OAMK OULUN AMMATTIKORKEAKOULU

Juho Vatanen

EXPLORING NVIDIA OMNIVERSE ECOSYSTEM

EXPLORING NVIDIA OMNIVERSE ECOSYSTEM

Juho Vatanen
Opinnäytetyö
Kevät 2024
Tietotekniikan tutkinto-ohjelma
Oulun ammattikorkeakoulu

TIIVISTELMÄ

Oulun ammattikorkeakoulu
Tietotekniikan tutkinto-ohjelma, Ohjelmistokehityksen suuntautumisvaihtoehto

---

Tekijä(t): Juho Vatanen
Opinnäytetyön nimi: NVIDIA Omniverse-ekosysteemin tutkiminen
Työn ohjaaja(t): Samu Teittinen
Työn valmistumislukukausi ja -vuosi: Kevät 2024 Sivumäärä: 40

Tämä opinnäytetyö tutkii NVIDIA Omniverse-ekosysteemiä joka on rakennettu käyttäen Pixarin julkaiseman avoimen lähdekoodin Universal Scene Descriptorin (USD) ja sen hyödyntämistä reaaliaikaisen yhteistyön tehostamisessa.
Keskeisenä toteutustapana on ollut käyttää Omniverse Code ja Omniverse USD Composer työkaluja ja niiden dokumentaatiota sekä Remedy Entertainment Oyj:n julkaisemaa Book of USD:ta.
Tutkimusmetodologia sisältää yhdistelmää teoreettista analyysia ja käytännön demonstraatioita yhteistoimivuuden arvioimiseksi Omniverse-ympäristöissä.
Tärkeimmät havainnot viittaavat siihen, että USD parantaa merkittävästi virtuaaliympäristöjen yhteistoimivuutta ja tämä tehostaa reaaliaikaisen yhteistyötä digitaalisen sisällön luomisessa. Myös monimutkaisten virtuaalisten ympäristöjen rakentaminen on tehokkaampaa skaalautuvan kehitysympäristön ansiota. Omniverse parantaa näitä ominaisuuksia reaaliaikaisilla renderöinti- ja synkronointiominaisuuksilla.
Opinnäytetyön lopussa annetaan suosituksia näiden työkalujen jatkokehityksestä sekä mahdollisista tulevaisuuden sovelluksista eri toimialoilla, kuten peli, animaatio, elokuva ja teollisuuden alalla.

Asiasanat: Universal Scene Descriptor, 3D-mallin tietorakenne, Omniverse, Raytracing

ABSTRACT

Oulu University of Applied Sciences
Degree Programme in information technology, Option of Software development orientation

---

Author(s): Juho Vatanen
Title of thesis: Universal Scene Descriptor standard comparison
Supervisor(s): Samu Teittinen
Term and year when the thesis was submitted: Spring 2024
Number of pages: 40

In this thesis we are investigating integration and benefits of NVIDIA Omniverse in enhancing real-time collaboration across platforms. As digital collaboration becomes even more important in technology-driven industries, tools like USD and Omniverse are central to streamlining workflows and promoting effective interaction between remote teams.
Thesis is evaluation effectiveness of these tools in real-time virtual environment creation.

The research has combination of theoretical analysis and practical demonstrations for performance and integration evaluation of the USD in Omniverse environments.
Key findings are that USD improves interoperability of virtual scenes and collaboration efficiency. Omniverse enhances these with real-time ray tracing rendering and integration capabilities to industrial tools.

At the end of this thesis are recommendations for exploring further of these tools and their possible applications in different industries. Integration of USD and Omniverse is assumed to be transformative approach to promote real-time collaboration in digital content creation.

