

VIRTUAL REALITY SIMULATION FOR INDUSTRIAL PURPOSE

Submitted in partial fulfillment of the requirements for
the award of Bachelor of Engineering degree in
Computer Science and Engineering

by

SAI HARISH SA (Reg. No. 37110659)

K ROHAN KUMAR (Reg. No. 3611646)



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING**

SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

**Accredited with Grade "A" by NAAC
JEPPIAAR NAGAR, RAJIV GANDHI SALAI,
CHENNAI – 600 119**

APRIL – 2021



SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited with “A” grade by NAAC
Ippiaar Nagar, Rajiv Gandhi Salai, Chennai – 600 119
www.sathyabama.ac.in



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this project report is the bonafide work of **SAI HARISH SA (Reg. No. 37110659)** and **K ROHAN KUMAR (Reg. No.36110646)** who carried out the project entitled “**VIRTUAL REALITY SIMULATION FOR INDUSTRIAL PURPOSE**” under my supervision from **September 2020** to **April 2021**.

Internal Guide

Dr. S. Prince Mary, M.E., Ph.D.

Head of the Department

Submitted for Viva voce Examination held on_____

Internal Examiner

External Examiner

DECLARATION

I **SAI HARISH SA** hereby declare that the Project Report entitled “**VIRTUAL REALITY FOR INDUSTRIAL PURPOSE**” is done by me under the guidance of **Dr. S. Prince Mary, M.E., Ph.D.** Department of Computer Science and Engineering at Sathyabama Institute of Science and Technology is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Computer Science and Engineering.

DATE:

PLACE: CHENNAI

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management** of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala, M.E., Ph.D., Dean**, School of Computing, **Dr. S. Vigneswari, M.E., Ph.D.,** and **Dr. L. Lakshmanan, M.E., Ph.D., Heads of the Department** of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr. S. Prince Mary, M.E., Ph.D., Associate Professor**, for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

ABSTRACT

The analysis of the cost responsibility and the cost cause in the production process shows that design departments are responsible for up to 70% of the production costs. A large potential of savings can be made if the interface between design and assembly planning would be more efficient. The planner needs a tool which allows him to plan and simulate the real assembly under real production circumstances. The technique of VR is the ideal basis for spatial assembly and disassembly planning of products. This means different assembly and disassembly possibilities can be compared and tested in a virtual world. Which uses this feature in order to carry out assembly planning, maintenance planning and disassembly operations. This contribution reports on ongoing research activities, focusing on an innovative assembly planning system using virtual reality. Today, manufacturing industries are trying to improve their competitiveness by combining manufacturing with information technology. Virtual reality is being used in product development processes in manufacturing enterprises as a helpful technology to achieve rapid consolidation of information and decision-making through visualization and experience.

TABLE OF CONTENT

CHAPTER No.	TITLE	PAGE No.
	ABSTRACT	v
	LIST OF FIGURES	vi
1.	INTRODUCTION	8
	1.1 MACHINE ASSEMBLY AND DISMANTALING	8
	1.2 VIRTUAL REALITY	9
	1.3 USE OF C#	10
	1.4 SCRIPTING COMPONENTS IN UNITY	10
	1.5 SCRIPTING BACKENDS IN UNITY	10
	1.6 BENEFITS OF DATA-ORIENTED DESIGN	11
	1.7 FULL ADVANTAGE OF MODERN HARDWARE	12
	1.8 FUTURE OF VR	13
2	LITERATURE SURVEY	14
3.	AIM AND SCOPE	15
	3.1 AIM	15
	3.2 THE FUTURE OF VIRTUAL REALITY	15
	3.3 SCOPE	16
	3.4 ASSEMBLY PLANNING USING VIRTUAL REALITY	16
	3.5 HOW CORPORATE TRAINERS CAN USE VR	18
	3.6 APPLICATION OF VR IN INDUSTRIES	19
4	SYSTEM ANALYSIS REQUIREMENTS AND DESIGNS	20
	4.1 SYSTEM ANALYSIS	20
	4.2 SYSTEM REQUIREMENT	20
	4.3 SYSTEM CONFIGURATION	21
	4.4 DESIGN AND IMPLEMENTATION	23
5.	MODULES AND METHODOLOGY	27
	5.1 PLATFORM CREATION	27
	5.2 FIRST PERSON CONTROLLER FOR MOVEMENT AND LOOK	30
	5.3 COMBINED MODEL	32
	5.4 MACHINE LAYOUT	32
	5.5 STATE TRANSITION	34
	5.6 PROPOSED ARCHITECTURE	35
6	RESULTS AND CONCLUSIONS	37
	6.1 Static state of all machine	
	6.2 Disassembled State	
	REFERENCES	40
	APPENDIX	41

LIST OF FIGURES

FIGURE NO.	FIGURE NAME	PAGE NO.
5.1	Platform	28
5.2	FPS controller	30
5.3	Combined model	32
5.4	Overall layout	33
5.5	State transition	34
5.6	ARCHITECTURE DIAGRAM	35

CHAPTER 1 INTRODUCTION

1.1 MACHINE ASSEMBLY AND DISMANTALING

In this project, we'll be implementing a scenario where the worker can assemble or dismantle a machine in a virtual reality simulation. He/she can basically see what's inside of the machine. Using this he/she can get a really good understanding of the parts of that machine. Instead of having a theoretical knowledge, if a person had hands on experience as well then he can troubleshoot or handle it very efficiently. Since this is a simulation it can be replayed again and again. This can be very efficient. Over the last will help perform various assembling and dismantling in virtual format without the risk of breaking a real machine this system will provide the same experience as of a real machine the integrated assembly database contains the information about possible connections of the parts as well. Henceforth, the database contains information about function and degrees of freedom of the parts and assembly tools, especially when their geometry is variable. An example for such an information is the radius of an assembly crane or the needed pick space of a robot. This system supports the planner when having to take the restricted assembly room economy into account. The planer will have the possibility to assemble and disassemble a virtual product in a virtual

1.2 VIRTUAL REALITY

Virtual Reality (VR) is the use of computer technology to create a simulated environment. Unlike traditional user interfaces, VR places the user inside an experience. Instead of viewing a screen in front of them, users are immersed and able to interact with 3D worlds. By simulating as many senses as possible, such as vision, hearing, touch, even smell, the computer is transformed into a gatekeeper to this artificial world. The only limits to near-real VR experiences are the availability of content and cheap computing power.

The world of virtual reality is still new and relatively unexplored – it is a concept that is still being developed. Even though it seems very futuristic, virtual reality is a very old idea which started to develop rapidly in the beginning of the twenty-first century. Of course, it took a lot of time until it became a reality. Virtual Reality (VR) is a computer technology that creates a simulated environment. It is a realistic and immersive simulation of a three-dimensional environment which places the user inside that environment. It means that users can't see the screen of their TV in front of them and instead of that they can enjoy in a marvelous view and surreal reality with the help of their 3D virtual glasses.

As soon as you put on your virtual 3D glasses you are absorbed in that magical world, full of incredible landscapes, futuristic sensations and amazing realistic experiences. It opens a door to a parallel world. Imagine seeing a beautiful image on your computer screen. Let's say that it is an image of a very modern building, an extremely tall glass tower. It is very nice and you can really appreciate it for what it is. Now, imagine that you are wearing your 3D virtual reality glasses. You are looking at the same building but now your experience is completely different: you are standing there, in front of that magnificent building. You cross the street in order to be closer to the building, look up and decide to go around it a little bit so you can see how it looks like from the other side. The experience is unreal and it feels like you are actually there, after seeing that amazing piece of architecture up-close. You are able to view the building at different angles, walk around it or even go inside it if you want to. The possibilities are practically endless.

