Office Collaboration using Metaverse

Unveiling a New Dimension of Virtual Experiences and Applications

Pavuluri Prudhvi Krishna

P. N. B. S. Eswar

Office Collaboration using Metaverse

Unveiling a New Dimension of Virtual Experiences and Applications

Project submitted to the

SRM University – AP, Andhra Pradesh

for the award of the degree

of

Bachelor of Technology

by

Pavuluri Prudhvi Krishna

P. N. B. S. Eswar

Under the Guidance of

Prof. Dr. V. Udaya Shankar

DEPARTMENT OF ELECTRONICS AND

COMMUNICATION ENGINEERING

SRM UNIVERSITY - AP, ANDHRA PRADESH

May 2024

CERTIFICATE OF APPROVAL

Certified that the project report entitled **Office Collaboration using Metaverse Unveiling a New Dimension of Virtual Experiences and Applications**, submitted by

1. Pavuluri Prudhvi Krishna (AP20110020021)
2. P. N. B. S. Eswar (AP20110010011)

to the SRM University - AP, Andhra Pradesh, for the award of the degree Bachelor of Technology has been accepted by the examiners and that the student(s) has (have) successfully completed the viva-voce examination held today.

(Examiner) (Examiner)

(Examiner) (Examiner)

(Supervisor)

Head of the Department,
Department of ECE

Date:

Place:

CERTIFICATE

This is to certify that the report entitled **Office Collaboration using Metaverse Unveiling a New Dimension of Virtual Experiences and Applications**, submitted by

1. Pavuluri Prudhvi Krishna (AP20110020021)
2. P. N. B. S. Eswar (AP20110010011)

to the SRM University - AP, Andhra Pradesh, is a record of bona fide work carried out under my supervision and is worthy of consideration for the award of the degree of Bachelor of Technology of the Institute.

<Dr. V. Udaya Shankar>

(Supervisor)
Assistant Professor
Department of Electronics and Communication Engineering
SRM University – AP,
Andhra Pradesh – 522502
INDIA

Date:

Place:

Contents

Title Page i

Approvalcert iii

Certificate v

Contents vii

List of Abbreviations ix

List of Figures xiii

List of Tables xv

Abstract xix

1 Introduction 1

1.1 Motivation 4
1.2 A Layout of the Report.. 4
1.3 Problem Statement 5
1.4 Objectives of the project 5

2 Background Theory 6

2.1 Literature Survey 7
2.2 Tools and Components... 8
2.3 Block Diagram 14
2.4 Methodology for Data Collection and Analysis 14

3 Implementation 15

3.1 Setting up an Omniverse Development Environment 16
3.2 Setting up the Oculus Environment 38
3.3 Selection of meeting platforms 40
3.4 Setup and Configuration of meeting Platforms 43
3.5 Conducting Virtual Meetings 45

4 Results 47
4.1 Office Collaboration using Nvidia Designs 48
4.2 Comparison of User Experiences 49
4.3 Comparison of Features and Functionality 50
4.4 Comparison of Performance and Stability 51
4.5 Comparison of Integration and Compatibility 52
4.6 Comparison of Cost and Accessibility 53
4.7 Comparison of Security and Privacy 54

5 Conclusion 55
5.1 Conclusion 56
5.2 Future Trends 56
5.3 Future Scope 57

A Appendix: Hardware and Software Specifications 58

References 61

List of Abbreviations

AR	Augmented Reality
VR	Virtual Reality
MR	Mixed Reality
XR	Extended Reality
GUI	Graphical User Interface
3D	Three Dimensional
AI	Artificial Intelligence
IoT	Internet of Things

List of Figures

2.3 Block diagram of Implementation …………………………………………………… 14

3.1 Implementation picture in Visual Studio ……………………………………………17
3.111 Over View in Visual Studio……………………………………………………….…17
3.112 Omniverse Kit in Visual Studio……………………………………………...……19
3.113 Extension pack in Visual Studio………………………………………….……20
3.114 my_name.my_app.kit……………………………………………………………...22
3.115 my_name.my_app.kit ………………………………………………...…...……22
3.116 View Port Image – 1 ………………………………………………….................23
3.117 View Port Image – 2……………………………………….…………………………23
3.118 KIT Debug Settings…………………………………………………………..………25
3.119 USD in Kit…………………………………………………………………………26
3.120 My custom app…………………………………………….………………………30

3.121 My custom app workspace………………………………………………...….33
3.122 Layout – 1………………………………………………………………….………34
3.123 Layout – 2…………………………………………………………….…………36

3.2 Oculus Meta Quest – 2……………………………………………….………….38

3.31 Image for Meet in VR……………………………………………..…………. 41
3.32 Google Meet Conference…………………………………………….…………. 42
3.33 Zoom Video Conferencing……………………………………...………… 43

4.11 Office Collaboration Design – 1…………………………………….………… 48
4.12 Office Collaboration Design – 2 ………………………………………….……. 48

List of Tables

4.2 Comparisons of User Experiences ……………………………………….…………… 49

4.3 Comparisons of Features and Functionality ………………………………...………. 50

4.4 Comparisons of Performance and Stability ………………………………..……….. 51

4.5 Comparisons of Integration and Compatibility ……………………………..………. 52

4.6 Comparisons of Cost and Accessibility …………………………………….……….. 53

4.7 Comparisons of Security and Privacy …………………………...…...………..…. 54

Abstract

The metaverse, a conceptual space where digital and physical realities, has emerged as a transformative paradigm in the realm of virtual experiences and applications. This study delves into the multifaceted dimensions of the metaverse, aiming to unravel its underlying technologies, implications, and diverse applications across various domains. The research delves into the practical applications of the metaverse in different industries. It scrutinizes how businesses leverage the metaverse for virtual meetings, collaborative workspaces, and immersive customer experiences. Additionally, the role of the metaverse in healthcare, education, and training, exploring its potential to revolutionize remote learning, medical simulations, and professional development may grow in future. Office collaboration using the metaverse is a transformative approach to teamwork that leverages virtual environments to enhance communication, creativity, and productivity. By immersing team members in a shared digital space, the metaverse enables seamless collaboration regardless of physical location, fostering a sense of presence and connection.

Keywords: Metaverse, VR, Office Collaboration, Nvidia Omniverse, Meetings, 3D.

CHAPTER 1

Introduction

Metaverse is based on two critical technologies: virtual Reality (VR) and augmented reality (AR), which provide access to the three-dimensional online environment via dedicated headsets and other devices connected to computers or game consoles. Artificial intelligence aids in the creation of a virtual version of each user, known as an 'avatar', who is the primary actor in the metaverse and is also utilized for seamless communication in conjunction with Internet of Things technologies. Oculus plays an important part in the "metaverse," which is a linked virtual reality realm where users may communicate, socialize, and engage in a variety of digital experiences.

It contributes to the metaverse by offering a platform for users to access and participate in a wide range of virtual experiences, social interactions, entertainment, and productivity applications. The concept of the Metaverse, along with its associated hardware and software components like sensors, glasses, and 3D object design, has a rich history spanning various stages and phases. Continuous efforts by researchers, designers, and developers aim to enhance its services, attracting more users to its fantasy world. Despite the increasing popularity, challenges such as privacy concerns persist, urging researchers to delve deeper into understanding and improving the Metaverse. Additionally, this review delves into the definitions, characteristics, structure, and practical uses of the Metaverse.

THE METAVERSE CONCEPT: Metaverse also called “WEB 3.0”.

Web 1.0 is said to be WWW (World Wide Web) and Web 2.0 is said to be social media. The metaverse concept has evolved from science fiction to a tangible, albeit nascent, reality. Coined by Neal Stephenson in his 1992 science fiction novel "Snow Crash," the metaverse was initially envisioned as a virtual reality space where users could interact with each other and digital environments. Fast forward to the present, and the metaverse has transcended its fictional origins, propelled by advancements in technologies such as virtual reality (VR), augmented reality (AR), blockchain, and artificial intelligence (AI). The metaverse is not merely a speculative concept but an unfolding phenomenon with real-world applications that span across diverse industries. The Metaverse relies on key technologies like Virtual Reality (VR) and Augmented Reality (AR), enabling users to access a three-dimensional online environment using specialized headsets and connected devices. Artificial Intelligence is utilized to generate virtual representations of users, known as 'avatars,' who play a central role within the Metaverse and facilitate seamless communication. Additionally, Internet of Things (IoT) technology contributes to enhancing connectivity and interaction within the Metaverse. The Metaverse refers to a communal virtual area that includes interconnected virtual worlds, augmented reality environments, and immersive experiences.

It is a digital realm in which people can interact with one another and digital items in real-time, usually via avatars or digital versions of themselves. The Metaverse concept encompasses many technologies, such as virtual reality, augmented reality, artificial intelligence, and the Internet of Things, which allow people to explore, create, and engage in diverse activities within this virtual domain. It is envisioned as a next-generation internet, offering boundless possibilities for social interaction, entertainment, education, commerce, and much more.

1.1 Motivation

In the face of pressing global challenges such as increasing infrastructure costs, depletion of finite resources, and escalating pollution levels, there is an urgent need for innovative solutions that not only address these issues but also foster sustainable development and community well-being. Our research endeavors to tackle these multifaceted challenges head-on by exploring the intersection of environmental sustainability, shared infrastructure, and technological advancements, with a focus on promoting education, training, and healthcare applications. The motivation behind our research stems from a profound commitment to environmental stewardship and the recognition of the critical role that infrastructure plays in shaping the future of our planet. As infrastructure costs continue to rise, and traditional energy sources approach exhaustion, it is imperative to seek alternative approaches that prioritize efficiency, conservation, and renewable resources. By advocating for shared infrastructure models and leveraging technological innovations, we aim to mitigate the environmental impact of infrastructure development while maximizing its societal benefits.

By reimagining the way we conceptualize and utilize infrastructure, we aspire to create more inclusive, resilient, and livable environments that enhance the quality of life for all. Education, training, and healthcare are fundamental pillars of societal progress, yet access to these essential services remains unequal and fragmented. By harnessing the power of connectivity, virtual interaction, and community building, we endeavor to bridge these gaps and democratize access to knowledge, skills, and healthcare resources. Through innovative applications of technology, such as virtual reality and telemedicine, we aim to revolutionize how education is delivered, how skills are acquired, and how healthcare is accessed, particularly in underserved communities.

1.2 A Layout of the Report

The report is structured in the following way:

- Chapter 1 contains Introduction
- Chapter 2 contains Background Theory
- Chapter 3 contains Implementation
- Chapter 4 contains Result
- Chapter 5 contains Conclusion

1.3 Problem Statement

The escalating costs of infrastructure development, coupled with the depletion of finite fuel sources, the proliferation of pollution, exacerbated traffic congestion, and the rampant spread of land pollution, pose significant challenges to sustainable urban development and environmental stewardship. These pressing issues demand urgent attention and innovative solutions to mitigate their adverse impacts on societal well-being, economic growth, and ecological integrity.

1.4 Objectives of the project

The main objectives of the project are:

1. To analyze the user experience of each AR/VR platform in a virtual office environment.
2. To compare the features and functionality offered by each platform for collaborative work.
3. To assess the performance and stability of the platforms during virtual meetings.
4. To evaluate the integration and compatibility of the platforms with other tools and software.
5. To compare the cost and accessibility of each platform for users.
6. To assess the security and privacy features of each platform.

CHAPTER 2

Background Theory

2.1 Literature Survey

This comprehensive review examines the intersection of the metaverse and office collaboration, exploring the latest advancements, challenges, and opportunities in this emerging field. It delves into the potential of the metaverse to transform traditional office environments, enabling remote collaboration, virtual meetings, and immersive teamwork experiences. This research investigates the role of virtual reality (VR) and augmented reality (AR) technologies in enhancing office collaboration. It discusses various VR/AR platforms and applications designed to facilitate team communication, task coordination, and knowledge sharing in virtual office environments.

Focusing on the future of work, this research explores the potential of metaverse-based office collaboration. It examines the implications of transitioning from traditional office settings to virtual workspaces, highlighting the benefits of increased flexibility, productivity, and inclusivity offered by the metaverse. Addressing both challenges and opportunities, this paper provides a critical analysis of metaverse office collaboration. It identifies technical, social, and organizational barriers to adoption and proposes strategies for overcoming these obstacles to unlock the full potential of the metaverse in the workplace. Focusing on user experience (UX) design, this paper explores best practices for developing metaverse office collaboration tools. It discusses the importance of intuitive interfaces, seamless interactions, and accessibility features in enhancing user engagement and satisfaction within virtual office environments.

2.2 Tools and Components

A) Hardware description
B) Software description

A) Hardware description

The Oculus VR headset, with its immersive capabilities, and the high-performance i9 processor coupled with a 6GB RAM system form the essential hardware foundation for our metaverse project on office collaboration. The Oculus headset facilitates virtual presence and interaction, while the powerful i9 processor and ample RAM ensure smooth rendering of virtual environments and seamless multitasking, enabling an engaging and effective collaborative experience in the metaverse.

a. Oculus (AR /VR Headsets)
b. System with i9 processor and graphic card (6gb)

B) Software description

The software stack for our metaverse project on office collaboration comprises a strategic selection of tools and platforms aimed at enabling seamless integration, immersive experiences, and efficient collaboration. Key software components include Amazon Web Services (AWS), NVIDIA Omniverse, and Python (Colab /Jupiter Notebooks).

a. Nvidia Omniverse Platform
b. AWS (Amazon Web Services)
c. Python (Colab /Jupiter Notebooks)

1. **Nvidia Omniverse Platform**:

Omniverse refers to NVIDIA Omniverse, a platform developed by NVIDIA, a technology company known for its graphics processing units (GPUs) and artificial intelligence (AI) technologies. NVIDIA Omniverse is an ambitious and cutting-edge platform designed to revolutionize the way content is created, simulated, and experienced in the fields of 3D content creation, virtual production, and collaborative workflows. NVIDIA Omniverse and the metaverse are related in the sense that both concepts revolve around creating immersive, interconnected, and collaborative virtual experiences. However, it's essential to understand that NVIDIA Omniverse and the metaverse are not the same thing.