Keywords: Universal Scene Descriptor, 3D model data structure, Omniverse, Raytracing

CONTENTS

VOCABULARY 7

1 INTRODUCTION 8

1.1 Thesis Objectives and research question 8

1.2 Research methodology, and limitations 9

1.3 Thesis Structure 9

2 DEFINITION OF UNIVERSAL SCENE DESCRIPTOR 11

2.1 Issues leading to development of Universal Scene Descriptor 11

2.2 Breaking down structure of Universal Scene Descriptor 12

2.3 Performance implications of Universal Scene Descriptor 13

2.4 Industries adopting Universal Scene Descriptor 14

2.5 Benefits and Challenges of Universal Scene Descriptor 15

2.5.1 Benefits of Universal Scene Descriptor 15

2.5.2 Challenges of Universal Scene Descriptor 16

2.6 The near future of Universal Scene Descriptor 16

3 OVERVIEW OF NVIDIA OMNIVERSE ECOSYSTEM 18

3.1 Omniverse USD Composer 19

3.2 Exploring available Omniverse connectors 20

3.3 Integrated Development Environment with Omniverse Code 21

3.4 Virtual Reality interaction with CreateXR 21

3.5 Omniverse real-time collaboration with Live sync 21

4 DEVELOPED PROJECTS WITH OMNIVERSE 24

4.1 Omniverse to Teams integration 24

4.2 Omniverse Replicator for synthetic data generation 25

5 DEVELOPING CUSTOM TOOLS FOR OMNIVERSE 28

5.1 The goal, purpose, and methods of development work 28

5.2 Omniverse extension structure 29

5.3 Exploring Extensions window 30

5.4 Visual Studio Code Link 31

5.5 Software development languages 33

5.6 Webhook integration example project 33

6 CONCULSION 36

REFERENCES.....38

VOCABULARY

USD: Universal Scene Descriptor

MDL: NVIDIA Material Definition Language

RTX: NVIDIA's Ray Tracing

IoT: Internet of things

VFX: Visual Effects

OS: Operating System

Scene graph: Graphical scene data structure.

BIM: Building information data

CAD: Computer-aided design

EV: Electric Vehicle

PyPi: Python Package Index

1 INTRODUCTION

In recent years, we have seen great increase in remote collaboration, as multiple teams are working on a single project. There is growing need of collaboration tools and technologies to allow people work in real-time, regardless of their location (10). Real-time collaboration is particularly important in the context of 3D content creation, as it allows artists, designers and developers create complex and dynamic virtual environments.

Traditional workflow has been to work in silos, meaning that having a toolset that does not collaborate well with other tools and it is limiting collaboration capabilities, as each tool can have their own file formats and data structures. This introduces file conversations and manual synchronization of data making it hard to work efficiently (9).

To combat these challenges, a new generation of tools and assets has been released, most notably Pixar's developed Universal Scene Descriptor, known as USD. It was released as an open-source project in 2016, it is an open standard to represent and interchange 3D scenes (2). A couple of years later in 2020 NVIDIA launched their own collaboration platform called Omniverse. Omniverse is dedicated development platform to work natively with USD, it enables creating larger scenes for visualizations while also allows real-time ray tracing rendering, a capability previously unattainable on such scale. Together, USD and Omniverse tools create a powerful solution for real-time collaboration and data synchronization across various platforms through extensive integration capabilities.

1.1 Thesis Objectives and research question

The main research question of the study is:

1. How can efficient scene synchronization between multiple platforms that use different data formats, and how can global real-time collaboration across different platforms be allowed?

Two sub-research questions that are done to assist to allow answering main research question.

The first one is focusing on understanding about USD – what is it about, what benefits and challenges it has, and who needs it.

2a. How does USD compare to existing formats such as FBX, OBJ, .level and .unity formats.

2b. How other platforms are already integrated to this new ecosystem and shows its benefits and challenges.

Second sub-research question focuses more on leveraging this new technology in actual use:

3. How can custom tools be developed that allow real-time collaboration.

The research approach of the study is described in the following sub-chapter to show how objectives of the study are planned to be achieved and the known limitations for the study.

1.2 Research methodology, and limitations

The study conducted in this thesis is an demonstration assessment, where the use of tools and their results are documented. Additionally, custom Omniverse extensions were created to work in USD scene.
Though the limitations of this thesis should be taken into consideration while interpreting the findings, USD and especially Omniverse are very complex subjects with multiple sub-features; this thesis can only introduce and show a few solutions that are within the scope of this research. A scope limitation becomes necessary to make the study doable within the available time and resources for work.

1.3 Thesis Structure

This thesis starts with theoretical chapters concentrating on describing USD and followed by a chapter about NVIDIA Omniverse and purpose of these, going over basic issues of why they were developed, users, benefits, challenges, and the future.