1.3 USE OF C#

Unity supports C#, an industry-standard language with some similarities to Java or C++. In comparison to C++, C# is easier to learn. Additionally, it's a "managed language", meaning that it automatically does the memory management for you: allocating-deal locating memory, covering memory leaks, and so on. Generally, C# is preferable to C++ if you want to make a game first, and then deal with more advanced aspects of programming later. All gameplay and interactivity developed in Unity is constructed on three fundamental building blocks: Game Objects, Components, and Variables. Any object in a game is a Game Object:

characters, lights, special effects, props—everything. Game Objects can't do anything on their own. To actually become something, you need to give a Game Object properties, which you do by adding Components. Components define and control the behavior of Game Objects they are attached to. A simple example would be the creation of a light, which involves attaching a Light Component to a Game Object (see below). Or, adding a Rigid body Component to an object to make it fall. Components have any number of editable properties, or variables that can be tweaked via the Inspector window in the Unity editor and/or via script. In the above example, some properties of the light are range, color, and intensity.

1.4 Scripting Components in Unity

Unity's built-in Components are very versatile, but you will soon find you need to go beyond what they can provide to implement your own logic. To do this, you use scripts to implement your own game logic and behavior and then add those scripts as Components to Game Objects. Each script makes its connection with the internal workings of Unity by implementing a class which derives from the built-in class called Mono Behavior.

Your script Components will allow you to do many things: trigger game events, check for collisions, and apply physics, respond to user input, and much, much more. See the Unity Scripting API for more information.

1.5 Scripting backends in Unity

.NET: Unity has used an implementation of the standard Mono runtime for scripting that natively supports C#. Unity currently ships with Visual Studio for Mac instead of MonoDevelop-Unity on macOS. On Windows, Unity ships with Visual Studio. The .NET 4.6 scripting runtime in Unity supports many of the new exciting C# features and debugging available in C# 6.0 and beyond. This also provides a great C# IDE experience to accompany the new C# features. IL2CPP: This is a Unity-developed scripting backend which you can use as an alternative to Mono when building projects for some platforms. When you choose to build a project using IL2CPP, Unity converts IL code from scripts and assemblies into C++ code, before creating a native binary file (.exe, apk, .xap, for example) for your chosen

platform. Note that IL2CPP is the only scripting backend available when building for iOS and WebGL.

1.6 Benefits of data-oriented design

The traditional GameObject-Component concept continues to work well because it's easy to understand for programmers and non-programmers alike, and easy to build intuitive UIs for. You add a Rigidbody Component to a GameObject and it will start falling, or a Light Component to a Game Object and it will emit light. And so on.

However, the Component system was written in an object-oriented framework and it creates challenges for developers when it comes to managing cache and memory in ever-evolving hardware.

Components and Game Objects are “heavy C++” objects. All Game Objects have a name. Their Components are C# wrappers on top of C++ components. This makes them easy to work with, however, it can come at a cost to performance because they potentially end up stored in an unstructured way. That C# object could be anywhere in memory. The C++ object can also be anywhere in memory. Things are not grouped together in contiguous memory. Every time anything is loaded in CPU for processing, everything has to be fetched from multiple locations. It can get slow and inefficient and therefore, require a lot of optimization workarounds.

To address these performance problems, we're rebuilding the core foundation of Unity with the high-performance, multithreaded Data-Oriented Technology Stack or DOTS (currently in Preview).

DOTS makes it possible for your game to fully utilize the latest multicore processors efficiently. It's comprised of:

- ✓ The C# Job System for running multithreaded code efficiently.
- ✓ The Entity Component System (ECS) for writing high-performance code by default.
- ✓ The Burst Compiler for producing highly optimized native code.

In DOTS, the ECS is the new Component system; what you do with a GameObject in the traditional object-oriented way, you do with an Entity in this new system. Components are still called just that. The critical difference is in the data layout. You can read more about this in the blog post “On DOTS: Entity Component System”.

1.7 Take full advantage of modern hardware

In addition to being a better way of approaching game programming for design reasons, using ECS puts you in an ideal position to leverage Unity’s C# Job System and Burst Compiler, letting you take full advantage of today’s modern hardware.

DOTS’ multithreaded systems enable you to create games that run on a variety of hardware and build richer game worlds with more elements and more complex simulations. Performing code in turn contributes to optimal thermal control and battery lifetime on players’ mobile devices. By moving from object-oriented to data-oriented design, it can be easier for you to reuse your code and for others to understand and work on it.

As some of the technology of DOTS is in Preview, it is advised that developers use it to solve a specific performance challenge in their projects, as opposed to building entire projects on it. Check out the “More Resources” section at the bottom of this page for links to key DOTS resources.

1.8 FUTURE OF VR

Virtual reality (VR) has gone beyond its gaming and entertainment roots. The most current virtual reality trends show the technology being applied across various industries, including healthcare, education, entertainment, tourism, and business institutions. But more importantly, key players in the industry, especially in the marketing sector, are still exploring ways to tap into the immense potential of VR technology.

VR technology helps businesses improve their operations. Field employees get to virtually experience life-like scenarios to accomplish tasks. Consumers, meanwhile, experience entertainment on another level and get to have more accurate material for product reviews. Surely, you can’t afford not to know what’s happening in this field if you’re into business one way or another. In this article, we present the latest virtual reality industry trends from this

technology's role in gaming and corporate LMS to its use in travel and retail, as well as how the future is shaping for this unique industry given the disruptions caused by the COVID-19 pandemic.

Some businesses have started applying VR into their training systems. In the retail landscape, for instance, VR technology allows staff to immerse themselves in a virtual environment where they need to manage long queues or shoppers crowding the store. This eliminates the need to disturb daily business operations to train the entire team.

Walmart applied this technique to prepare their employees for Black Friday shoppers. The company has partnered with VR creator Stir, which also helped train NFL players through VR technology. Walmart subjected its employees to a simulation of real-world scenarios. Employees were then asked to make choices based on what they saw in the scenes created by the computer-generated vision.

The goal of VR in training is to map scenes and prepare employees to deal with situations like the holiday rush or handling unexpected issues prevalent in the retail sector. Overall, there are 57.4 million users of VR in the US (eMarketer, 2020) and a good chunk of it is used for the retail sector (9.5 million as of 2020). However, a study predicts that the total number of VR users in the retail sector will reach 31.5 million by 2025 (Statista, 2016).

Aside from retail staff training, VR will also be used to improve and streamline customers' online shopping experience. While AR (augmented reality) in the form of AR-powered virtual fitting rooms have been around a while, VR is predicted to penetrate the industry by facilitating virtual stores (PYMNTS, 2019).

A challenge for the retail sector lies in the cost of VR technology. The use of budgeting software platforms can determine if such technology can prove beneficial to businesses. As a strategy, business owners are limiting the number of people who get to use VR headsets, while the rest would watch on a flat-screen monitor.

CHAPTER 2

LITERATURE SURVEY

Many researches have been done on VA and VM. The current researches mainly focus on product assembly information modeling, interactive operation methods, and tools modeling. Some representative product modeling methods have been proposed, such as characteristic-based method, geometric constraint-based method, and semantic-based method. The methods mentioned above determine the mode of interactive operation in virtual environment (VE), such as final position-based mode, constraint-based mode, and semantic-based mode. Based on these researches, many VA or VM systems are developed.

VADE is a typical VA system developed by Washington State University. Its assembly data are imported from CAD software using Pro/Toolkit. Users can perform assembly operations interactively with virtual hands. University of Sanford develops an immersive assembly and maintenance environment for assimilability and maintainability evaluation for large-scale product. In this environment, VICON optical tracking system is used to track the position of the user's head and hands for interactive operation.

IVAE is another VA system and developed by Shanghai Jiao Tong University in constraint-based mode. Data gloves and FOB tracker are used to control virtual hands, which interactively assemble product with virtual tools and equipment in the VE. The system is mainly aimed to deal with assembly path planning and analyzing.

Yao et al. Develop a pragmatic system to support interactive assembly planning and training in an immersive virtual environment. Recently, some virtual training systems are proposed. Xia et al. describe a new type of haptic-based virtual environment system to support assembly training, where both physics based modeling and haptic feedback are adopted to make assembly operations more realistic.

Sun et al. develop a visual simulator based on VR technology for the injection molding machine operations training. Systems mentioned above mainly focus on the interactive manipulating modes and the using of some simple tools. Nevertheless, during the assembly

or disassembly operation processes, only virtual hands are driven to manipulate virtual objects. The idea or practice of integrating a complete virtual human model into a VA or VM.