2.2 Tools and Components

Omniverse is a platform for collaboration and simulation

OMNIVERSE FOR EVERYONE

i. **Omniverse for Individuals**:
   1. It is free for individuals
   2. It supports public forums and training videos
   3. It has only nucleus work station
   4. Use of all connectors including beta

ii. **Omniverse Enterprise**:
   1. It has a license per creator
   2. Full enterprise support and training videos
   3. Use of all LTS connectors
   4. Nucleus enterprise [Private Cloud]
   5. Omniverse create [License per creator]
   6. Omniverse View [Unlimited]

Omniverse has five main components:

ii. **NUCLEUS**: Nucleus is how we manage the data between all of these applications. It manages the delta of change between the different users.

iii. **CONNECT**: Connect is how we connect to the third-party products. So, as we build more and more connectors, more and more third-party tools become live on the Omniverse platform.

iv. **KIT**: Kit is a toolkit that's on the platform that allows developers to build their own custom applications, or for advanced users to enhance the platform and customize it for their own workflows.

v. **SIMULATION**: Simulation tools are used and famous for real time simulating fire, water, smoke, etc.

vi. **RTX RENDER**: It was so advanced; it applies multi-GPU technology using RTX for real time photorealistic rendering.

2. AWS (Amazon Web Services)

Amazon Web Services (AWS) is a comprehensive and widely-used cloud computing platform offered by Amazon.com. It provides a range of cloud services, including computing power, storage solutions, networking capabilities, databases, machine learning, artificial intelligence, analytics, and more. AWS offers scalable, reliable, and secure infrastructure services, allowing businesses and organizations to build, deploy, and manage applications and services with ease.

Key uses and services offered by AWS include:

1. **Compute**: AWS offers various compute services, including Amazon Elastic Compute Cloud (EC2) for scalable virtual servers, AWS Lambda for serverless computing, and Amazon Elastic Container Service (ECS) for containerized applications.

2. **Storage**: AWS provides scalable storage solutions such as Amazon Simple Storage Service (S3) for object storage, Amazon Elastic Block Store (EBS) for block storage, and Amazon Glacier for long-term archival storage.

3. **Networking**: AWS offers networking services such as Amazon Virtual Private Cloud (VPC) for isolated cloud resources, Amazon Route 53 for domain name system (DNS) management, and AWS Direct Connect for dedicated network connections.

4. **Databases**: AWS provides various database services, including Amazon Relational Database Service (RDS) for managed relational databases, Amazon DynamoDB for NoSQL databases, and Amazon Redshift for data warehousing.

5. **Machine Learning and Artificial Intelligence**: AWS offers machine learning and AI services such as Amazon Sage Maker for building, training, and deploying machine learning models, Amazon Comprehend for natural language processing, and Amazon Recognition for image and video analysis.

6. **Analytics**: AWS provides analytics services such as Amazon Athena for interactive query analysis, Amazon EMR for big data processing, and Amazon Quick Sight for business intelligence.

7. **Internet of Things (IoT)**: AWS offers IoT services such as AWS IoT Core for connecting IoT devices to the cloud, AWS IoT Greengrass for extending cloud capabilities to edge devices, and AWS IoT Analytics for IoT data analysis.

8. **Security and Identity**: AWS provides security and identity services such as AWS Identity and Access Management (IAM) for managing user access, Amazon Inspector for security assessment, and AWS Key Management Service (KMS) for encryption key management.

Now, regarding how AWS can help in the metaverse:

1. **Scalability**: AWS offers scalable computing resources, allowing metaverse platforms to handle fluctuating user demands and scale resources up or down as needed.

2. **Storage**: AWS provides reliable and scalable storage solutions, enabling metaverse platforms to store vast amounts of user-generated content, virtual environments, and assets.

3. **Networking**: AWS offers networking services that support low-latency communication, facilitating real-time interactions and immersive experiences within the metaverse.

4. **Machine Learning and AI**: AWS's machine learning and AI services can enhance the metaverse with intelligent features such as personalized content recommendations, natural language understanding, and computer vision capabilities.

5. **Analytics**: AWS analytics services can help metaverse platforms gain insights into user behavior, engagement patterns, and content preferences, enabling them to optimize experiences and content delivery.

6. **Security**: AWS provides robust security and identity management services to protect user data, transactions, and virtual assets within the metaverse, ensuring a safe and secure environment for users.

3. Oculus (AR /VR Headsets)

Oculus is a brand of virtual reality (VR) hardware and software products developed by Oculus VR, a subsidiary of Meta Platforms (formerly known as Facebook). The company's primary focus is on creating VR headsets that allow users to experience immersive virtual environments and interact with digital content as if they were physically present within the virtual world. Oculus headsets come with advanced motion tracking technology, which enables the system to detect and respond to your head and hand movements. This tracking allows for a sense of presence in the virtual world and enables you to interact with objects and elements within the environment.

Example: Reaching out your hand in VR to pick up a virtual object or ducking to avoid an incoming virtual projectile.

OCULUS role in Metaverse:

In the context of the "metaverse," which is an interconnected virtual reality space where users can interact, socialize, and engage in various digital experiences, Oculus plays a crucial role. It contributes to the metaverse by offering a platform for users to access and participate in a wide range of virtual experiences, social interactions, entertainment, and productivity applications.

1. **Immersive Experiences**: Oculus headsets provide a high level of immersion, making it feel like you are physically present in the virtual world.
2. **Social Interaction**: In the metaverse, people from all over the world can connect and socialize. Oculus offers social VR experiences that allow users to meet and interact with friends, family, or even strangers in virtual environments.
3. **Entertainment and Gaming**: Oculus supports a wide range of VR games and entertainment applications.
4. **Education and Training**: Oculus can be utilized for educational purposes, allowing users to learn in interactive and immersive environments.
5. **Virtual Workspaces**: As remote work becomes more prevalent, VR could offer virtual workspaces where people can collaborate and conduct meetings in a virtual environment, making work interactions more engaging and productive.
6. **Creative Expression**: Oculus also enables users to create and share their virtual experiences, art, and content within the metaverse.

4. **Python (Colab /Jupiter Notebooks)**

Python, renowned for its versatility and ease of use, plays a pivotal role in both building the metaverse and developing 3D designs, particularly in collaboration with NVIDIA technologies. Here's how:

Building the Metaverse:

1. **Scripting and Automation**: Python's scripting capabilities enable developers to automate various tasks involved in building the metaverse, such as asset creation, scene generation, and interaction programming.
2. **Integration with APIs**: Python boasts extensive support for APIs, allowing seamless integration with metaverse platforms and services. Developers can leverage Python to interact with APIs for virtual worlds, social platforms, and cloud services, facilitating the creation of immersive experiences.
3. **Prototyping and Rapid Development**: Python's concise syntax and rich ecosystem of libraries expedite prototyping and rapid development of metaverse applications. From virtual environments to avatar customization tools, Python empowers developers to bring their creative visions to life efficiently.

Developing 3D Designs with NVIDIA:

1. **NVIDIA Omniverse and Python**: NVIDIA Omniverse, a platform for real-time collaboration and simulation, offers Python APIs that enable developers to manipulate 3D content programmatically. Python scripts can control aspects of scene composition, lighting, materials, and physics simulations within Omniverse, facilitating the creation of complex and dynamic 3D designs.
2. **Integration with Blender and Maya**: Python serves as the primary scripting language for popular 3D modeling software such as Blender and Autodesk Maya. NVIDIA provides plugins and tools that leverage Python to bridge the gap between these modeling environments and Omniverse, enabling seamless interoperability and content exchange.
3. **AI-Assisted Design**: NVIDIA's AI-powered tools, such as GauGAN for image generation and SimNet for physics simulation, can be accessed and controlled using Python APIs. This integration empowers designers to harness the power of AI to enhance their creative workflows and generate realistic 3D content more efficiently.

2.3 Block Diagram

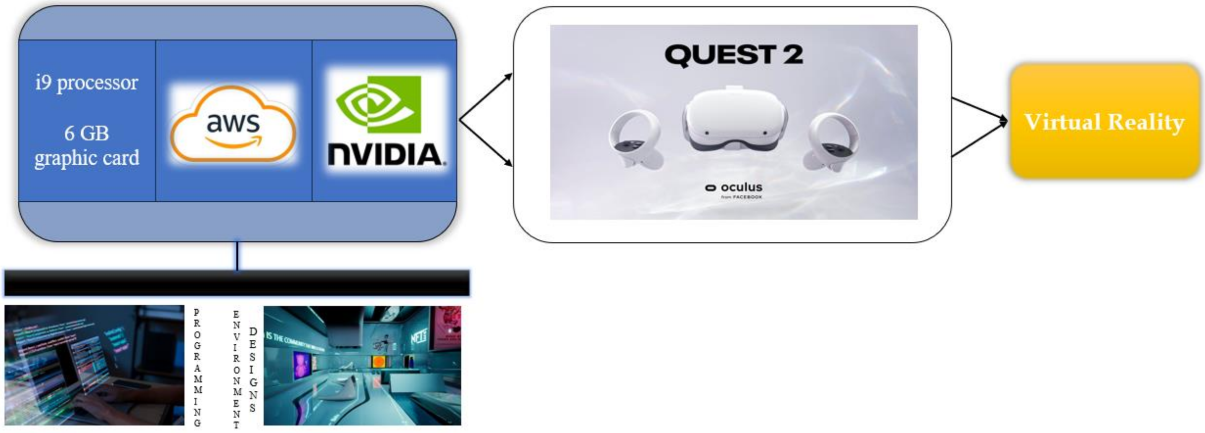


Figure 2.3: Block diagram of Implementation

2.4 Methodology for Data Collection and Analysis

Data collection will involve gathering feedback from participants through surveys, interviews, and observation of their interactions with the platforms. Quantitative data, such as user ratings and performance metrics, will be collected to measure the effectiveness of each platform. Qualitative data, such as user comments and suggestions, will also be collected to gain deeper insights into the user experience.

Data analysis will include statistical analysis of quantitative data to compare the performance of each platform. Qualitative data will be analyzed thematically to identify common themes and patterns in user feedback. The findings will be used to draw conclusions and make recommendations for office collaboration in the metaverse.

CHAPTER 3

Implementation

3.1 Setting up an Omniverse Development Environment

Step – 1:

1. Download Visual Studio Code.
2. Perform the standard installation.

Step – 2: Tutorial

Developing on Nvidia Omniverse - How to Build an App

What Is an App?

An Omniverse Kit app is a ***.kit file***. It is a single file extension. The Kit executable (e.g., Kit.exe) is the engine that runs the app based on the configuration defined in the *Kit file*.

Step 3: **Create an Omniverse App**

Step 3.1: Clone the Repo

i. **Open** VSCode

ii. **Open** the command palette using Ctrl + Shift + P

iii. In the palette prompt **enter** gitcl then **select** Git: Clone command

iv. **Paste** https://github.com/NVIDIA-Omniverse/kit-project-template into the repository URL then **select** Clone from URL

v. **Select (or create)** the local directory into which you want to clone the project

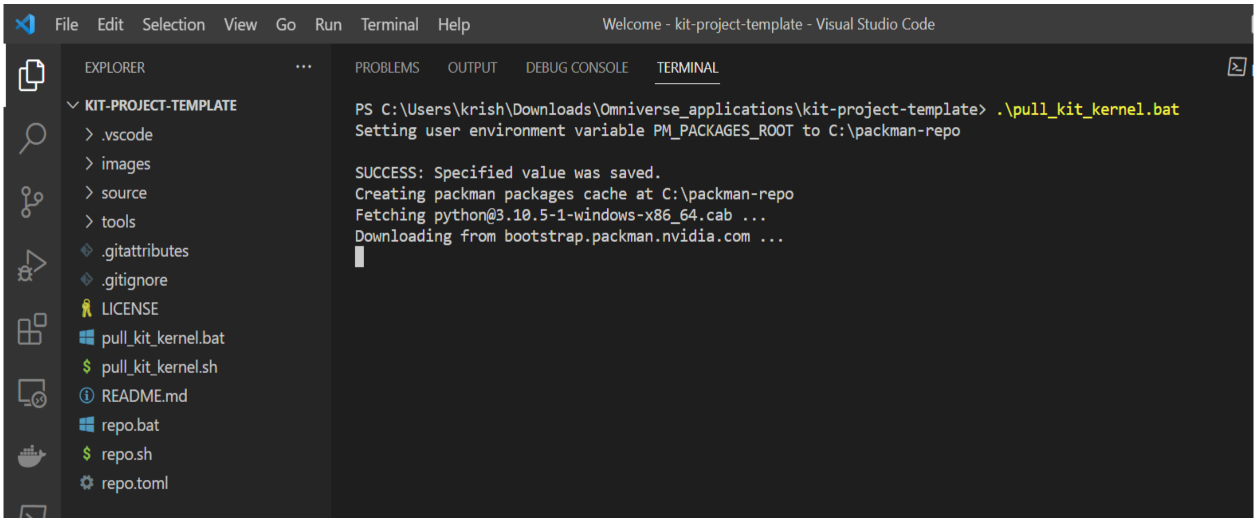
vi. Once it has finished cloning it will ask if you want to open the cloned repository, select **Open**

Step 3.2: Pulling Kit Code

i. **Open** a new terminal in *VSCode*

ii. **Type** .\pull_kit_kernel.bat (windows) or .\pull_kit_kernel.sh (linux)

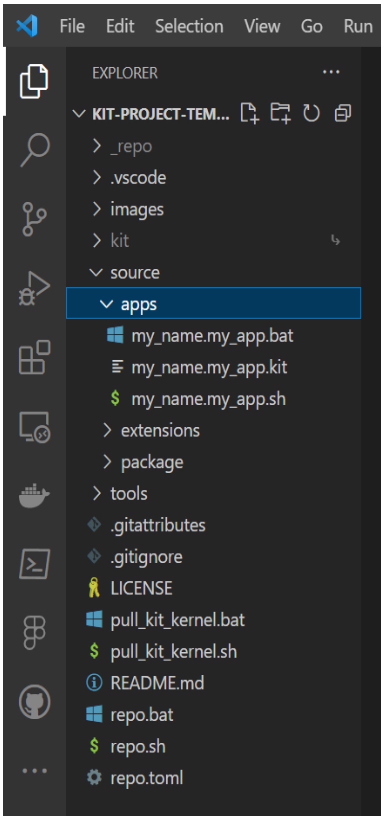
iii. **Hit enter** to run



```
PS C:\Users\krish\Downloads\Omniverse_applications\kit-project-template> .\pull_kit_kernel.bat
Setting user environment variable PM_PACKAGES_ROOT to C:\packman-repo

SUCCESS: Specified value was saved.
Creating packman packages cache at C:\packman-repo
Fetching python@3.10.5-1-windows-x86_64.cab ...
Downloading from bootstrap.packman.nvidia.com ...
```

Figure 3.11: Implementation picture in Visual Studio



Overview of Files

1. **pull_kit_kernel.bat (Windows)**: A script that pulls the Omniverse Kit Kernel. By running this, a kit folder link will be created.