Then there is a chapter about breaking down the development work plan, by explaining how to leverage USD and Omniverse by creating custom extension. The last actual chapter concludes by providing results of this study and summarizes the answers for research questions.

2 DEFINITION OF UNIVERSAL SCENE DESCRIPTOR

Universal Scene Descriptor was developed by Pixar, it was first introduced at SIGGRAPH 2013, and it was then in 2016 released as open source under modified Apache license (1,2,3). USD is versatile framework designed for robust, scalable, and efficient management and interchange of 3D data (4). USD has also emerged as crucial tool for VFX, animation and gaming industries (2,5).

2.1 Issues leading to development of Universal Scene Descriptor

Current 3D / Digital twin industry is relying on multitude of incompatible files formats as most tools have their own unique data structure (7). This has been attempted to be resolved by creating general-multipurpose file format, such as FBX and OBJ. These formats are still actively used in the game, animation, and movie industries, and with Unity, 3ds Max, Maya filetypes are used as well. Industry sector has BIM data and CAD drawings on industry tools (9). In the end of all these file formats are for presenting scenario information.

FBX and OBJ file formats have been the most popular general-purpose data format in 3D industry, but they have been limited by their lack of support on complex scene structures and advanced features such as non-destructive editing, layering and deferred loading. FBX is versatile and supports range of data, from animations to complex geometries it is a propriety format depending on Autodesk ecosystem. OBJ is much simpler and widely support geometry exchange, it is lacking on support for animations, lighting, and cameras. Making it less suitable for interactive media applications. Both formats struggle on scalability and efficiency on very large datasets and collaboration.

These issues led to the development of USD as a unified framework, as it enables seamless data exchange across various applications and platforms. With seamless data exchange it is possible to have real-time collaboration, and currently good demonstration of it is done with NVIDIA Omniverse.

Another great improvement with USD is that it solves struggle with large-scale and complex scenes that traditional 3D file formats did have (8). USD addressed this with several architectural and design features as it was designed from ground up to provide a highly scalable and performant solution, allowing efficient manipulation and processing of massive scenes and datasets (8, 12, 17).

2.2 Breaking down structure of Universal Scene Descriptor

To demonstrate USD structure, USD scenario using Omniverse USD Composer tool was created, it is structed from simple cube and environment lights. Same scenario is then opened with Omniverse USD Composer and USDView tools.