Gomes De Sá A, Zachmann G applied virtual reality (VR) for virtual prototyping (VP) to verify assembly and maintenance processes. After a review of today's business process in vehicle prototyping, we discuss CAD-VR data integration and identify new requirements for design quality. We present several new interaction paradigms so that engineers and designers can experiment naturally with the prototype. Finally, a user survey evaluates some of the paradigms and the acceptance and feasibility of virtual prototyping for our key process. The results show that VR will play an important role for VP in the near future.

Seth A, Vance JM, Oliver JH stated that a well designed assembly process should take into account various factors such as optimum assembly time and sequence, tooling and fixture requirements, ergonomics, operator safety, and accessibility, among others. Existing computer-based tools to support virtual assembly either concentrate solely on representation of the geometry of parts and fixtures and evaluation of clearances and tolerances or use simulated human mannequins to approximate human interaction in the assembly process. Virtual reality technology has the potential to support integration of natural human motions into the computer aided assembly planning environment. This would allow evaluations of an assembler's ability to manipulate and assemble parts and result in reduced time and cost for product design. This paper provides a review of the research in virtual assembly and categorizes the different approaches. Finally, critical requirements and directions for future research are presented.

Wu DL, Zhu HM, Zhen XJ, stated that Tools and equipment (T&E) play an important role in the simulation of virtual assembly, especially in the assembly process simulation and plan. The interactive operation simulation of T&E is also a challenge for interactive assembly operation because of variety in function and complexity in structure and manipulation. Based on detailed analysis of interactive operations, the functional requirement for T&E is given. Then, a unified modelling method for information expression and function realization of general T&E is represented, and the handling methods of some typical manual, semi-automatic, automatic T&E are discussed. Finally, the rear suspension and the front suspension assembly workplace operation simulation in the RA-750 car assembly line are

introduced. The result shows that the modelling and handling methods are applicable in the interactive simulation of various T&E.

CHAPTER 3

AIM AND SCOPE

3.1 Aim

Over the last few years, manufacturing companies have been facing various challenges particularly related to volatile demand and changing requirements from customers and suppliers, requiring new technological road maps and interventions in manufacturing systems. So this system will help perform various assembling and dismantling in virtual format without the risk of breaking a real machine this system will provide the same experience as of a real machine.

3.2 The future of virtual reality

You might think you've experienced VR, and you might have been pretty impressed. Particularly if you're a gamer, there are some great experiences to be had out there (or rather, in there) today But over the next few years, in VR, as in all fields of technology, we're going to see things that make what is cutting-edge today look like Space Invaders. And although the games will be amazing, the effects of this transformation will be far broader, touching on our work, education, and social lives.

Today's most popular VR applications involve taking total control of a user's senses (sight and hearing, particularly) to create a totally immersive experience that places the user in a fully virtual environment that feels pretty realistic.

Climb up something high and look down, and you're likely to get a sense of vertigo. If you see an object moving quickly towards your head, you'll feel an urge to duck out of the way.

Very soon, VR creators will extend this sensory hijacking to our other faculties – for example, touch and smell – to deepen that sense of immersion. At the same time, the devices we use to visit these virtual worlds will become cheaper and lighter, removing the friction that can currently be a barrier.

It's believed that extended reality (XR) – a term that covers virtual reality (VR), augmented reality (AR), and mixed reality (MR) – will be one of the most transformative tech trends of the next five years. It will be enabled and augmented by other tech trends, including super-fast networking that will let us experience VR as a cloud service just like we currently consume music and movies. And artificial intelligence (AI) will provide us with more personalized virtual worlds to explore, even giving us realistic virtual characters to share our experiences with.

3.3 Scope

Our main objective is to target all types of industries where we can implement our project to support various kind of machine where we can assemble different machine parts And also we can see where the faulty part is present at.

Built-In Support

1) Microsoft visual studio community 2019

Is used to do scripting in unity 3d the language we use here is c# and all the languages that unity operates with are object oriented scripting language.

2) Android and ios build support

Is used so that we can use the same project and make it as an apk file for an android or ios phone.

3) Mac build support

So it can be supported by mac os

4) web-gl build support

Unity web-gl build allows you to render 2d and 3d graphics in a web browser without the need of additional tools or plug-ins

3.4 Assembly planning using virtual reality

The productivity of assembly systems in a company is determined by the operations

planning and scheduling department. Here, the effectiveness of a Solution depends very much on the individual planner in the department. In order to increase the flexibility As well as to adapt continuously the assembly Working system to changing environmental Conditions and other influences in an economic away, it is necessary to supply the planner's with Innovative planning tools. Hence, when planning the Assembly sequence of a product, the planner has to take the product features into account, but has to consider the other assembly conditions such as Availability of assembly room, handling of parts as Well as the necessary assembly tools.

The planning results are responsible for high costs in production and assembly. Today, the majority of the precedence graphs and assembly plans are created manually. This results in a lack of information actuality, systematically planning and in taking Changing conditions into account.

The planner needs a tool which allows him to plan and simulate the real assembly under real production circumstances. The result should be a sequence of the assembly structure taking all production aspects into account. Based on the constructive structure of the product, the tool will consider restrictions such as less or more assembly room or the accessibility of the products. The tool will take especially manual assembly scenarios into account, such as the assembly of a large machine tool, or the manual assembly of e.g. automobiles and trucks. The result is the kind and sequence of the assembly operations. The tool will also have the possibility to follow the assembly in a simulative way. The use for Industrial Engineering is developing such a planning system which takes those requirements into account. This system is based on the new technology of virtual reality the technique of VR gives the possibility for three dimensional interactions with objects. These interactions can be, for example the changing of positions and orientations of objects as well as the connection of parts. This is the ideal basis for spatial assembly planning of products. This means different assembly possibilities can be compared and tested in virtual world. The basic requirement for a VR - assembly planning tool is, that the geometry of the parts are represented in the tool. In a simple case the parts have a stiff structure, parts which have got a flexible structure are more difficult to describe. In addition all the degrees of freedom must be described. The recognition of collisions of parts, this means an impact of some Parts while moving, is essential for assembly planning. This is simple for real prototypes, but it means a significant calculation expense for the

3.5 How Corporate Trainers Can Use VR

VR allows users to have fun while learning, something a classroom or slide show presentation can't always achieve. The fun aspect could help employees remember the experience and apply it in the future. At least, that's the logic employed by VR industry leaders, and it seems to be working. The possibilities that experiential VR learning gives trainers in different industries are almost endless. From oil rigs to surgery theaters, organizations utilize VR to teach employees new skills, improve their overall learning retention and develop safe work habits. Here are just a few examples of how various industry leaders use VR to improve their training process.

) **Human Resources**

Employee onboarding can set the tone for a worker's experience within the company moving forward. VR is becoming a way to make the onboarding process more fun, improving employees' outlooks on their job and the organization they work for.

Further, human resources departments can use VR to stress the seriousness of harassment and the importance of inclusion and diversity training. By engaging in simulations of potential interactions that might come up in real life, employees can learn how their behavior affects others without the risk that an actual situation might entail.

VR can also provide employers an edge in the recruitment process. Prospective employees can virtually experience what life on the job might be like. By engaging recruits both mentally and physically in a fun way, companies can attract more qualified, highly desirable candidates that otherwise might opt for a different employer.

) **Safety**

VR provides a safe space for workers to engage with and learn from volatile environments. For example, VR is helping train firefighters to resolve dangerous situations. VR allowed the firefighters to experience a high-risk environment and learn from it without putting their lives or others' safety at risk.

Another example of this is the oil industry. Simulations can train new oil rig workers on what to do if a gas leak were ever to occur, an extremely volatile situation that requires quick thinking and in-depth knowledge of advanced safety tools. The VR situation lets workers learn from their mistakes, a much better outcome than the result of their inability to cope with the event in real life.

) **Performance Productivity**

A driving factor in the growing adoption of VR is how the technology can increase a company's bottom line. Management can experiment with different work and production methods in VR and avoid the vast expenses that real-life experimentation entails.