2. **repo.bat (Windows)**: Script file which runs the Repo Tools

3. **repo.toml**: Configuration file used by Repo Tools

4. **source/apps/my_name.my_app.kit**: A template Kit file. This will be our working file throughout the tutorial

5. **source/apps/my_name.my_app.bat**: This script runs kit.exe and passes it my_name.my_app.kit

6. **source/extensions/**: Location of local extensions. Currently there is a template extension called hello world.

7. **/kit**: Contains reference to Kit directory after running pull_kit_kernel.bat/sh

8. **/kit/apps**: Contains documented sample Kit files that can be used as a reference.

Figure 3.12: Over view in Visual Studio

Step 3.3: Running the App

i. **Open** a new terminal if one isn't already open

ii. **Type** .\source\apps\my_name.my_app.bat (windows)
or .\source\apps\my_name.my_app.bat (linux)

iii. **Hit** Enter

Step 3.4: Adding an Extension

i. **Close** any running instance of your project, then head back to *VSCode*

ii. **Navigate** to source/apps/ and **Open** my_name.my_app.kit

This is your working *Kit file* that contains your configurations for your App

iii. **Locate** the line [dependencies] and **add** a new line after it Extensions will go under [dependencies] this tells the project which Extensions are required. Order in which it gets placed does not matter.

iv. **Add** the following lines of code under [dependencies]

```
 # Extensions window
"omni.kit. window.extensions" = {}
"omni.app.setup" = {}
```

v. **Remove** the following lines:

```
 # Create Kit UI Based applications
 "omni.app.editor.base" = {}
```

vi. **Save** the file

Step 3.5: Finding Extensions

i. **Run the project**: .\source\apps\my_name.my_app.bat

ii. **Click on** Window > Extensions in the App

You can drag and dock the window

iii. In the Extensions window, type "extensions" in the search bar

Note: i.e. omni.kit.welcome.extensions, omni.kit.windows.extensions, etc., These are the package names for each extension.

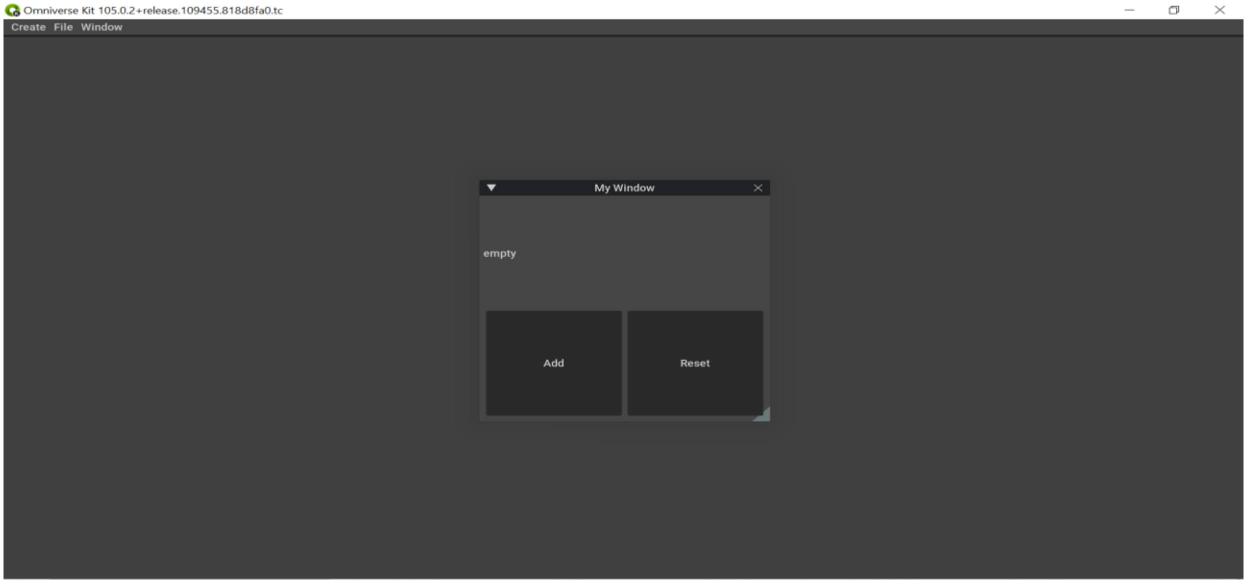


Figure 3.13: Omniverse Kit in Visual Studio

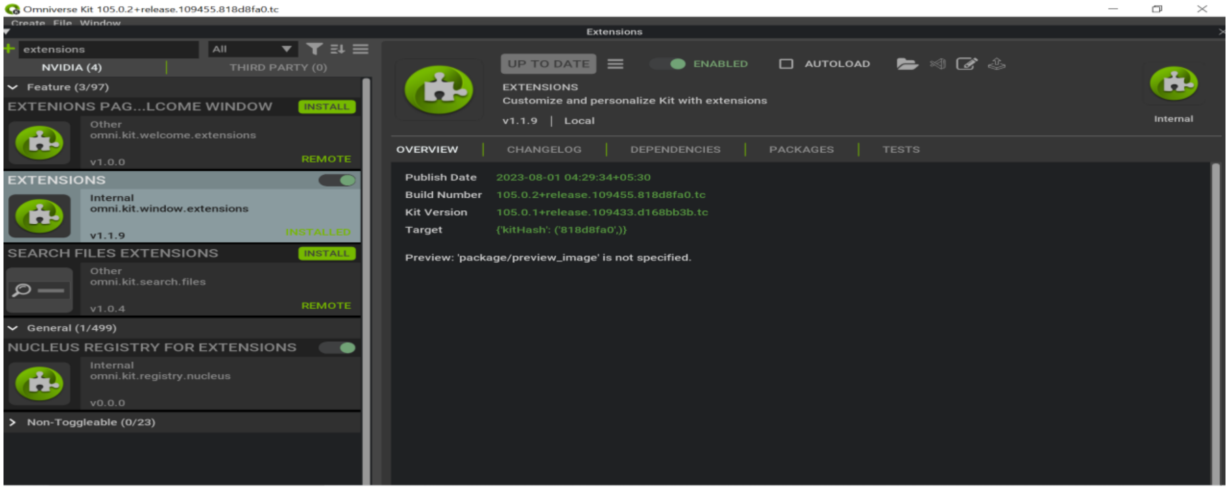
iv. **Select / Click** on Extensions

v. **Click** on Dependencies

vi. For a list view you can select the Toggle View button in the bottom right of the window.

For node view, you can zoom with the mouse wheel and use the middle mouse wheel to pan around.

vii. **Close** the project and head back to VSCode



Publish Date 2023-08-01 04:29:34+05:30
Build Number 105.0.2+release.109455.818d8fa0.tc
Kit Version 105.0.1+release.109433.d168bb3b.tc
Target {'kitHash': ('818d8fa0',)}

Preview: 'package/preview_image' is not specified.

Figure 3.14: Extension pack in Visual Studio

Simply by adding in **omni.kit.window.extensions** we have also imported all the extensions listed in the dependency graph.It is good to note what the dependencies of each extension are. That way you can simplify your *Kit file* and not have to add every extension.

Step 5.6: Adding the Viewport

i. Back in VSCode add the following lines in the [dependencies] section of

   a. **my_name.my_app.kit**:

```
# Viewport Bundle
"omni.kit.viewport.bundle" = {}
# Enable Pixar Storm for your viewport
"omni.hydra.pxr" = {}
"omni.hydra.pxr.settings" = {}
```

ii. Next, you need to **enable** Pixar Storm through settings. To do this, **add** the following lines after [settings.app]:

```
content.emptyStageOnStart = true
exts."omni.kit.viewport.window".startup.windowName = "Viewport"
exts."omni.kit.renderer.core".compatibilityMode = true
# here you can choose what renderers are enabled
renderer.enabled = "pxr"
renderer.active = "pxr"
```

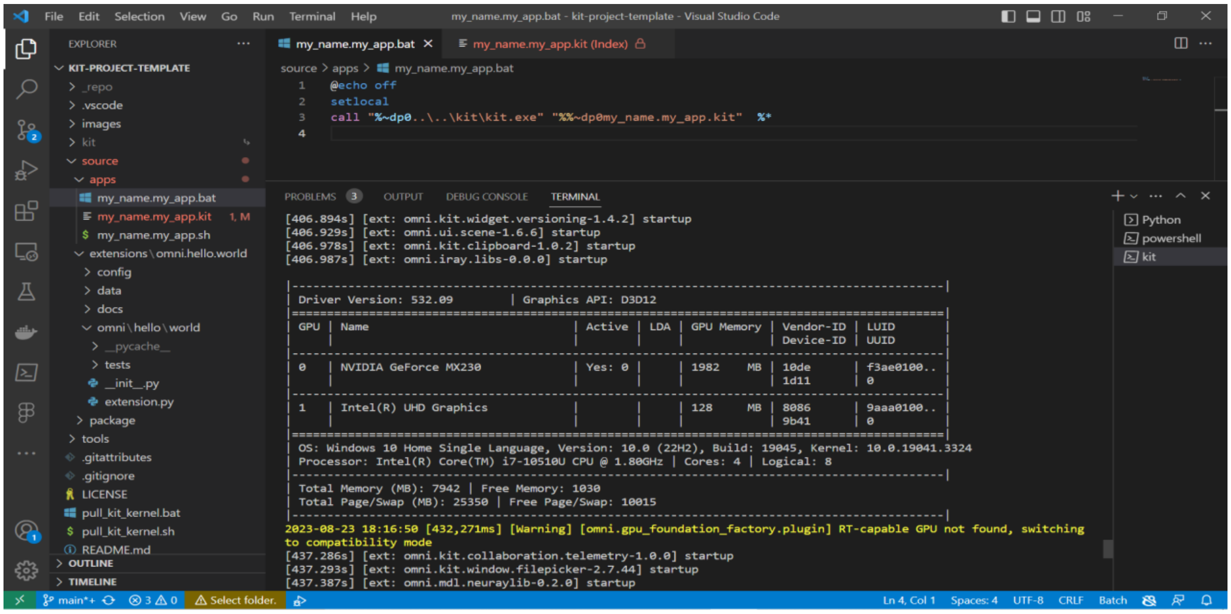
iii. Make sure you **save** your *Kit file*

i. **Let’s view the results of adding an extension**:

- **Close** any instance of the project and **run** the project again. You should now see a **Viewport window**
- **Move** the ‘My Window’ off to the side
- **Go to** *Create > Mesh > Cone*
- Notice how the manipulator is enabled. You can also navigate the camera in the viewport as well
- Press the Right Mouse button to Orbit around
- Use the Scroll Wheel or Press Right Mouse Button + "OPT"/"Alt" to zoom in
- Press the Middle Mouse Button to Pan

With just a few extensions, you already have the capability to view, create and manipulate prims in the scene.- Prim is short for "**primitive**", a fundamental unit in Omniverse. Anything imported or created in a USD scene is a prim. This includes, cameras, sounds, lights, meshes, and more.

Below are the images and user interfaces listed, we can view and edit them directly in Visual Studios.



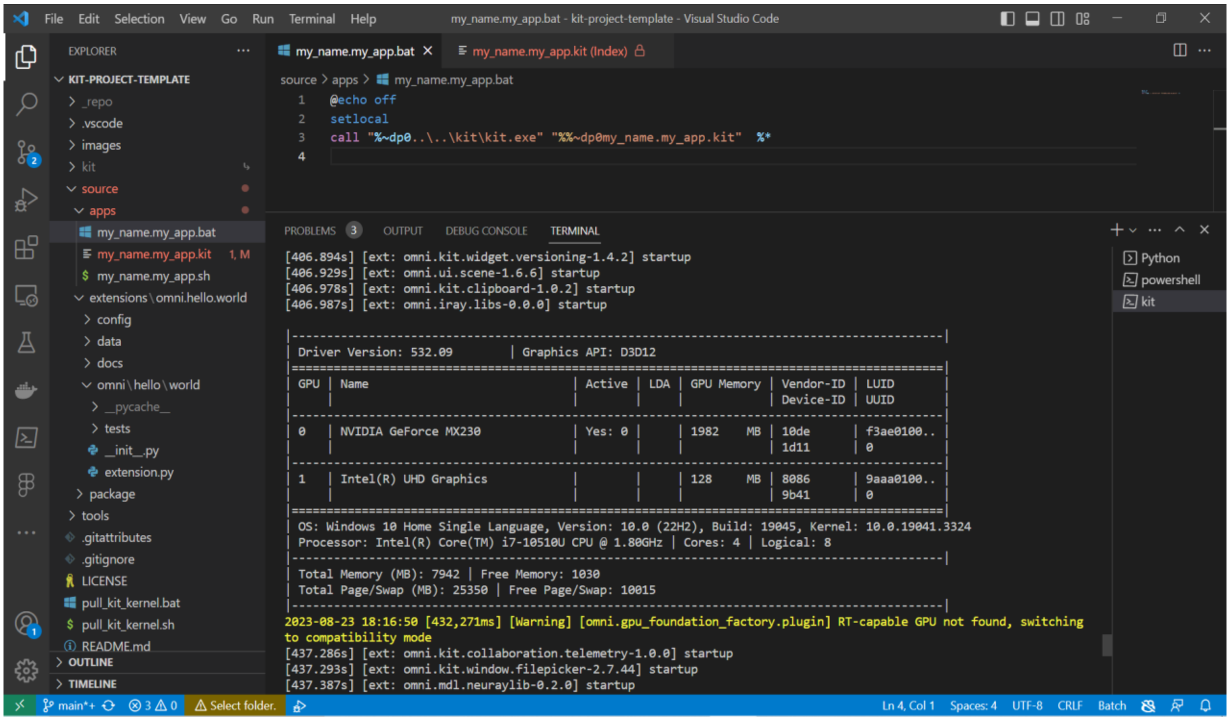
```
@echo off
setlocal
call "%~dp0..\..\kit\kit.exe" "%%~dp0my_name.my_app.kit"  %*