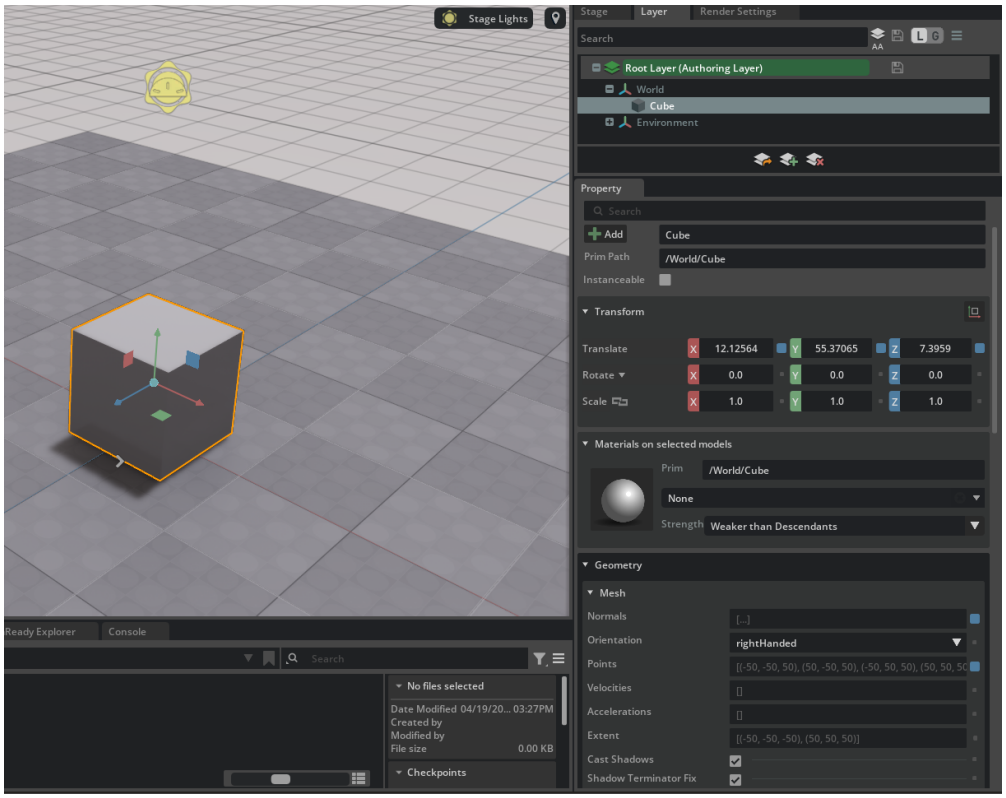

FIGURE 1. Screenshot from Omniverse USD Composer with demonstration scenario.

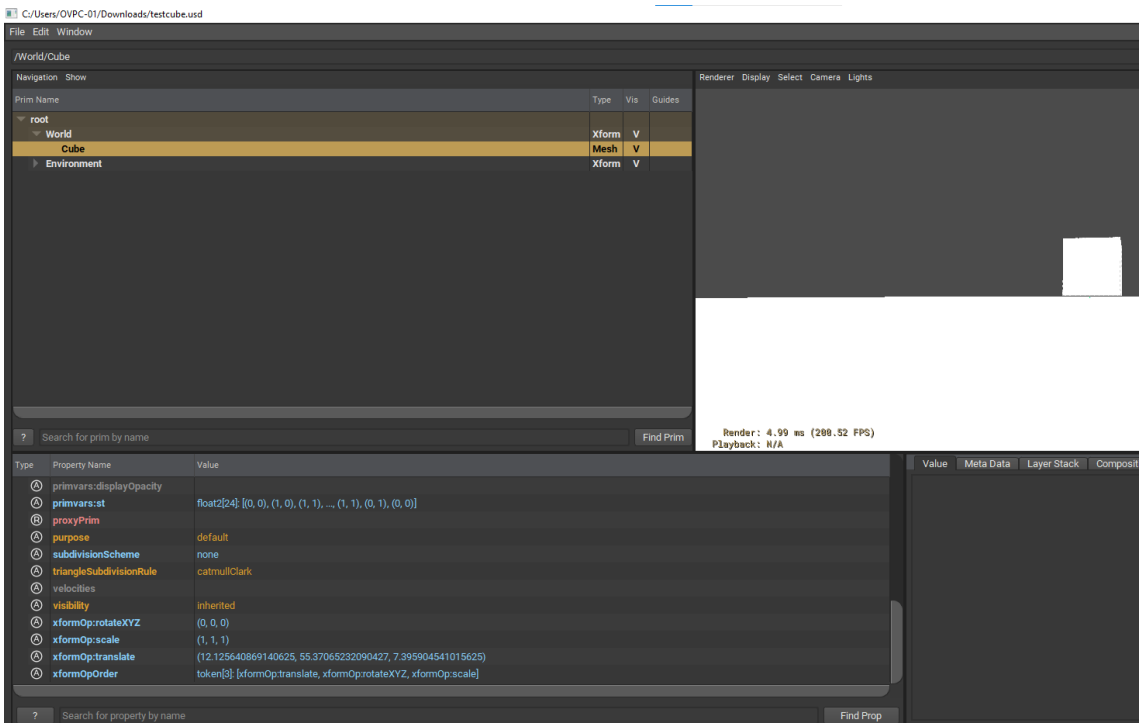

FIGURE 2. Screenshot of USDView tool with demonstration scenario open.

USD Hierarchy is structured as node-based system, where each node represents an object or component in the scene. Each node is a primitive, known also as “prim”. This parent-child relationship can mimic real-world relation dynamic, this can be seen on Figure 1 and 2 by presentation of “World” and “Cube” primitives. This hierarchical arrangement supports effective scene management and scalability.

Primitives can facilitate the definition and manipulation of geometry, material, and metadata enhancing interoperability between different 3D applications. This can be view on Figure 1 on the bottom right corner in “Property” window tab and in Figure 2 property viewer at the bottom.

2.3 Performance implications of Universal Scene Descriptor

Performance improvements from architectural and design choices can be split to three main categories. Data transferring, hardware utilization and storage efficiency.