Also, VR allows workers from different departments or corporate locations to collaborate in the virtual space across vast distances. Employees can change and complete projects quicker while reducing the resources that the company would otherwise spend on transporting personnel and hardware to new locations.

) **Embracing the Future**

Although many people still believe that VR is a passing trend that has no real implications on how corporations conduct training, I believe that it's here to stay. Employee onboarding, new skill learning and safety procedures can all be streamlined through VR to help reduce costs and increase efficiency and employee retention rates across the board.

3.6 APPLICATION OF VR IN INDUSTRIES

Digital mockup simulation platform (DMSP) is an interactive assembly and disassembly simulating system, developed by the Computer Integrated Manufacturing

Institute, Shanghai Jiaotong University. It is an evolution of the IVAE and VAPPE. Figure 13 shows the system module structure. It mainly includes four independent modules: & Interface module: This module provides user interface for DMSP, including dialogs, attribute pages, and tip boxes etc. & Simulation module: This is the core module of the system and includes four sub-modules, namely simulation objects management sub-module, CAD system interface sub module, collision scene processing sub-module, and virtual scene rendering sub-module. Simulation objects in DMSP include the product, environment models, tools, equipment, and virtual hands. The CAD system interface sub-module can transform product data from CAD software such as Pro/E, CATIA, and UG into DMSP. & Device module: this module manages the VR devices, controls their initialization, unloading, status setting and signal processing, etc. VR has reduced the time spent on revisions in product design. A study published on Advances in Computer Science Research shows that VR has had a huge impact on industrial design, providing more intuitive ways to produce new things. While the 3D modelling of product designs are nothing new, they don't communicate the actual vision of the industrial designers hence, taking up more time in back and forth revisions. With VR, the clients and manufacturers can scrutinize the design more meticulously by seeing it from a holistic point of view that can situate the future product in real-world settings. This not only saves time, but also increases the overall quality of the end product. From concept to detail design, VR enables an open environment where actual product quality can match the rendered models. This is especially crucial in industries where there is no room for error. VR can be used in the design process to give a more accurate depiction and immersive creation of 3D models. When design engineers can see their models from different angles, it gives them a better idea of how to refine and improve them. Designers can even team up to ensure the highest quality results with tools like the Active Wall system, which uses wall and floor projections that enable designers to collaborate within that unique space. And in the future, engineers from around the world can work together as if they're in the same room through VR technology, helping push down costs and time spent developing any single product.

CHAPTER 4

SYSTEM ANALYSIS REQUIREMENTS AND DESIGNS

4.1 SYSTEM ANALYSIS

Virtual reality systems make use of (in decreasing prevalence) visual, aural, and haptic displays. Use of other sensory displays has also been done. Of these the vestibular sense (the sense of balance) is the most common. In fact, it has been a very common form of display for flight simulation for decades. Olfactory display (smell) has been experimented with sparingly, and computer-controlled display of gustation (taste) is virtually nonexistent.

The most common form of vestibular display is the “motion platform.” A motion platform is basically a large surface (the platform) mounted on top of hydraulic actuators that can raise, lower, and tilt the platform. The user (typically) sits on the platform, and in the case of flight simulators within a cockpit mounted to the platform. Sometimes the visual display is also mounted on the platform; other times it is projected onto a large dome that can be seen through the windows of the cockpit. By tilting the motion platform, the pilot can then sense when the aircraft begins to pitch, yaw, or roll, and by how much.

Another style of vestibular display is the bladder-equipped chair. By inflating and deflating different portions of the chair, the user can feel acceleration and deceleration. For example, when undergoing strong acceleration, pilots will feel themselves being pushed back in their chair. To recreate this, the bladder on the back of the display seat can be filled, and thus create a similar pressure sensation on the back of the pilot. A similar effect can be implemented for sensing the effective loss of gravity while riding in a roller coaster by deflating the seat of the chair.

4.2 SYSTEM REQUIREMENT

Virtual reality technologies and their practical applications are becoming more and more popular. At the dawn of its development, the equipment was bulky, complex to install and the software setup procedure was quite complicated, not to mention a very limited number of people could get an access to it both technically and financially. However, not so long ago the situation has changed dramatically. In 2012 Oculus released their first breakthrough virtual reality head-mounted display and developer kit – the Oculus Rift.

This gear and software has opened the doors to a greater number of VR developers and enthusiasts at an affordable price, but more importantly it was becoming popular and well known amongst a wide number of users, the more questions were being asked whether home gaming computers are able to handle it. And although we've mentioned that VR technologies are more affordable nowadays, the answer is likely to be a bit disappointing for some users – their PCs might still not be ready to handle the demands of VR applications. So, the ultimate question is – what are the system requirements for custom PCs that would be able to handle VR technologies such as the Oculus Rift?

Before we get too technical, it is worth to mention that along with Oculus, there are a number of other advanced VR developments that have been announced not so long ago, such as Samsung Gear VR (for Android smartphones), HTC Vive (co-production with Valve), Sony PlayStation VR (for the PlayStation), Windows Holographic and some others. All these projects are in different stages of development and not yet fully implemented to a final consumer products. So, we will focus on the Oculus Rift as it's already available for developers and consumer version is just about to be launched in the first quarter of 2016. The HTC Vive is similarly scheduled for release in April of 2016.

4.3 SYSTEM CONFIGURATION

- J A dedicated NVIDIA GTX 600 or AMD Radeon HD 7000 series graphics card with HDMI or DVI-D output ports
- J two USB 2.0 ports (at least one of them is powered)
- J Windows 7 or Windows 8 operating system

As an additional recommendation to a minimum specs it is recommended that your dedicated graphics card should be capable of running current generation 3D games at 1080p resolution at 75fps or higher. Unfortunately, Oculus doesn't mentioned CPU and RAM minimum requirements, but we assume it shouldn't be lower than Intel Core i5/AMD A series CPUs and 8+ Gb of RAM.

- J Intel i5-4590 processor equivalent or greater
- J NVIDIA GTX 970 / AMD 290 equivalent or greater
- J 8GB+ RAM
- J Compatible HDMI 1.3 video output
- J 2 USB 3.0 ports
- J Windows 7 SP1 or newer

According to Atman Bitstock, chief architect at Oculus, "The Rift will be capable of delivering comfortable presence for nearly everyone. However, this requires the entire system working well...We believe this 'it just works' experience will be fundamental to VR's success." Bitstock also states that most likely, the Rift will be a device for desktop PCs owners, meaning that almost no current laptops meet Oculus's minimum requirements in order to get full experience. However, the new Avant line of gaming laptops from AVADirect will certainly provide enough performance to power an Oculus Rift.

There are also other exciting new developments that could change the situation and lower the performance requirements can be considered as well: VR optimized GPUs, VR displays that will be able to run at lower than native resolutions, DirectX 12 performance improvements, optimization of VR compatible games and applications towards simple and easy to render graphics.

As a piece of advice here for custom PC builders: in the case that you decide to invest money into either an upgrade of you current system or to build a new one from scratch in order to be compatible with modern VR gear, make sure that it will still be relevant in the next 6 to 12 months. If you have the graphics card that already meets the minimum requirements, try to avoid spending your money for today's top of the line GPUs, which would give you a slight performance boost at quite substantial cost that will have a reduced value by the arrival of the new, next generation VR optimized GPUs

4.4 DESIGN AND IMPLEMENTATION

The design and implementation of a distributed virtual reality (VR) platform that was developed to support the training of multiple users who must perform complex tasks in which situation assessment and critical thinking are the primary components of success. The system is fully immersive and multimodal, and users are represented as tracked, full-body figures. The system supports the manipulation of virtual objects, allowing users to act upon the environment in a natural manner. The underlying intelligent simulation component creates an interactive, responsive world in which the consequences of such actions are presented within a realistic, time-critical scenario. The focus of this work has been on the training of medical emergency-response personnel. Bio Simmer, an application of the system to training first responders to an act of bio-terrorism, has been implemented and is presented throughout the paper as a concrete example of how the underlying platform architecture supports complex training tasks. Finally, a preliminary field study was performed at the Texas Engineering Extension Service Fire Protection Training Division. The study focused on individual, rather than team, interaction with the system and was designed to gauge user acceptance of VR as a training tool. The results of this study are presented.