```

```
[406.894s] [ext: omni.kit.widget.versioning-1.4.2] startup
[406.929s] [ext: omni.ui.scene-1.6.6] startup
[406.978s] [ext: omni.kit.clipboard-1.0.2] startup
[406.987s] [ext: omni.iray.libs-0.0.0] startup

 |---------------------------------------------------------------------------------------|
 | Driver Version: 532.09        | Graphics API: D3D12
 |=======================================================================================|
 | GPU | Name                              | Active | LDA | GPU Memory | Vendor-ID | LUID        |
 |     |                                   |        |     |            | Device-ID | UUID        |
 |---------------------------------------------------------------------------------------|
 | 0   | NVIDIA GeForce MX230              | Yes: 0 |     | 1982    MB | 10de      | f3ae0100.. |
 |     |                                   |        |     |            | 1d11      | 0           |
 |---------------------------------------------------------------------------------------|
 | 1   | Intel(R) UHD Graphics             |        |     | 128     MB | 8086      | 9aaa0100.. |
 |     |                                   |        |     |            | 9b41      | 0           |
 |=======================================================================================|
 | OS: Windows 10 Home Single Language, Version: 10.0 (22H2), Build: 19045, Kernel: 10.0.19041.3324
 | Processor: Intel(R) Core(TM) i7-10510U CPU @ 1.80GHz | Cores: 4 | Logical: 8
 |---------------------------------------------------------------------------------------|
 | Total Memory (MB): 7942 | Free Memory: 1030
 | Total Page/Swap (MB): 25350 | Free Page/Swap: 10015
 |---------------------------------------------------------------------------------------|
2023-08-23 18:16:50 [432,271ms] [Warning] [omni.gpu_foundation_factory.plugin] RT-capable GPU not found, switching
to compatibility mode
[437.286s] [ext: omni.kit.collaboration.telemetry-1.0.0] startup
[437.293s] [ext: omni.kit.window.filepicker-2.7.44] startup
[437.387s] [ext: omni.mdl.neuraylib-0.2.0] startup
```

Figure 3.15: my_name.my_app.kit



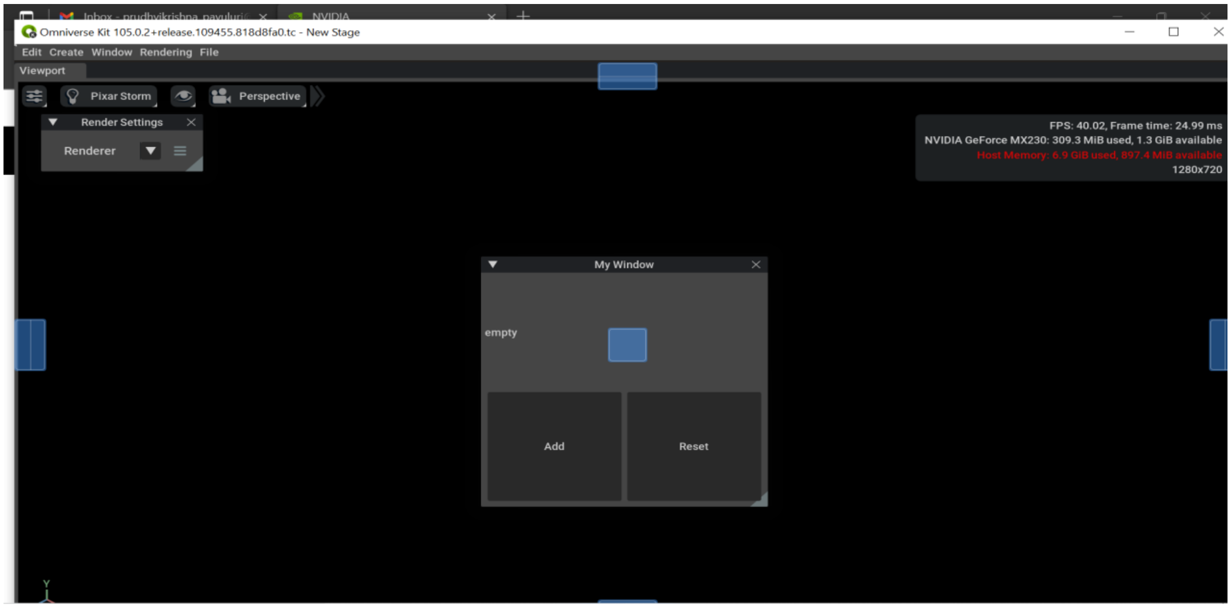
```
@echo off
setlocal
call "%~dp0..\..\kit\kit.exe" "%%~dp0my_name.my_app.kit"  %*

```

```
[406.894s] [ext: omni.kit.widget.versioning-1.4.2] startup
[406.929s] [ext: omni.ui.scene-1.6.6] startup
[406.978s] [ext: omni.kit.clipboard-1.0.2] startup
[406.987s] [ext: omni.iray.libs-0.0.0] startup

 |---------------------------------------------------------------------------------------|
 | Driver Version: 532.09        | Graphics API: D3D12
 |=======================================================================================|
 | GPU | Name                              | Active | LDA | GPU Memory | Vendor-ID | LUID        |
 |     |                                   |        |     |            | Device-ID | UUID        |
 |---------------------------------------------------------------------------------------|
 | 0   | NVIDIA GeForce MX230              | Yes: 0 |     | 1982    MB | 10de      | f3ae0100.. |
 |     |                                   |        |     |            | 1d11      | 0           |
 |---------------------------------------------------------------------------------------|
 | 1   | Intel(R) UHD Graphics             |        |     | 128     MB | 8086      | 9aaa0100.. |
 |     |                                   |        |     |            | 9b41      | 0           |
 |=======================================================================================|
 | OS: Windows 10 Home Single Language, Version: 10.0 (22H2), Build: 19045, Kernel: 10.0.19041.3324
 | Processor: Intel(R) Core(TM) i7-10510U CPU @ 1.80GHz | Cores: 4 | Logical: 8
 |---------------------------------------------------------------------------------------|
 | Total Memory (MB): 7942 | Free Memory: 1030
 | Total Page/Swap (MB): 25350 | Free Page/Swap: 10015
 |---------------------------------------------------------------------------------------|
2023-08-23 18:16:50 [432,271ms] [Warning] [omni.gpu_foundation_factory.plugin] RT-capable GPU not found, switching
to compatibility mode
[437.286s] [ext: omni.kit.collaboration.telemetry-1.0.0] startup
[437.293s] [ext: omni.kit.window.filepicker-2.7.44] startup
[437.387s] [ext: omni.mdl.neuraylib-0.2.0] startup
```

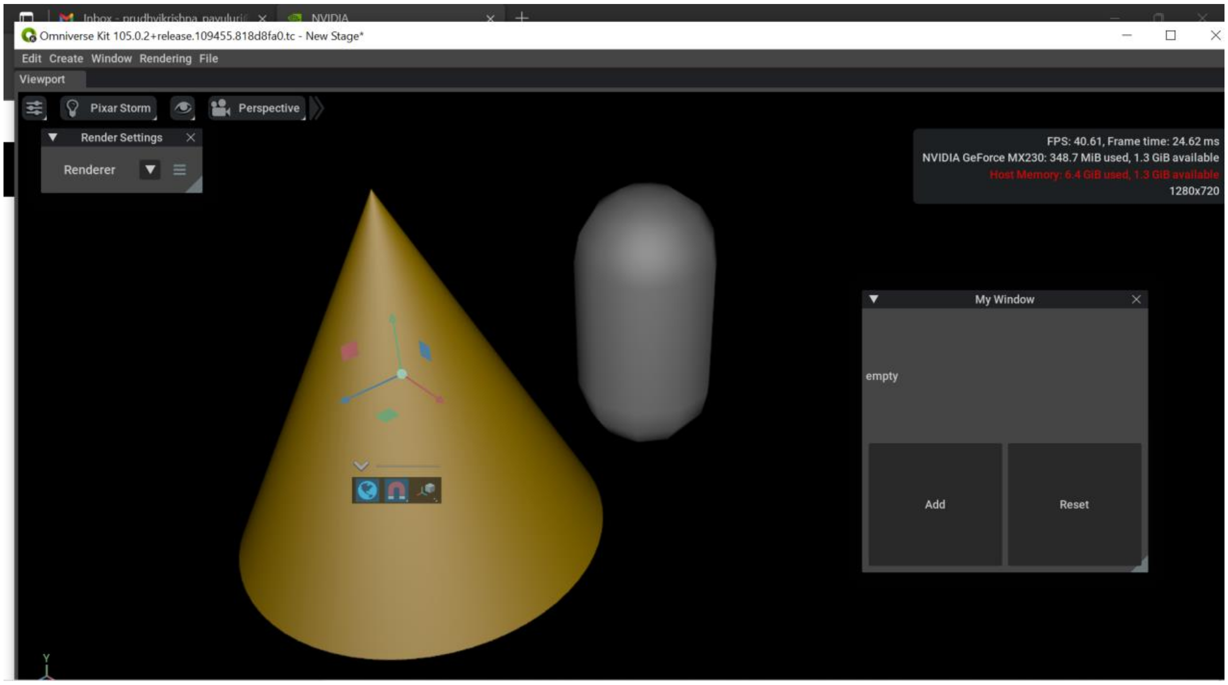
Figure 3.16: my_name.my_app.kit



Inbox - prudhvikrishna pavuluri
NVIDIA
Omniverse Kit 105.0.2+release.109455.818d8fa0.tc - New Stage
Edit Create Window Rendering File
Viewport
Pixar Storm
Perspective
Render Settings
Renderer
My Window
empty
Add
Reset
Y

```
FPS: 40.02, Frame time: 24.99 ms
NVIDIA GeForce MX230: 309.3 MiB used, 1.3 GiB available
Host Memory: 6.9 GiB used, 897.4 MiB available
1280x720
```

Figure 3.17: View Port Image-1



Inbox - prudhvikrishna pavuluri
NVIDIA
Omniverse Kit 105.0.2+release.109455.818d8fa0.tc - New Stage*
Edit Create Window Rendering File
Viewport
Pixar Storm
Perspective
Render Settings
Renderer
My Window
empty
Add
Reset
Y

```
FPS: 40.61, Frame time: 24.62 ms
NVIDIA GeForce MX230: 348.7 MiB used, 1.3 GiB available
Host Memory: 6.4 GiB used, 1.3 GiB available
1280x720
```

Figure 3.18: View Port Image-2

Viewport Settings:

- This is the explanation to the above code we added.

i. **content.emptyStageOnStart = true**

This will create an empty stage during the startup process

ii. **exts."omni.kit.viewport.window".startup.windowName = "Viewport"**

Setup Kit to create an ‘omni.kit.viewport.window` Window named Viewport

iii. **exts."omni.kit.renderer.core".compatibilityMode = true**

This forces Compatibility Mode on. Compatibility mode allows the user to run the project if they do not have RTX.

iv. **renderer.enabled = "pxr"**

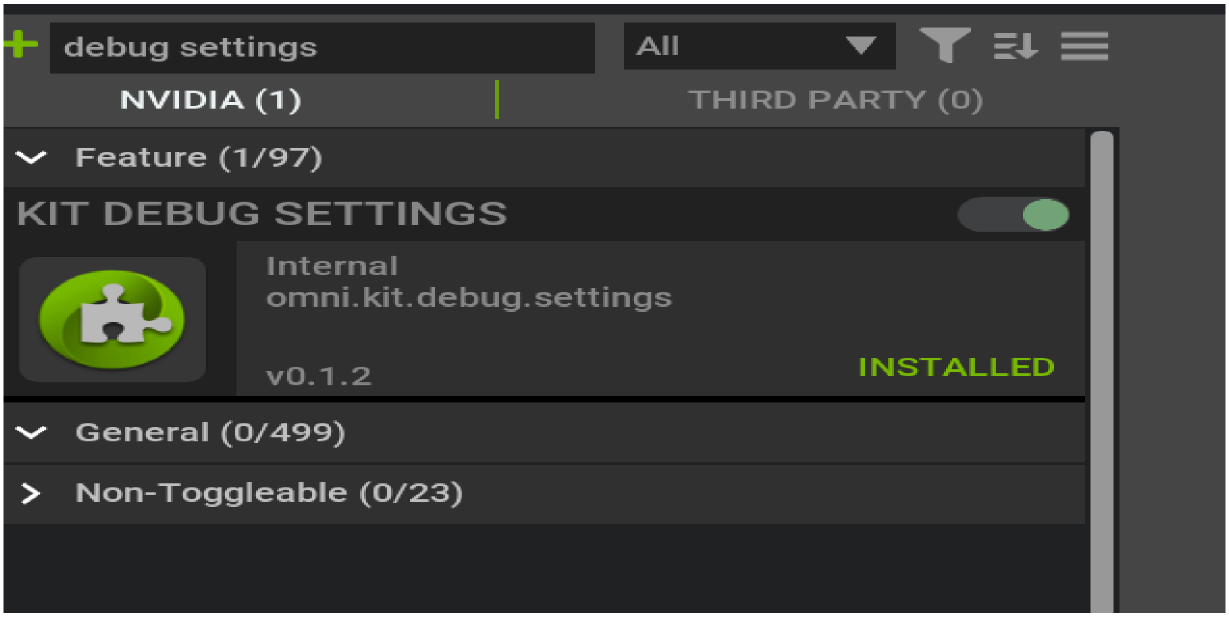
Renderers that the user can pick between

v. **renderer.active = "pxr”**

Default renderer that is active upon launching

Additional Settings:

1. **Run** the project
2. **Go to** Window > Extensions
3. **Search** for debug settings
4. **Install and Enable** "omni.kit.debug.settings"
5. Each setting can be hovered over so you can see the path.
6. For example, if we look at audio/enabled the path is /app/audio/enabled. To add this setting in the *Kit file* we add it to [settings.app].



debug settings

All

NVIDIA (1)

THIRD PARTY (0)

Feature (1/97)

KIT DEBUG SETTINGS

Internal
omni.kit.debug.settings

v0.1.2

INSTALLED

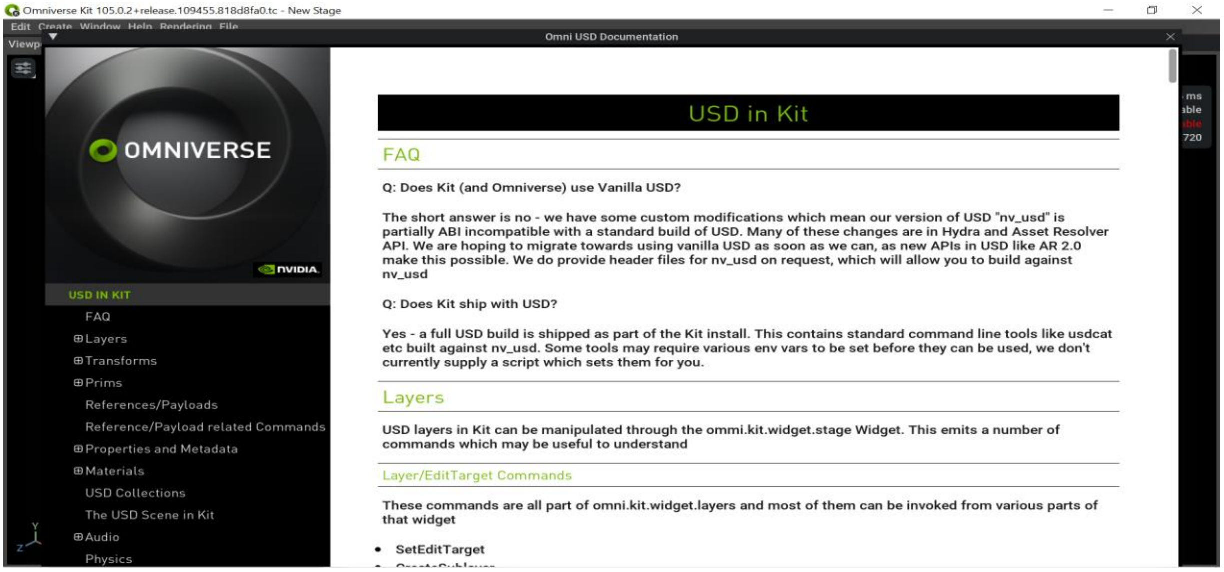
General (0/499)

Non-Toggleable (0/23)

Figure 3.19: KIT Debug Settings

Step 6: Finding an Extension Name

1. Inside the App Open the Extension Window.
   - Window > Extensions
   - Notice for each Extension box it includes the Title, and inside the box it has the category and a short name i.e. omni.graph.bundle.action, omni.kit.window.environment
2. In the search bar type usd_doc.
3. Take note the package name is omni.kit.usd_docs.
4. Close the App and head back to VSCode and add the following line in our [dependencies] section. Order does not matter as long as it's under [dependencies].
   - "omni.kit.usd_docs" = {}
5. Save the Kit file and rerun the project.
6. Go to Help in the menu bar, you should see USD Docs in the list.



