer Science, vol 10324. Springer, Cham  
[https://doi.org/10.1007/978-3-319-60922-5\\_6](https://doi.org/10.1007/978-3-319-60922-5_6)
- [51] L. Gao, Z. Gao, F. Wu, P. Wu, S. Liu and L. Liu, "Research on Real-time Data-driven Simulation Technology and Application of Digital Workshop," 2019 IEEE 8th Joint International Information Technology and Artificial Intelligence Conference (ITAIC), Chongqing, China, 2019, pp. 1743-1747, doi: 10.1109/ITAIC.2019.8785458.