There are two ways to import 3D models into Unity:

-) Drag the 3D model file from your file browser straight into the Unity Project window.
-) Copy the 3D model file into the Project's Assets folder.

Select the file in the **Project** view and navigate to the **Model** tab in the Inspector window to configure import options. See documentation on [Models](#) for more information about import options.

Unity supports importing models from most popular 3D applications. For more guidance on how to import from specific 3D packages, see the following pages:

-) Maya
-) Cinema 4D
-) 3ds Max
-) Cheetah3D
-) Modo
-) Lightwave
-) Blender

Textures

You must store Textures in a folder called **Textures**, placed inside the **Assets** folder (next to the exported Mesh) within your Unity Project. This enables the Unity Editor to find the Textures and connect them to the generated Materials. For more information, see documentation on Importing Textures.

3D formats

Unity supports importing Meshes from two different types of files:

Exported 3D file formats, such as .fbx or .obj. You can export files from 3D modeling software in generic formats that can be imported and edited by a wide variety of different software.

Proprietary 3D or DCC (Digital Content Creation) application files, such as .max and .blend file formats from 3D Studio Max or Blender, for example. You can only edit proprietary files in the software that created them. Proprietary files are generally not directly editable by other software without first being converted and imported. An exception to this is Sketch Up .skip files, which both by Sketch Up and Unity can read.

Unity can import and use both types of files, and each come with their own advantages and disadvantages.

Exported 3D files

Unity can read .fox, .die (Collide), .3ds, .duff, .ob., and .skip files. Refer to your 3D modeling software documentation for information about exporting 3D files.

Advantages:

Instead of importing the whole model into Unity, you can import only the parts of the model you need.

Exported generic files are often smaller than the proprietary equivalent.

Using exported generic files encourages a modular approach (for example, using different components for collision types or interactivity).

You can import these files from software that Unity does not directly support.

Exported 3D files (.fbx, .obj) can be reimported into 3D modeling software after exporting, to ensure that all of the information has been exported correctly.

Disadvantages:

Models must be re-exported manually if changes are made to the original file. Extra care must be taken to keep track of versions between the source file and the files imported into Unity. Proprietary 3D application files Unity can import proprietary files from the following DCC software: Max, Maya, Blender, Cinema4D, Houdini, and Lightwave & Cheetah3D. Files imported this way are converted into .fbx files by Unity during the import process.

Advantages:

Updates made to the original model are automatically imported into Unity. This is initially simple - but it can become more complex later in development.

Disadvantages:

A licensed copy of the software used must be installed on all machines using the Unity project.

Software versions should be the same on each machine using the Unity project. Using a different software version can cause errors or unexpected behavior when importing 3D models.

Files can become bloated with unnecessary data.

Big files can slow down Unity project imports or Asset re-imports, because you have to run the DCC software you use as a background process when you import the model into Unity.

Unity exports proprietary files to .fbx internally, during the import process. This makes it difficult to verify the .fbx data and troubleshoot problems.

Note: Assets saved as .ma, .mb, .max, .c4d, or .blend files fail to import unless you have the corresponding DCC software installed in your computer. This means that everybody working on your Unity project must have the correct software installed. For example, if you use Maya to create *ExampleModel.mb* and copy it into your project, anyone else opening that project also needs to have Maya installed on their computer.

Models

Model files that are placed in the Assets folder in your Unity project are automatically imported and stored as Unity Assets.

A model file can contain a 3D model, such as a character, a building, or a piece of furniture. The model is imported as multiple Assets. In the Project window, the main imported object is a model Prefab. Usually there are also several Mesh objects that are referenced by the model Prefab.

A model file can also contain animation data, which can be used to animate this model or other models. The animation data is imported as one or more Animation Clips.

Debugging C# code in Unity

Using a debugger allows you to inspect your source code while your application or game is running. Unity supports debugging of C# code using the following code editors:

-) Visual Studio (with the Visual Studio Tools for Unity plug-in)
-) Visual Studio for Mac
-) JetBrains Rider
-) Visual Studio Code

Although these code editors vary slightly in the debugger features they support, all provide

If Visual Studio for Mac is already installed on your computer, use its Extension Manager to locate and install the Visual Studio Tools for Unity plug-in.

Jet Brains Rider

The default installation of JetBrains Rider can debug code in Unity on Windows or Mac. Please visit the [Jet Brains website](#) to install it.

VS Code

VS Code requires you to install an extension to debug code in Unity. Please follow the [instructions](#) specific to this extension to install it.

CHAPTER 5

MODULES AND METHODOLOGY

This system is basically divided into 4 parts each part is crucial for the virtual mechanism to work efficiently.

3.1 Platform creation

3.2 First person controller for movement and look

3.3 Combined model

3.4 Machine layout

3.5 Adding texture

Main elements which we can find in Unity3D:-

Project Window: this area shows the files and folders of our project. The only folder we'll see in our new scene is the Assets folder, which is created automatically and it's where we'll place all the game assets (3D models, scripts, audio files, images, etc.).

Hierarchy Window: shows all the game objects that are present in our scene. By default, Unity creates a camera and a directional light.

Scene View: shows the 3D view of your game scene. All the objects that you create in the Hierarchy Window will appear here in their corresponding X, Y, Z positions, and by using different "gizmos" or tools you can move, rotate and scale these objects around.

Game View: this view shows what the game actually looks like. In this case, it shows whatever the camera (which was created by default) is looking at.

5.1 Platform creation

- We are making a machine assembly line where all the virtual machine will be placed.
- First, you have to open the unity3D project. Let's see the various objects one by one.
- Select the Game Object menu in the menu bar. The Game Object Menu has more objects to create a game. Select a 3D object and pick the cube option.

- Select the rotation tool, it helps to rotate your plane object. It also rotates all the objects in your game.
- Go to the hierarchy view, drag the platform, and drop the platform option cylinder after dropping platform it will get placed in the 3 axis plane.