USD in Kit

FAQ

Q: Does Kit (and Omniverse) use Vanilla USD?

The short answer is no - we have some custom modifications which mean our version of USD "nv_usd" is partially ABI incompatible with a standard build of USD. Many of these changes are in Hydra and Asset Resolver API. We are hoping to migrate towards using vanilla USD as soon as we can, as new APIs in USD like AR 2.0 make this possible. We do provide header files for nv_usd on request, which will allow you to build against nv_usd

Q: Does Kit ship with USD?

Yes - a full USD build is shipped as part of the Kit install. This contains standard command line tools like usdcat etc built against nv_usd. Some tools may require various env vars to be set before they can be used, we don't currently supply a script which sets them for you.

Layers

USD layers in Kit can be manipulated through the ommi.kit.widget.stage Widget. This emits a number of commands which may be useful to understand

Layer/EditTarget Commands

These commands are all part of omni.kit.widget.layers and most of them can be invoked from various parts of that widget

- SetEditTarget

Figure 6: USD in Kit

Step 7: Hello World

1. Underneath "omni.hydra.pxr.settings" = {} add the following lines:

```
# File Menu
"omni.kit.menu.file" = {}
# Stage
"omni.kit.window.stage" = {}
# Content Browser
"omni.kit.window.content_browser" = {}
```

```
# Edit Menu
"omni.kit.menu.edit" = {}
# ToolBar
"omni.kit.window.toolbar" = {}
# Property Window
"omni.kit.property.bundle" = {}
# Variant presenter
"omni.kit.variant.presenter" = {}
# USD Docs
"omni.kit.usd_docs" = {}
# USD Paths
"omni.kit.window.usd_paths" = {}
# USD Collection window
"omni.kit.window.collection" = {}
```

2. **Save** the *Kit file* and **close** any instances of Omniverse.

3. **Run** the project, you should now see the extensions added.

Note: If you have more than the listed extensions in the previous steps, your project may look a bit different from the above images provided.

Step 7: Extensions?

- **File Menu & Edit Menu**: File menu offers access to file operations both locally and on your Nucleus. Edit Menu contains editing tools like Undo, Redo, Select, along with other more advanced capabilities.

- **Stage**: Here you will be able to manage all the assets in your USD Scene, listed in a hierarchical (parent/child) order. This is essentially for large scenes.

- **Content Browser**: Items listed in the Content Browser provide meaningful information about the asset without having to open it.

- **Tool Bar**: Provides easy access to common commands needed when editing a USD Scene.

- **Property Window**: Provides you a way of configuring properties for the prim selected.

- **Variant Presenter**: Gives you convenient access to all the USD Variants in a scene. Can also be used to organize variants into custom groups using USD Collections.

- **USD Docs**: Interactive Documentation with coding examples for omni.kit.usd

- **USD Paths**: Easy tool for you to search and replace paths of prims in the scene.

- **USD Collection Window**: Collections are a way to group/organize prims and properties in USD

Step 8: Styling

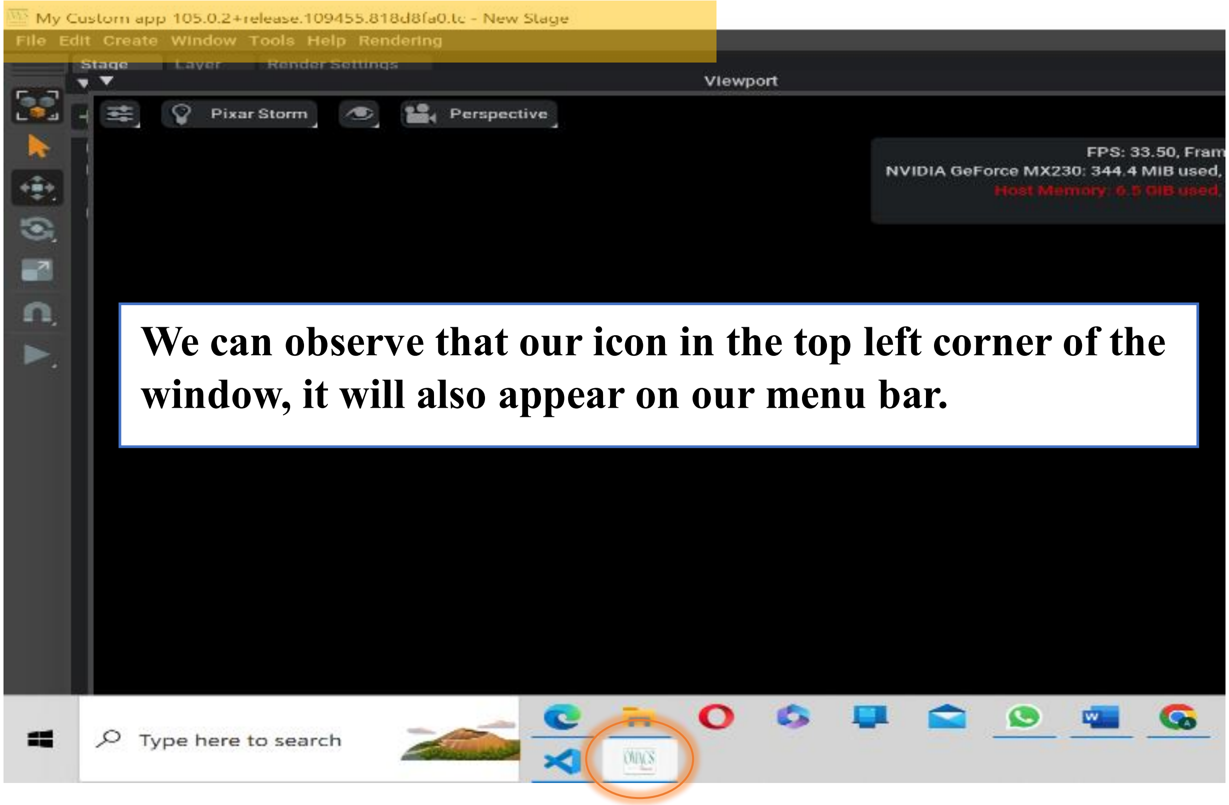
Some of the methods in styling:

1. **Changing the Icon that appears in the top left corner of the app window**
2. **Customizing the Title Bar color**
3. **Adding Splash Screens**
4. **Changing Fonts**
5. **Shading Extension Colors**

Step 8.1: Change the Icon

1. In VSCode, create a new folder in apps called data
2. Make sure your .ico image is stored in this folder: You need to have a .ico image with a size of 64x64 and resolution of 72.
3. In VSCode, Add [settings.app.window] section if you do not have one.
4. Add the following line under [settings.app.window]: iconPath = "${app}/data/icon.ico"

   Note: ${app} refers to /source/apps folder
5. Save and run the project: You should see your icon in the top left corner of the window, it will also appear on your menu bar.


My Custom app 105.0.2+release.109455.818d8fa0.tc - New Stage
File Edit Create Window Tools Help Rendering
Stage Layer Render Settings
Viewport
Pixar Storm
Perspective
FPS: 33.50, Fram
NVIDIA GeForce MX230: 344.4 MIB used,
Host Memory: 6.5 GIB used
Type here to search

We can observe that our icon in the top left corner of the window, it will also appear on our menu bar.

Figure 8.1: My custom app

Step 8.2: Customize the Title Bar

The Title Bar is the Application Window’s main menu bar where you can minimize, maximize and close the window.

1. First, we need to **add** the extension that allows us to change the title bar settings. In the dependencies section **add** the following line: "omni.kit.window.modifier.titlebar" = {}
2. Next, **add** the following section:

   [settings.exts."omni.kit.window.modifier.titlebar"]

   titleFormatString = " My Custom App {verKey:/app/version,font_color=0x909090,font_size=16} {separator} {file, board=true}"

```
icon.file = "${app}/data/icon.ico"
icon.size = 64
name = "Arial"
defaultFont.size = 16
defaultFont.color = 0xD0D0D0
separator.color = 0x00B976
separator.width = 1
windowBorder.color = 0x0F0F0F
windowBorder.width = 2
colors.caption = 0x0F0F0F
colors.client = 0x0F0F0F
```

3. **Save** the file and **Rerun** the project.

Step 8.3: Customize the Title Bar Deep Dive

1. **Title**: titleFormatString – Contains the title that will be displayed in the Title Bar
2. **Icon**:

   icon.file – This is the Icon file that will appear. The extension will override what we provided previously in iconPath but only for the Title Bar.

   icon.size – Defines the Size of the Icon
3. **Font**:

   defaultFont.name – Default font for the Title Bar

   defaultFont.size - Default Font Size

   defaultFont.color – Font Color

4. **Separator**:

   separator.color – Color for the Separator

   separator.width – The Width of the separator

5. **Border**:

   windowBorder.color - Border Color of the window

   windowBorder.width - Width of the window border

6. **Window**

   colors.caption – The Color of the Title Bar

   colors.client – The Color of the window outline

Step 8.4: Updating the Layout

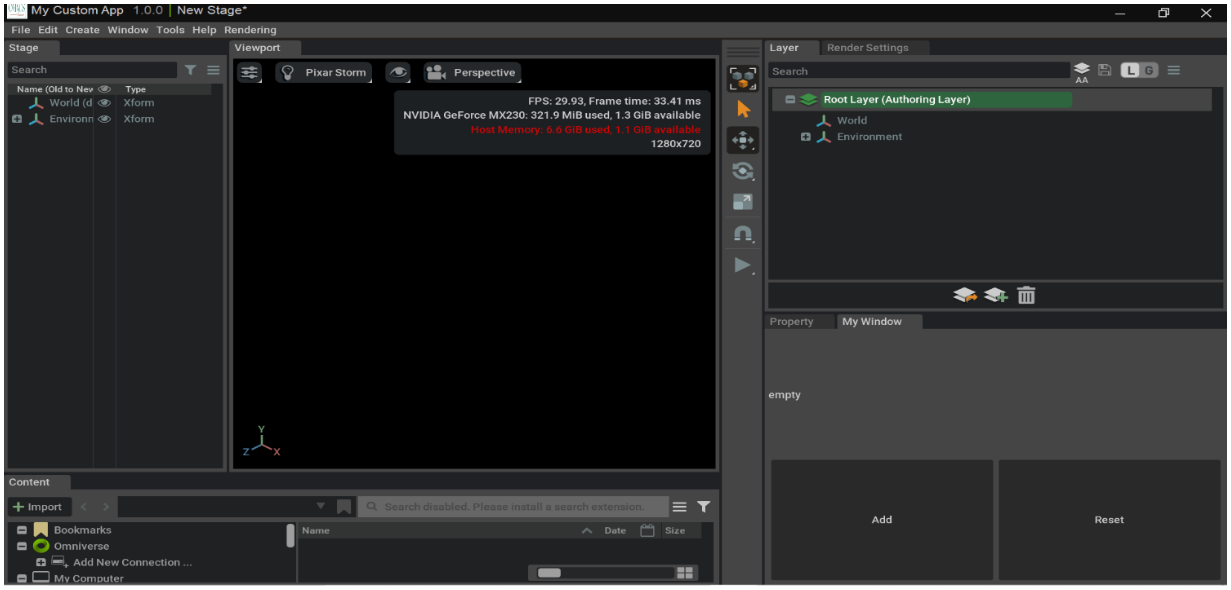
Currently, your windows might not be in the appropriate default locations. You can dock the windows where you like them and then save a layout based on the configuration you've created.

Step 8.5: Configuring a Custom Layout

1. **Run** the project if it is not running already
2. **Dock** your windows in the locations that appeal to you.