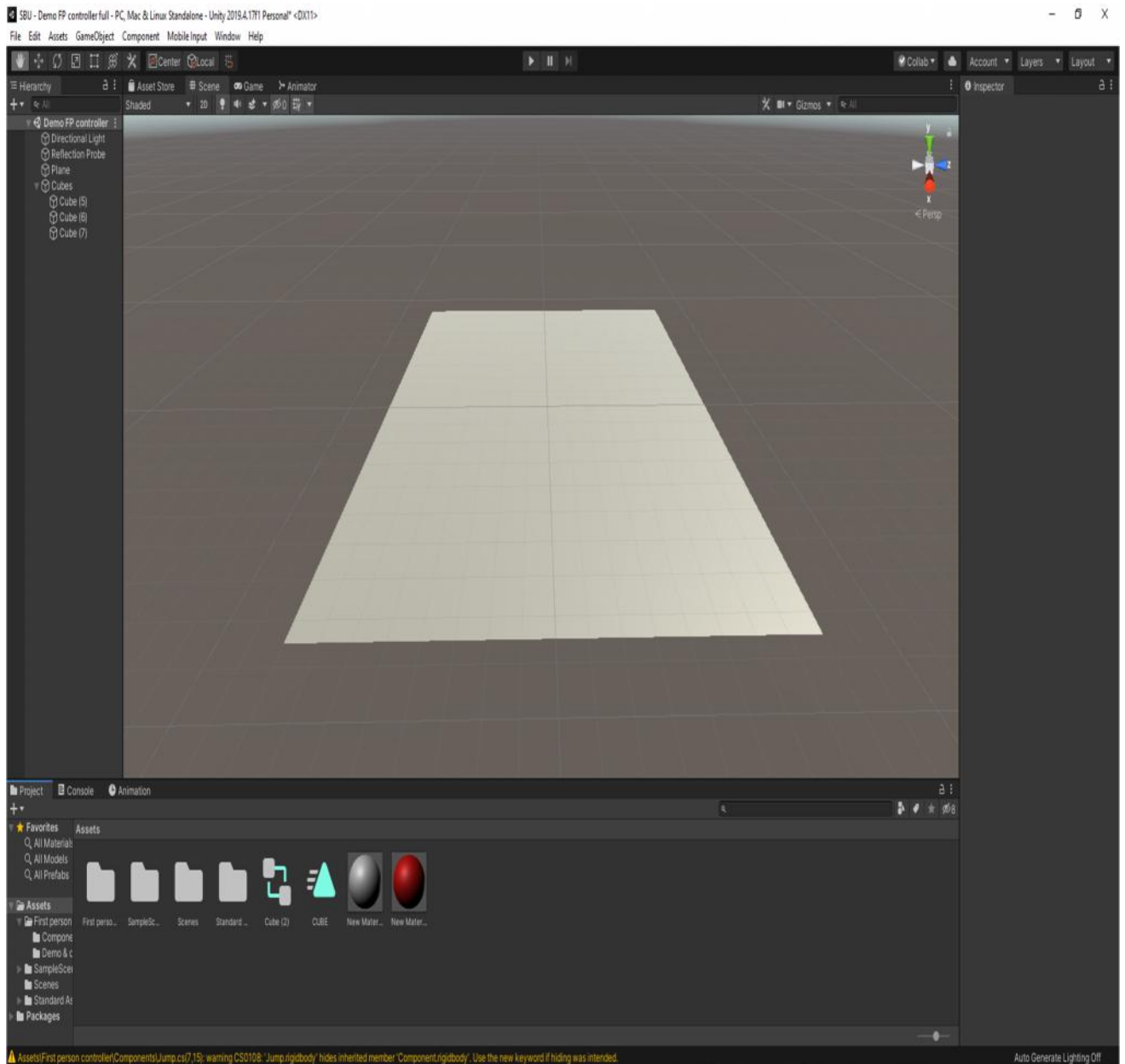


Fig5.1: Platform

5.2. First person controller for movement and look

FPS controller has two function

Movement

Look

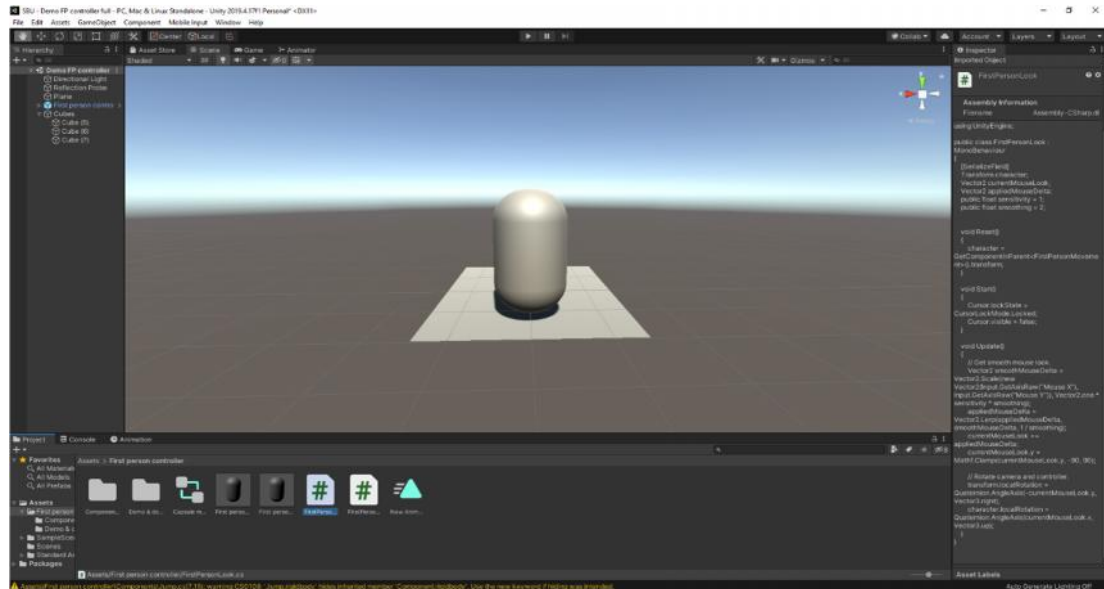


Fig5.2: First person controller for movement and look

Unity 3D FPS Controller FPS (or First-Person Shooter) is a type of game where the main character is controlled from a first-person perspective. With usual controls being W,A,S,D to walk, Mouse Look to look around, Space to jump and Left Shift to sprint, the player is able to freely move around the level. Create a new Game Object (GameObject -> Create Empty) and name it "FPSPlayer" Create new Capsule (GameObject -> 3D Object -> Capsule) and move it inside "FPSPlayer" Object Remove Capsule Collider component from Capsule and change its position to (0, 1, 0) Move Main Camera inside "FPSPlayer" Object and change its position to (0, 1.64, 0).

5.3 Combined model

Combined model is the combined state of all the above model which includes all the machine which are present in the system



Fig5.3: Combined model

5.4 Machine layout

In machine layout model all the machine will start assembling and disassembling and the person will be able to see which is the faulty part and can replace it with the new functional part.

Manufacturing setting is defined as the determination of relative locations for, and allocation of, the available space among a given number of workstations. Although most facility layout solutions have, in the past, focused on minimizing the amount of transportation, the effect of a given layout design on the production function of a manufacturing system is much more than just the cost of material handling. While material handling cost remains critical, shorter cycle times have become much more important in today's manufacturing systems.



Fig5.4: overall layout

5.5 State transition

All the animated parts will be present in the state transition layout to check the activity of the animations which are performing.

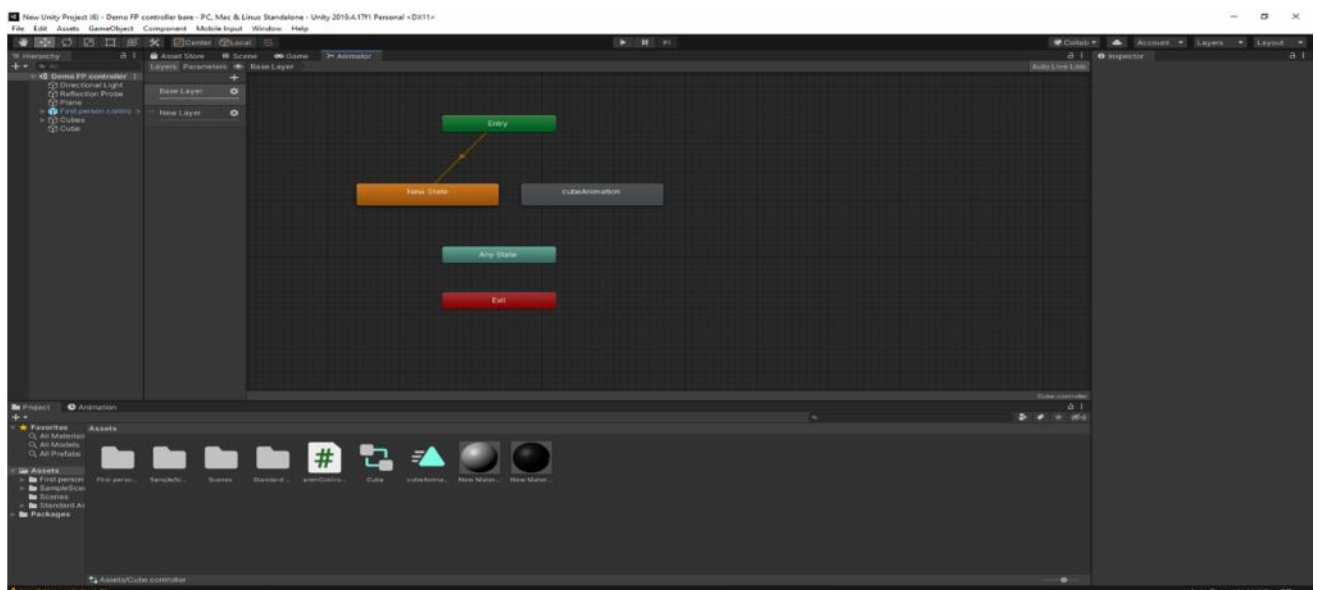


Fig5.5: State transition

5.6 PROPOSED ARCHITECTURE

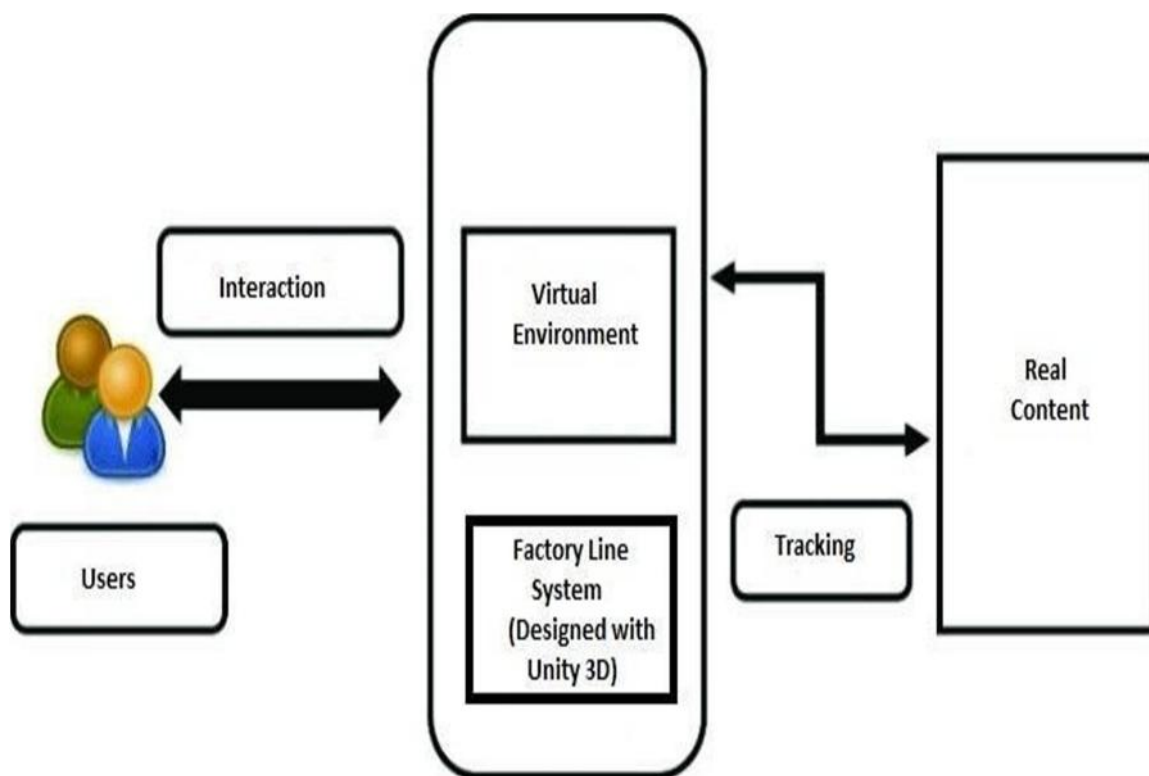


Fig5.6: ARCHITECTURE DIAGRAM

While virtual reality offers numerous benefits to both the client and the design team, it does come with its limitations. Currently, virtual reality is primarily used in the design process to explain a design to clients. Due to the significant amount of data that virtual reality requires, designers must be far along enough in the creative process before a client can virtually tour a project.

Right now, VR is really useful for helping clients understand a project that's pretty well cooked. But as designers, we want to be able to use it like the tasting spoon as we're actually doing the cooking.

Despite this current limitation, client satisfaction is key to a design project's success. And virtual reality can help ensure client satisfaction by providing the client with the opportunity to experience the design process and easily provide feedback. This in turn makes the design team's job much easier.

Virtual reality gives the designers a clear direction of what the client wants and eliminates confusion, which improves the overall efficiency of the design process. It also allows the designers to determine if a design works the way they want it to before presenting to the client. And because virtual reality is so immersive, it makes detecting errors and mistakes far less challenging.

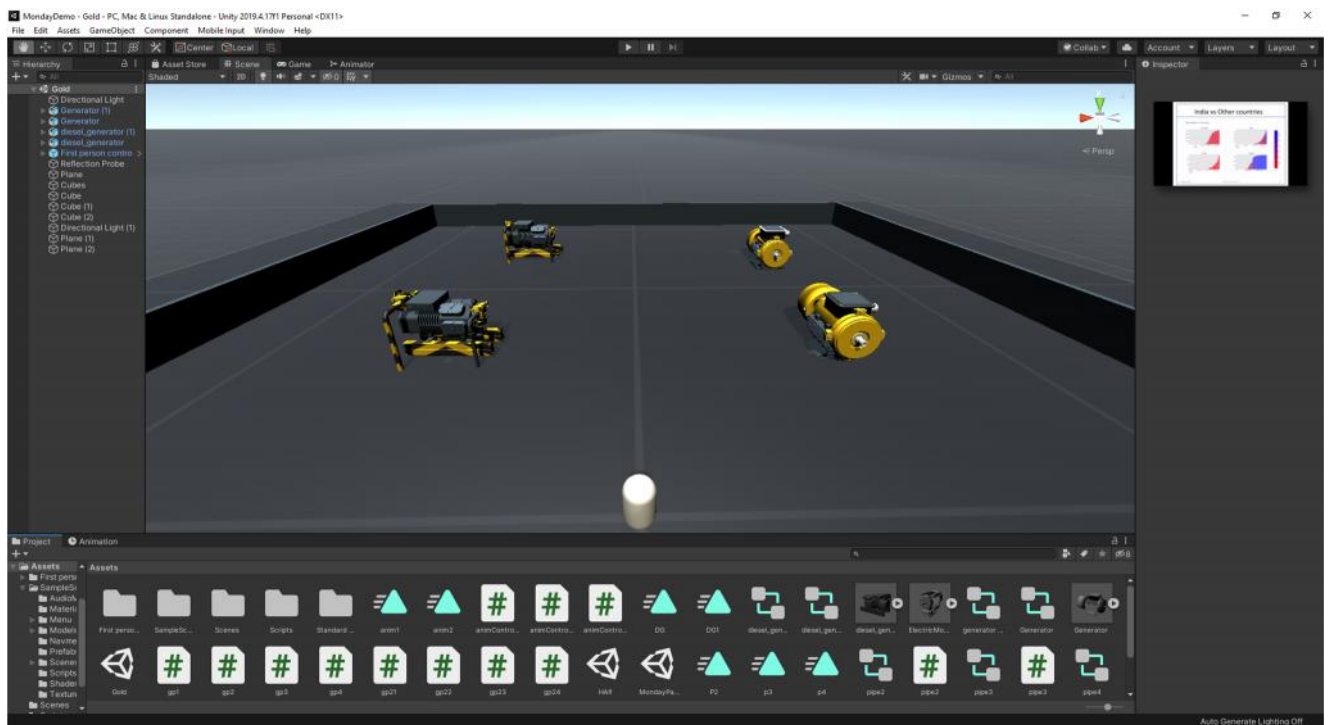
Designers can rely on virtual reality to make the revision process run smoother since the client can pinpoint precisely which changes they want made. Finally, design teams can use virtual reality to enhance their collaboration and internal communication, which is just as important to the design process as the client-designer relationship.

In short, virtual reality in interaction design evens the playing field between designer and client. Virtual reality allows both parties to be on the same page and experience the design process in a highly immersive, inclusive manner. This helps the design team and client establish effective communication throughout the project, thereby increasing the likelihood of a greater outcome.