   To position the window Left-Click + Drag near the blue preview images.
3. **Repeat** for any other extension windows

Arrange the tabs accordingly

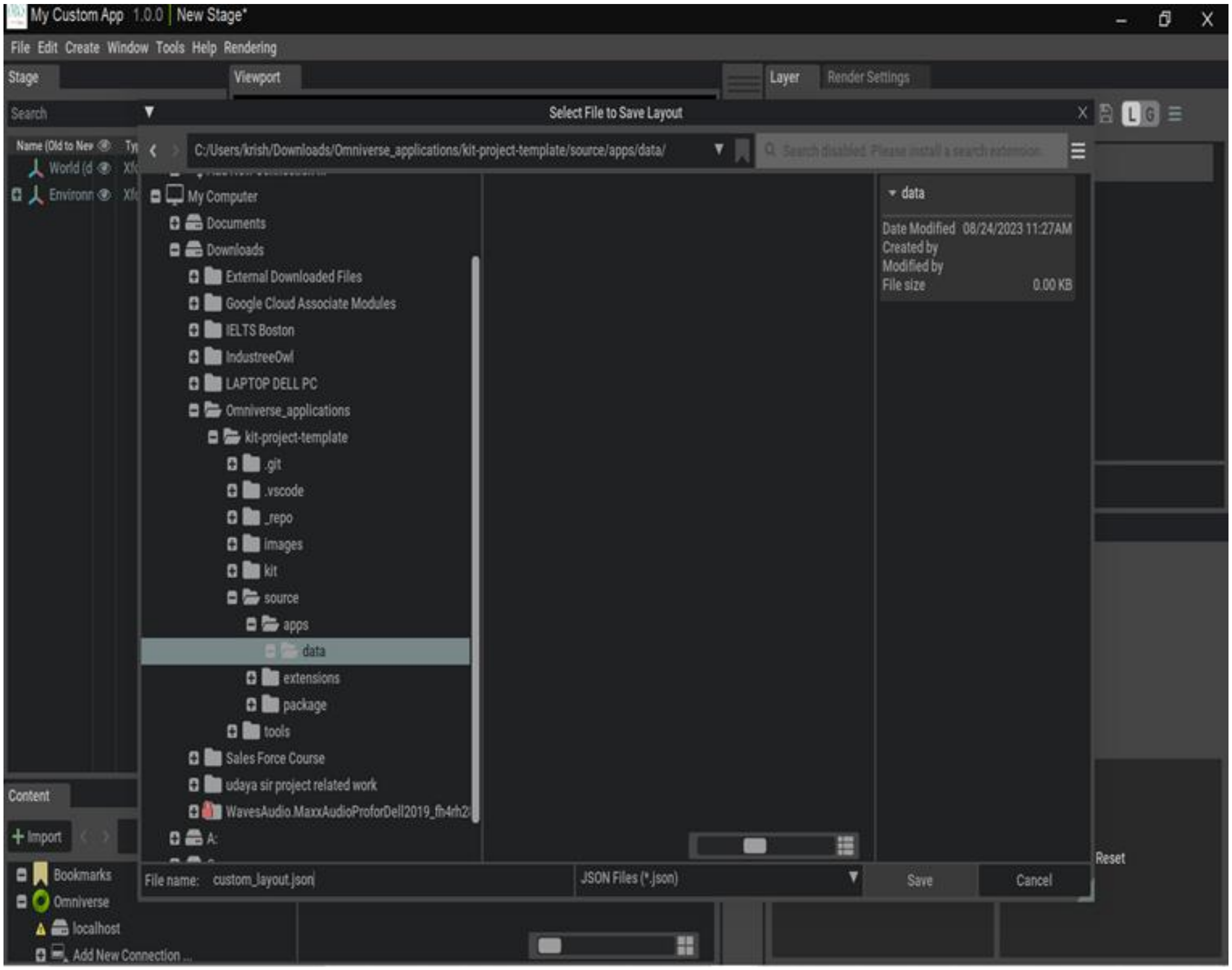


```
FPS: 29.93, Frame time: 33.41 ms
NVIDIA GeForce MX230: 321.9 MiB used, 1.3 GiB available
Host Memory: 6.6 GiB used, 1.1 GiB available
1280x720
```

Figure 8.5: My custom app workspace

Step 8.6: Saving the Layout

1. After setting up the layout *go to* the *Menu Bar* click on Window > Layout > Save Layout....
2. A new prompt window will open. Navigate to your project folder. From your project folder go to source > apps > data.
3. Under filename name it custom_layout then click Save.



C:/Users/krish/Downloads/Omniverse_applications/kit-project-template/source/apps/data/

Date Modified 08/24/2023 11:27AM

File size 0.00 KB

File name: custom_layout.json

JSON Files (*.json)

Figure 8.6: Layout

Step 8.7: Updating Your Application's Layout Reference

1. **Close** the application. Go back to VSCode and in your Kit file.
2. **Add** the following line to the [settings.app] section: This will use the newly created .json layout file. layout.default = "${app}/data/custom_layout.json"
3. **Save** the Kit file.
4. **Run** the project. Now every time you open your App it will retain the same layout.

Step 9: Adding Third Party Extensions

So far, you have added Extensions that come with Omniverse. You can also add Third Party Extensions, these are extensions other community developers create and publish on GitHub publicly.

Step 9.1: Add Third Party Extensions

i. In our *Kit file* in *VSCode*, **scroll down** to the **[settings.exts."omni.kit.registry.nucleus"]** section.

ii. **Add** the following inside the *registries dictionary*, between **{name="kit/sdk....}** and:

{name = "kit/community", url = "https://dw290v42wisod.cloudfront.net/exts/kit/community"}

iii. **Add** a comma at the end of **{name="kit/sdk...}**

iv. **Save** the file and run the project.

v. **Open** the Extension Manager: Window > Extensions

vi. **Click on** Third-Party tab. With the registry now in our Kit file you can now pull all available extensions that are live on GitHub.

vii. Notice the first extension, Align Tool (omni.kit.property.align). You will be adding this as an example. Close the App and go back to VSCode.

Locate the [dependencies] section and **add** the following line: "omni.kit.property.align" = {}

By default, all *Untrusted Extensions* will **NOT** install. To override this, you must change the setting. *Use this wisely.* **Locate** [settings.app] section and **add** the following line: extensions.installUntrustedExtensions = true

Save the file and **run** the project.

Head back to the *Extension Manager* and under the *Third-Party tab* you should see that the *Align Tool* is installed and enabled.

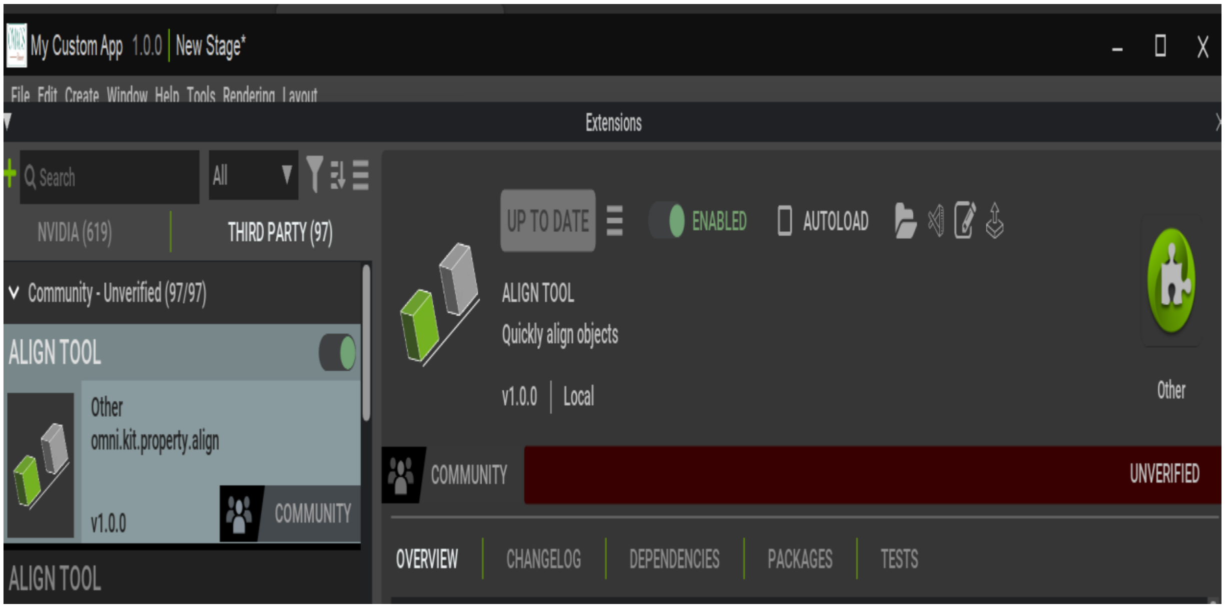


Figure 9.1: Layout

Step 9.2: Add Extensions Locally

1. **Take** any locally created extension and **copy ONLY** the extension. The project directory should not be included. i.e. my.local.extension, omni.hello.world, omni.spawn.prims, etc.
2. In the project folder, **add/paste** your extension to source/extensions. Currently, there is one extension in the folder, omni.hello.world
3. In your *Kit file*, **add** your extension name. For example, omni.example.spawn_prims has been added to my extension folder so in my *Kit file* I will add the following line under [dependencies]: "omni.example.spawn_prims" = {}
4. **Save** the file and **run** the project. You should now see your extension has been added to your App.

Step 10: Packaging and Distribution

1. After Downloading the Zipped folder, **Extract it**
2. **Run** pull_kit_kernel.bat (windows) or pull_kit_kernel.sh (linux) inside the package
3. **Navigate** and **run** source/apps/[app_file].bat (windows)
   or source/apps/[app_file].bat (windows)

Step 11: Change the App Name

Step 12: How to Package the App

i. To package an app, **go to** *VSCode* and **Open** a *Terminal*
ii. **Type** the following in the *Terminal*:

```
.\tools\package.bat (windows) or .\tools\package.sh (linux) (or)

repo package
```

iii. **Hit** Enter

The package will be created in the _build/packages folder

3.2 Setting up the Oculus Environment

i. Oculus Meta Quest 2

Oculus Meta Quest 2 is a standalone virtual reality headset developed by Meta (formerly Facebook). It features a high-resolution display, built-in sensors for tracking head movement, and hand controllers for interacting with virtual environments. The Meta Quest 2 provides a wireless VR experience, allowing users to move freely within a virtual space without being tethered to a computer. It also offers a variety of VR applications and games through the Meta Store.

Figure 3.2: Oculus Meta Quest -2

Setting up an Oculus Quest 2 headset is straightforward and can be completed in a few simple steps. Here's a clear guide to help you through the process:

Step 1: Unboxing
Carefully remove the Oculus Quest 2 headset and all included accessories from the packaging.

Step 2: Charging
Before you begin setup, ensure that your Oculus Quest 2 headset is fully charged. Use the provided USB-C charging cable and power adapter to connect the headset to a power source and allow it to charge.

Step 3: Download the Oculus App

Open the App Store or Google Play Store on your smartphone or tablet.

Search for "Oculus" and download the Oculus app to your device.

Once the app is installed, open it to begin the setup process.

Step 4: Turn on your Oculus Quest 2

Press and hold the power button located on the right side of the Oculus Quest 2 headset until the Oculus logo appears on the display.

Step 5: Pairing your Oculus Quest 2 with the Oculus App

Open the Oculus app on your smartphone or tablet.

Follow the on-screen instructions to sign in with your Oculus account or create a new one if you don't already have one.

Select "Pair New Device" from the app menu and follow the prompts to connect your Oculus Quest 2 headset to your device via Bluetooth.

Step 6: Completing the Setup

Once your Oculus Quest 2 is paired with the Oculus app, follow the on-screen instructions to complete the setup process.

You'll be guided through various steps, including selecting your preferred language, connecting to Wi-Fi, and configuring your Guardian boundary (the virtual safety boundary that prevents you from bumping into real-world objects while using the headset).

Step 7: Adjusting the Fit

Put on your Oculus Quest 2 headset and adjust the straps to ensure a comfortable and secure fit. Make sure the lenses are aligned with your eyes for the best viewing experience.

Step 8: Exploring VR

Once setup is complete, you're ready to explore the world of virtual reality with your Oculus Quest 2 headset! Use the Oculus app to browse and download VR experiences, games, and apps from the Oculus Store.

That's it! You've successfully set up your Oculus Quest 2 headset and are ready to dive into the immersive world of virtual reality. Enjoy your VR experience!

3.3 Selection of meeting platforms

The selection of meeting platforms for comparison in this study was based on several key criteria aimed at ensuring a comprehensive and meaningful analysis of platforms suitable for office collaboration in the metaverse. The following factors were considered during the selection process:

1. **Popularity and Adoption:** The selected platforms, including Oculus Meta Quest 2 with MeetinVR, Google Meet, Zoom, and Nvidia Omniverse, are widely recognized and extensively used for virtual collaboration. Their popularity ensures that the study's findings will be relevant and applicable to a wide range of users and organizations.

2. **Features Relevant to Office Collaboration:** Each platform offers a unique set of features that are particularly relevant to office collaboration. For example, Oculus Meta Quest 2 with MeetinVR provides immersive virtual meeting spaces, while Google Meet and Zoom offer video conferencing features such as screen sharing and chat. Nvidia Omniverse, on the other hand, focuses on collaborative design and visualization tools.

3. **Integration with Oculus Meta Quest 2**: One of the key objectives of this study is to evaluate the integration of AR/VR platforms with Oculus Meta Quest 2. The selected platforms are known to be compatible with Oculus Meta Quest 2, allowing for a seamless and integrated virtual meeting experience.

4. **Availability and Accessibility**: Accessibility was an important consideration, and the selected platforms are widely accessible to users around the world. They offer various pricing plans and support for different devices, making them accessible to a broad range of users.

5. **Industry Recognition and Use Cases:** The selected platforms are recognized in the industry for their innovative solutions and have been used in various real-world applications. This ensures that the study's findings are based on platforms with proven track records in office collaboration.

Overall, the selection of these AR/VR platforms for comparison was carefully considered to ensure that the study provides a comprehensive and insightful analysis of the effectiveness and usability of different platforms for office collaboration in the metaverse.

a. MeetinVR

MeetinVR is a virtual meeting and collaboration platform designed for use with VR headsets, including the Oculus Meta Quest 2. It provides users with a virtual meeting space where they can interact with colleagues and collaborate on projects in a 3D environment. MeetinVR offers features such as customizable virtual rooms, avatar customization, screen sharing, and interactive whiteboards, making it suitable for virtual office collaboration.