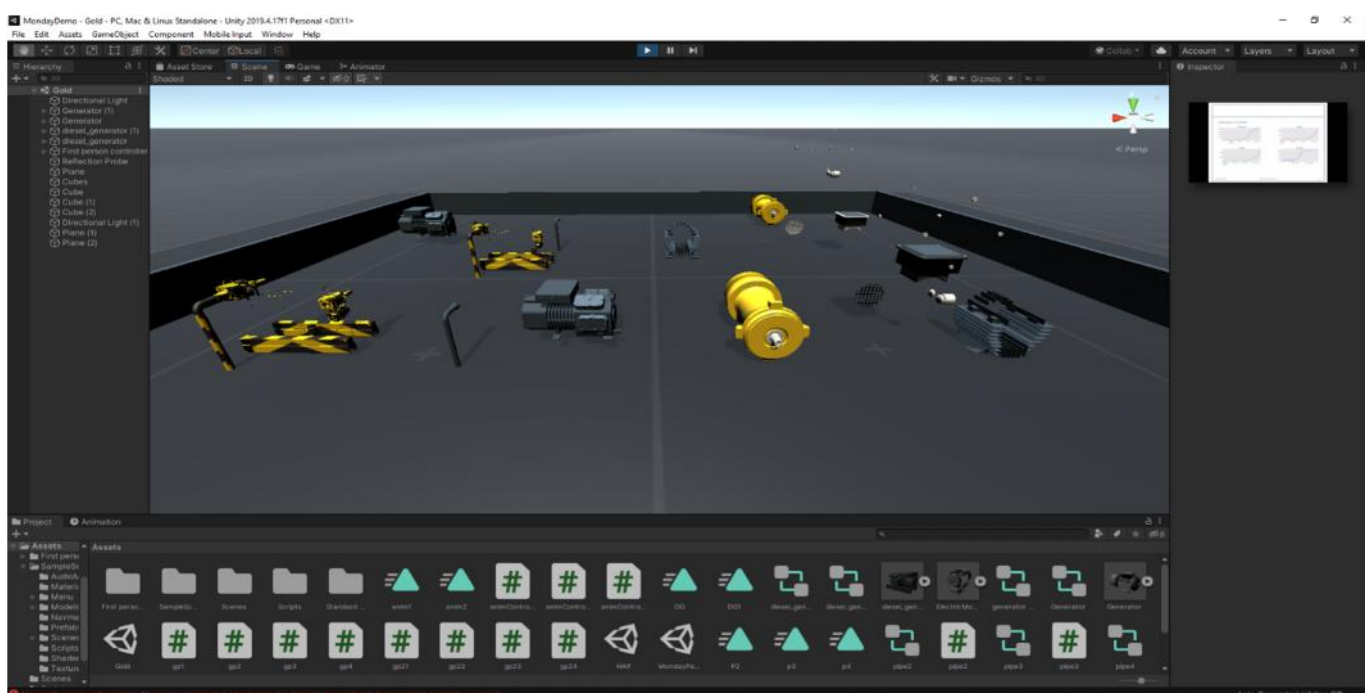
CHAPTER 6

RESULT AND CONCLUSION

6.1 Static state of all machine



6.2 Disassembled State



CONCLUSION

Radio, television, Internet, Web, and Mobile have all seen this progression play out and VR and AR will likely follow that model. At the end we will have a complete model of machine and assembly line to place machine our future machine. The new AR/VR ecosystem might not happen in near future but it is the future. The next big thing in marketing is sure to make existing marketing and advertising business models and replace them with new ones. Finally, an assembly simulation for automobile engine connecting rod caps is implemented and a user survey is carried out. The survey result demonstrates the feasibility and validity of the proposed model and method.

REFERENCES

- [1]. Burdea, G. Haptic feedback for virtual reality (Special issue on virtual prototyping). *Int J Des Innov Res* 2000; 2: 17–29.
- [2]. Boothroyd, G, Dewhurst, P. *Product design for assembly*. New York: McGraw-Hill, 1989.
- [3]. Boothroyd, G, Dewhurst, P, Knight, WA. *Product design for manufacture and assembly*. 3rd ed. Boca Raton, FL: CRC Press, 1994, 712 pp.
- [4]. Çakir, A. *Handbook of virtual environments: design, implementation, and applications*, second edition, edited by Kelly S Hale and Kay M Stanney, CRC Press, 2014. *Behav Inform Technol* 2015; 34: 752–754
- [5]. Maropoulos, PG, Ceglarek, D. Design verification and validation in product lifecycle. *CIRP Ann: Manuf Techn* 2010; 59: 740–759
- [6]. Ong, SK, Nee, AYC. *Virtual reality and augmented reality applications in manufacturing*. London: Springer-Verlag, 2004, 388 pp.
- [7]. Seth, A, Vance, J, Oliver, J. Virtual reality for assembly methods prototyping: a review. *Virtual Real* 2011; 15: 5–20.
- [8]. Wang, ZB, Ng, LX, Ong, SK. Assembly planning and evaluation in an augmented reality environment. *Int J Prod Res* 2013; 51: 7388–7404.
- [9]. Yusof, Y, Latif, K. Survey on computer-aided process planning. *Int J Adv Manuf Tech* 2014; 75: 77–89.

APPENDIX

SOURCE CODE FOR PLATFORM

```
using System;
using UnityEngine;
using UnityStandardAssets.CrossPlatformInput;
namespace UnityStandardAssets._2D
{
    [RequireComponent(typeof (PlatformerCharacter2D))]
    public class Platformer2DUserControl : MonoBehaviour
    {
        private PlatformerCharacter2D m_Character;
        private bool m_Jump;
        private void Awake()
        {
            m_Character = GetComponent<PlatformerCharacter2D>();
        }
        private void Update()
        {
            if (!m_Jump)
            {
                // Read the jump input in Update so button presses aren't
                missed.
                m_Jump =
                CrossPlatformInputManager.GetButtonDown("Jump");
            }
        }
        private void FixedUpdate()
        {
            // Read the inputs.
            bool crouch = Input.GetKey(KeyCode.LeftControl);
            float h = CrossPlatformInputManager.GetAxis("Horizontal");
```

```

        // Pass all parameters to the character control script.
        m_Character.Move(h, crouch, m_Jump);
        m_Jump = false;
    }
}
}

```

SOURCE CODE FOR MOTION

```

using UnityEngine;

public class FirstPersonMovement : MonoBehaviour
{
    Public float speed = 5;
    Vector2 velocity;
    Void FixedUpdate()
    {
        velocity.y = Input.GetAxis("Vertical") * speed * Time.deltaTime;
        velocity.x = Input.GetAxis("Horizontal") * speed * Time.deltaTime;
        transform.Translate(velocity.x, 0, velocity.y);
    }
}

```

SOURCE CODE FOR LOOK

```

using UnityEngine;

public class FirstPersonLook : MonoBehaviour
{
    [SerializeField]
    Transform character;
    Vector2 currentMouseLook;
    Vector2 appliedMouseDelta;
    public float sensitivity = 1;
    public float smoothing = 2;
    void Reset()
    {

```

- {

```

        character = GetComponentInParent<FirstPersonMovement>().transform;
    }
    void Start()
    {
        Cursor.lockState = CursorLockMode.Locked;
        Cursor.visible = false;
    }
    void Update()
    {
        // Get smooth mouse look.
        Vector2 smoothMouseDelta = Vector2.Scale(new
Vector2(Input.GetAxisRaw("Mouse X"), Input.GetAxisRaw("Mouse Y")),
Vector2.one * sensitivity * smoothing);
        appliedMouseDelta = Vector2.Lerp(appliedMouseDelta,
smoothMouseDelta, 1 / smoothing);
        currentMouseLook += appliedMouseDelta;
        currentMouseLook.y = Mathf.Clamp(currentMouseLook.y, -90, 90);
        // Rotate camera and controller.
        transform.localRotation = Quaternion.AngleAxis(-currentMouseLook.y,
Vector3.right);
        character.localRotation = Quaternion.AngleAxis(currentMouseLook.x,
Vector3.up);
    }
}

```

SOURCE CODE FOR ANIMATION

```

using UnityEngine;
using System.Collections;

public class gp1 : MonoBehaviour
{

    public Animator anim;

```

```

// Use this for initialization
void Start()
{
    anim = GetComponent<Animator>();
}

// Update is called once per frame
void Update()
{
    if (Input.GetKeyDown("6"))
    {
        anim.Play("2GP1");
    }
    else if (Input.GetKeyDown("4"))
    {
        anim.GetComponent<Animator>().enabled = false;
    }
    else if (Input.GetKeyUp("4"))
    {
        anim.GetComponent<Animator>().enabled = false;
    }
    else if (Input.GetKeyDown("5"))
    {
        anim.GetComponent<Animator>().enabled = true;
    }
}
}

```