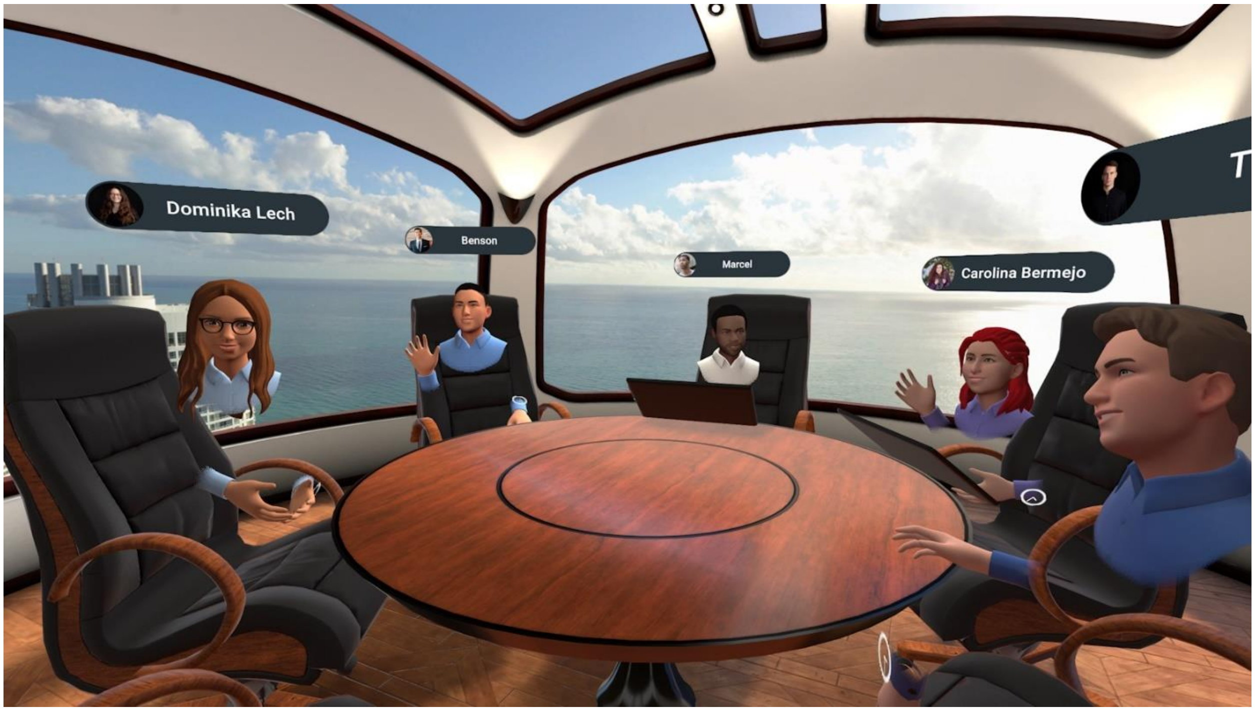
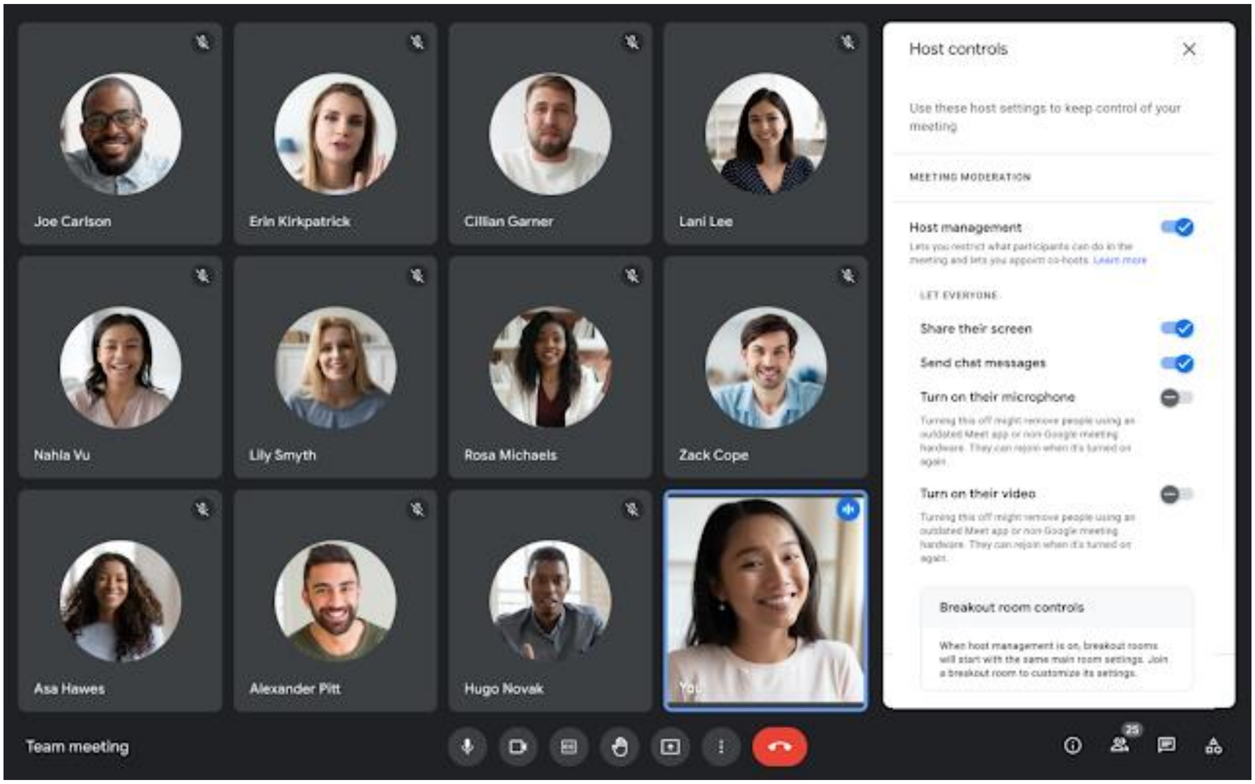


Figure 3.3: Image for Meet in VR

b. Google Meet

Google Meet is a video conferencing platform developed by Google. It allows users to host and join video meetings from a variety of devices, including desktop computers, laptops, and mobile devices. Google Meet offers features such as screen sharing, real-time captions, and integration with other Google Workspace apps. It is widely used for virtual meetings and has become a popular choice for remote collaboration.



Host controls ×

Use these host settings to keep control of your meeting

MEETING MODERATION

Host management ☑
Lets you restrict what participants can do in the meeting and lets you appoint co-hosts. Learn more

LET EVERYONE

Share their screen ☑

Send chat messages ☑

Turn on their microphone ○
Turning this off might remove people using an outdated Meet app or non-Google meeting hardware. They can rejoin when it's turned on again.

Turn on their video ○
Turning this off might remove people using an outdated Meet app or non-Google meeting hardware. They can rejoin when it's turned on again.

Breakout room controls

When host management is on, breakout rooms will start with the same main room settings. Join a breakout room to customize its settings.

Figure 3.32: Google Meet Conference

c. Zoom

Zoom is a video conferencing platform known for its ease of use and reliability. It offers features such as HD video and audio, screen sharing, and virtual backgrounds. Zoom can be used on a variety of devices, including computers, smartphones, and tablets. It has gained popularity for its ability to host large meetings and webinars, making it a popular choice for virtual collaboration.

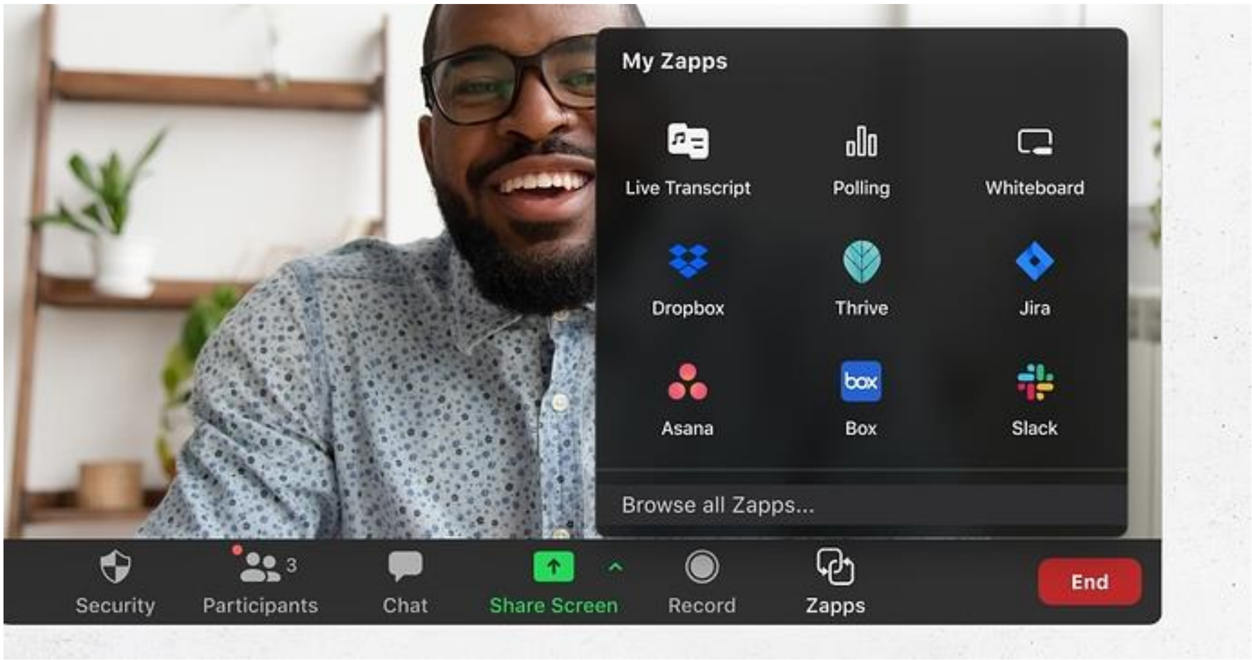


Figure 3.33: Zoom Video Conferencing

3.4 Setup and Configuration of meeting Platforms

Setting up and configuring the AR/VR platforms for office collaboration involves several steps to ensure a seamless and effective virtual meeting experience:

3.4 Setup and Configuration of meeting Platforms

a. Oculus Meta Quest 2 with MeetinVR Setup

1. Ensure that the Oculus Meta Quest 2 headset is properly charged and connected to a stable Wi-Fi network.
2. Download and install the MeetinVR application from the Meta Store.
3. Follow the on-screen instructions to set up your avatar and customize your virtual meeting space.
4. Test the audio and microphone settings to ensure clear communication during virtual meetings.

b. Google Meet Setup

1. Access Google Meet through a web browser or the Google Meet app on your device.
2. Sign in with your Google account and create a new meeting or join an existing one. Configure the audio and video settings to ensure compatibility with your device and network connection.
3. Invite participants to the meeting and manage their permissions as needed.

c. Zoom Setup

1. Download and install the Zoom application on your device.
2. Sign in with your Zoom account or create a new one.
3. Schedule a new meeting or join an existing one.
4. Configure the audio and video settings to optimize your virtual meeting experience.
5. Invite participants to the meeting and manage their permissions.

d. Nvidia Omniverse Setup

1. Download and install the Nvidia Omniverse application on your device.
2. Sign in with your Nvidia account or create a new one.
3. Explore the available virtual environments and select one for your virtual meeting.
4. Customize the virtual environment to suit your meeting requirements.
5. Test the audio and microphone settings to ensure clear communication during virtual meetings.

3.5 Conducting Virtual Meetings and Collaborative Sessions using Meetin VR

Once the platforms are set up and configured, conducting virtual meetings and collaborative sessions can be done as follows:

a. Start a Virtual Meeting:

1. Launch the MeetinVR application on your Oculus Meta Quest 2 headset.
2. Join a virtual meeting room or create a new one.
3. Invite participants to the meeting using their avatars or virtual identities.
4. Use the interactive tools and features available in MeetinVR to collaborate with participants in the virtual environment.

b. Collaborate in Virtual Environments:

1. Use the interactive whiteboards in MeetinVR to brainstorm ideas and visualize concepts.
2. Share screens or presentations with participants to discuss and review documents or designs.
3. Use the avatar customization features to personalize your virtual identity and enhance the sense of presence in the virtual environment.

c. Manage Virtual Meetings:

1. Assign roles and permissions to participants to manage the flow of the meeting.
2. Use the moderator controls to mute or remove participants if necessary.
3. Record the meeting for future reference or share it with participants who could not attend.

d. Wrap Up the Virtual Meeting:

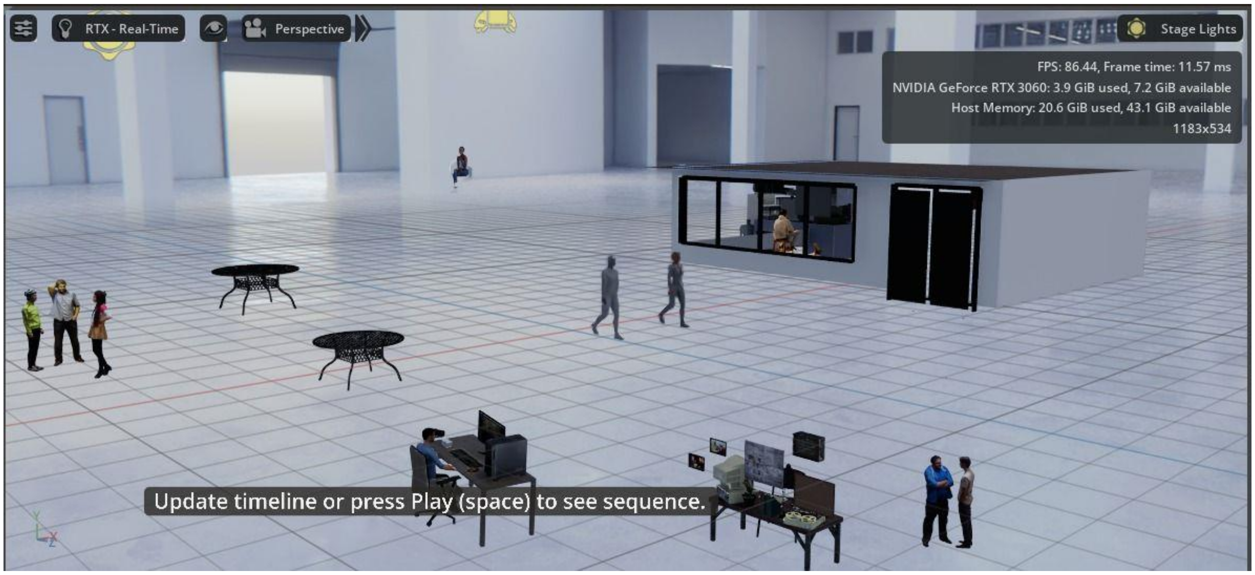
1. End the virtual meeting when the agenda has been covered.
2. Provide participants with a summary of the meeting outcomes and any action items.
3. Gather feedback from participants to improve future virtual meetings.

CHAPTER 4

Results

RESULTS AND DISCUSSION

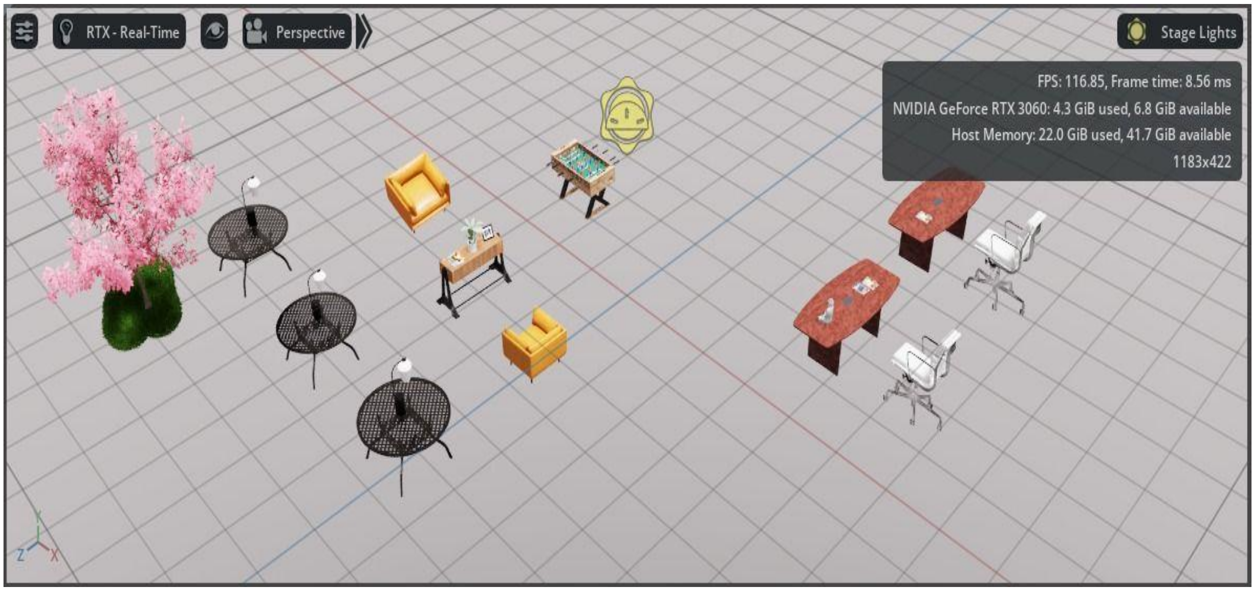
4.1 Office Collaboration using Nvidia Designs



FPS: 86.44, Frame time: 11.57 ms
NVIDIA GeForce RTX 3060: 3.9 GiB used, 7.2 GiB available
Host Memory: 20.6 GiB used, 43.1 GiB available
1183x534

Update timeline or press Play (space) to see sequence.

Figure 4.1: Design -1



FPS: 116.85, Frame time: 8.56 ms
NVIDIA GeForce RTX 3060: 4.3 GiB used, 6.8 GiB available
Host Memory: 22.0 GiB used, 41.7 GiB available
1183x422

Figure 4.12: Design -2

4.2 Comparison of User Experiences

In this section, we analyzed the below meeting platforms based on different aspects.

Platform	User Experience
Oculus Meta Quest 2 with Nvidia Omniverse	Highly immersive with interactive avatars and environments.
Google Meet	User-friendly interface but lacks the immersive experience of VR platforms.
ZOOM	Feature-rich with support for large meetings but less immersive than VR platforms.

Table 4.2: Comparisons of User Experiences

The MeetinVR along with Oculus Meta Quest 2 and Nvidia Omniverse provides the most immersive experience with interactive avatars and environments, setting it apart in user engagement and realism.

4.3 Comparison of Features and Functionality

In this section, we analyzed the below meeting platforms based on their functionalities.

Platform	Features and Functionality
MeetinVR	Interactive whiteboards, screen sharing, and customizable environments.
Google Meet	Basic features like screen sharing, and chat, suitable for standard meetings.
ZOOM	Feature-rich with support for large meetings, breakout rooms, and recording but lacks immersive experience.

Table 4.3: Features and Functionality

MeetinVR stands out for its interactive whiteboards, screen sharing, and customizable environments, offering advanced collaboration tools that enhance creativity and productivity.

4.4 Comparison of Performance and Stability

In this section, we analyzed the below meeting platforms based on their Performances.

Platform	Performance and Stability
Oculus Meta Quest – 2	Stable performance with minimal lag, especially in wireless setups.
Google Meet	Stable performance even in large meetings, reliable infrastructure.
ZOOM	Stable performance with reliable infrastructure, suitable for large-scale meetings.

Table 4.4: Performance and Stability

The Oculus Meta Quest 2 delivers stable performance with minimal lag, especially in wireless setups, ensuring a smooth and uninterrupted virtual meeting experience.

4.5 Comparison of Integration and Compatibility

In this section, we analyzed the below meeting platforms based on their Integrations and Compatibility.

Platform	Performance and Stability
Oculus Meta Quest – 2	Integrates well with MeetinVR for VR collaboration, limited compatibility with other VR platforms.
Google Meet	Broad compatibility with various devices and integrations, suitable for general use.
ZOOM	Compatible with a wide range of devices and software, offers robust integration options.

Table 4.5: Comparison of Integration and Compatibility

Google Meet excels in integration and compatibility with various devices and platforms, making it a versatile choice for users across different environments.

4.6 Comparison of Cost and Accessibility

In this section, we analyzed the below meeting platforms based on their Cost and Accessibility.

Platform	Cost and Accessibility
Oculus Meta Quest – 2	Initial investment for the headset, no additional costs for using MeetinVR. It is easy to access.
Google Meet	Free version with limited features, paid plans available for additional features.
ZOOM	Free version with limitations, paid plans available for more features and larger meetings.
Nvidia Omniverse	Free version for individual users, enterprise licenses may require a significant investment.

Table 4.6: Comparison of Cost and Accessibility

4.7 Comparison of Security and Privacy

In this section, we analyzed the below meeting platforms based on their Security issues and Privacy concerns.

Platform	Cost and Accessibility
Google Meet	Strong security features, compliance with data protection regulations.
ZOOM	Strong security features, encryption and password protection, compliance with data protection regulations.
Meetin VR	Secure virtual meeting environment, encryption of data transmitted during meetings.

Table 4.7: Comparison of Security and Privacy

MeetinVR takes the lead with its secure virtual meeting environment and encryption of data transmitted during meetings, ensuring the utmost privacy and security for users.

CHAPTER 5

Conclusion

5.1 Conclusion

The comparative study of AR/VR platforms for office collaboration has provided valuable insights into their strengths and limitations. Nvidia Omniverse offers advanced 3D design and simulation tools, making it ideal for complex design projects. Oculus Meta Quest 2 provides a wireless and intuitive VR experience, suitable for immersive virtual meetings. MeetinVR offers interactive collaboration tools and customizable virtual environments, enhancing the virtual meeting experience. Google Meet and Zoom are popular video conferencing platforms, but they lack the immersive features of AR/VR platforms.

Based on the findings, we recommend a hybrid approach to office collaboration using a combination of Nvidia Omniverse, Oculus Meta Quest 2, and MeetinVR. This approach leverages the strengths of each platform to create a seamless and immersive virtual meeting experience. Companies should invest in training employees to use these platforms effectively and ensure data security and privacy measures are in place.

5.2 Future Trends

Future Trends and Developments in AR/VR for Office Collaboration:

The future of AR/VR for office collaboration is promising, with several trends and developments on the horizon. These include:

a. **Improved hardware:** Advances in AR/VR hardware will lead to more comfortable and immersive experiences.

b. **Enhanced software capabilities:** AR/VR software will become more sophisticated, offering advanced collaboration tools and features.

c. **Integration with AI:** AI integration will enhance user interactions and automate certain tasks in virtual meetings.

d. **Enhanced security and privacy:** AR/VR platforms will continue to improve security and privacy features to protect user data.

The overall, AR/VR technology has the potential to revolutionize office collaboration by creating a more engaging and productive virtual work environment.

5.3 Future Scope

Metaverse for Various Applications:

a. **Health care accessibility and Telemedicine**: Reach of medical services through immersive telemedicine experiences.

b. **Virtual Teleportation and Presence**: Virtual teleportation enables fluid movement between virtual environments, while presence imbues users with a sense of immersion and interaction within the metaverse.

c. Further research can be carried out to analyze the Technological Advancements and Innovation.

APPENDIX A

Detailed specifications of the hardware components (e.g., Oculus Quest 2 headset) and software tools (e.g., NVIDIA Omniverse) used in the project, including models, versions, and configurations is listed below.

a. **Oculus Meta Quest – 2**:

1. **Display**:

Type: Fast-switch LCD

Resolution: 1832 x 1920 pixels per eye (3664 x 1920 pixels combined)

Refresh Rate: 72Hz (with an experimental 90Hz mode)

2. **Processor**: Qualcomm Snapdragon XR2 Platform
3. **Memory**: RAM: 6GB
4. **Storage**: 256GB Storage
5. **Audio**:

Integrated speakers with spatial audio support

3.5mm audio jack for external headphones

6. **Tracking**:

Inside-out tracking using Oculus Insight technology

Sensors: Integrated accelerometer, gyroscope, and magnetometer

7. **Controllers**:

Oculus Touch controllers with hand tracking support

Buttons: Trigger, Grip, A/B/X/Y buttons, Thumb sticks, Menu button.

8. **Connectivity**:

Wi-Fi 6 support and Bluetooth 5.0 support

USB-C port for charging and data transfer

9. **Battery**: Built-in rechargeable Lithium-ion battery
10. **Weight**: Approximately 503g (without facial interface)
11. **Operating System**: Oculus Quest 2 runs on the Oculus platform, based on a modified version of Android.
12. **Content Access**:

Access to the Oculus Store for downloading and purchasing VR games, apps, and experiences.

Compatibility with PC VR games through Oculus Link or Air Link (with a compatible gaming PC).

b. **NVIDIA Omniverse**:

To run NVIDIA Omniverse effectively, your system should meet certain hardware requirements to ensure smooth performance and optimal rendering capabilities. Here are the recommended hardware requirements:

1. **Processor (CPU)**: Recommended: Intel Core i7 or i9 processor (or equivalent AMD Ryzen processor).
2. **Graphics Card (GPU)**: Recommended: NVIDIA RTX series GPU (e.g., RTX 2080, RTX 3080, RTX 3090).
3. **Memory (RAM)**: Recommended: 16GB or higher RAM
4. **Storage**: Recommended: Solid State Drive (SSD) with sufficient storage capacity
5. **Operating System**: Recommended: Windows 10 or later, or Linux (Ubuntu)
6. **Internet Connection**: Recommended: High-speed internet connection
7. **Additional Peripherals**: VR Headset (optional): If you plan to use Omniverse for virtual reality experiences, a compatible VR headset (e.g., Oculus Rift, HTC Vive) may be required.
8. **External Monitors**: Additional monitors can enhance productivity and multitasking capabilities when working with Omniverse.

References

[1] Greener, R. (2022, December 19). Virtual reality statistics to know in 2023. XR Today. Retrieved February 5, 2023, from https://www.xrtoday.com/virtual-reality/virtual-reality-statistics-to-know-in-2023/

[2] Caserman, P., Garcia-Agundez, A., Konrad, R., Göbel, S., & Steinmetz, R. (2018). Real-time body tracking in virtual reality using a Vive Tracker. Virtual Reality, 23(2), 155–168. https://doi.org/10.1007/s10055-018-0374-z

[3] Alsop, T. (2022, February 21). AR/VR headset shipments worldwide 2019-2023. Statista. Retrieved February 5, 2023, from https://www.statista.com/statistics/653390/worldwide-virtual-and-augmented-reality-heads et-shipments/

[4] Dincelli, E., & Yayla, A. (2022). Immersive virtual reality in the age of the metaverse: A hybrid-narrative review based on the technology affordance perspective. The Journal of Strategic Information Systems, 31(2), 101717. https://doi.org/10.1016/j.jsis.2022.101717

[5] Bowman, D. A., & McMahan, R. P. (2007). Virtual reality: How much immersion is enough? Computer, 40(7), 36–43. https://doi.org/10.1109/mc.2007.257

[6] Z. Huang, H. Wang and X. Shi, "The Future of Work: Do We Need Interactive Collaborative Offices? The Most Interesting Survey in a Post-Covid World," 2023 IEEE 43rd International Conference on Distributed Computing Systems Workshops (ICDCSW), Hong Kong, 2023, pp. 115-120, doi: 10.1109/ICDCSW60045.2023.00023.

[7] L. Karlsson and M. Shamoun, ‘Virtual Realities for Remote Working: Exploring employee’s attitudes toward the use of Metaverse for remote working’, Dissertation, 2022.

[8] A. M. Al-Ghaili et al., "A Review of Metaverse’s Definitions, Architecture, Applications, Challenges, Issues, Solutions, and Future Trends," in IEEE Access, vol. 10, pp. 125835-125866, 2022, doi: 10.1109/ACCESS.2022.3225638.

[9] Sin-nosuke Suzuki, Hideyuki Kanematsu, Dana M. Barry, Nobuyuki Ogawa, Kuniaki Yajima, Katsuko T Nakahira, Tatsuya Shirai, Masashi Kawaguchi, Toshiro Kobayashi, Michiko Yoshitake, Virtual Experiments in Metaverse and their Applications to Collaborative Projects: The framework and its significance, Procedia Computer Science, Volume 176, 2020, Pages 2125-2132, ISSN 1877-0509, https://doi.org/10.1016/j.procs.2020.09.249.

References

[10] Davis, Alanah; Murphy, John; Owens, Dawn; Khazanchi, Deepak; and Zigurs, Ilze (2009) "Avatars, People, and Virtual Worlds: Foundations for Research in Metaverses," Journal of the Association for Information Systems, 10(2), DOI: 10.17705/1jais.00183

[11] Freeman, G. (2016). Intimate experiences in virtual worlds: The Interplay among hyperpersonal communication, avatar-based systems, and experiential drives. IConference 2016 Proceedings. https://doi.org/10.9776/16239

[12] Adams, D., Bah, A., Barwulor, C., Musaby, N., Pitkin, K., & Redmiles, E. M. (2018, August). Ethics emerging: the story of privacy and security perceptions in virtual reality. In SOUPS@ USENIX Security Symposium (pp. 427-442).

[13] Meta quest pro: Our most advanced new VR headset | meta store. (2023). Retrieved March 1, 2023, from https://www.meta.com/nl/en/quest/quest-pro/

[14] Andrews, G.E and D.Bradley (1972) The Burning Velocity of Methane- Air Mixtures, *Combustion & Flame*, 19, 275-288.

[15] J. U. Duncombe, Inrared navigation – Part I: An Assessment of feasi- bility, *IEEE Trans. Electron Devices*, Vol. ED-11, No.1, 34-39, Jan 1959

[16] Lefebvre, A. H., (1965) Progress and Problems in Gas Turbine Com- bustion, *10th Symposium (International) on Combustion*, The Combustion Institute, Pittsburg, 1129- 1137.

[17] Developing on Nvidia Omniverse - How to Build an App — kit-project-template 105.0 documentation

[18] https://developer.nvidia.com/omniverse/get-started

[19] NVIDIA