USD has a feature called payloads that allows parts of the scene to be loaded into memory only when they are needed. This keeps initial memory footprint small and creates improvement on the data transferring.

As USD has been structured around multi-threading capabilities, it allows for better utilization on CPU and GPU resources, and efficient RAM management through non-destructive editing and deferred loading on payloads (30).

With layer and override features, USD can re-use lot of data to reduce storage requirements and more efficient scene storing and retrieval. A good example of this would be creating variants of a cube. In practice there is only one base cube data and material and to make different versions of it, USD overrides only specific material parameter and reuses all other parameters.

2.4 Industries adopting Universal Scene Descriptor

USD started from need from Pixar studio to create extremely complex scenarios where they have millions of models, textures, lights, and colors (15). A good example of this use case was Toy Story 4's Antique Mall scene (15).

Activity around game industry has only recently picked up as game companies have been starting to integrate it to their systems (5,6,13). The most notable one is Remedy Entertainment Oy as they moved their game engine to use USD format in editor, but on creating builds of their product it is converted to more traditional formats (5), This topic was also briefly discussed privately with Remedy employees at an IGDA meetup, and it was noted by them that transforming their engine to use USD had been quite a long journey (14). Other companies such as Tencent has also recently moved their asset development pipeline to USD but are still sticking with more traditional formats for game engine (13).

Software vendors have had stable interest of USD, as they have to kept up with technology as being the enabling party to allow USD integrations, in most cases the software vendor must create themselves an asset bridge, commonly known as connector, for Omniverse/USD. Of course, exceptions exist where software permits the running of addons that are custom code/plugins. Then the possibility arises to create one's own Omniverse connector without assistance from the software vendor (16).

Industries have shown great interest in Omniverse capabilities as it has shown great capabilities for digital twins (17), as existing CAD and BIM visualization tools are not the most performance efficient (12).
BMW has created their new Debrecen EV plant and are able to review it two years before its construction is completed as digital twin in Omniverse platform with the whole car manufacturing system (17).
Apple, that has great relationship with Pixar, has chosen USDZ to be major 3D model format with Apple Vision Pro. It provides capabilities to view USDZ models in spatial space (35, 36). The opportunity to work with Apple Vision Pro glasses has been provided, and working with USDZ data using them can be described to be a good experience.

2.5 Benefits and Challenges of Universal Scene Descriptor

Beside benefits of USD mentioned earlier such as improved collaboration, efficiency, streamlined workflow and already surprisingly wide adoption of technology in animation, movie, and game industry. There are also challenges of it as it is still relatively fresh technology.

2.5.1 Benefits of Universal Scene Descriptor

USD presents several advantages over traditional file formats such as OBJ, FBX, unity, which are commonly used in video game, animation and VFX industries (25). As USD not only serves as a comprehensive interchange file format but also offers powerful composition and performance features that distinguish it from other formats (25).

Firstly, USD interchange capabilities that surpasses previous file formats by interchanging not only geometry and textures, but also shading, materials, rendering, linear-blend skinning and blend-shape animation, and rigid body physics. Furthermore, USD is extensible along numerous axes, ensuring its adaptability to various use cases and requirements (25).

Secondly, efficient asset sharing, and reusability were achieved, as elemental assets can be reused across multiple instances and scenes, resulting in more efficient asset management and reduced storage requirements. This benefit leads to cost saving, improved performance in the pipeline (25).

Thirdly, unlike traditional file format USD encodes Composition Arcs, which allow for the efficient assembly of small or large scene components into complex environments and complete scenes. This allows assets to be shared between instances and scenes, maintaining their integrity through USD's non-destructive editing capabilities (25).

Fourthly is USD customization and extensibility possibilities, it offers numerous plugin points, allowing for customizable asset-resolution behavior, the addition of new data schemas and associated imaging behaviors. This flexibility makes it easier to adapt USD to existing pipelines and infrastructure, ensuring a smoother integration process (5, 11, 25).

Fifthly USD's platform-agnostic design, meaning its enabling interoperability across different OS's and hardware configurations. Currently focus from most industries is on Linux operating system to maximize performance, but Windows support is widely used to run Omniverse platform on the local OS (22).

2.5.2 Challenges of Universal Scene Descriptor

As USD is still relatively fresh technology, it is still ongoing a lot of adoption and integrations. This can be challenging for companies without USD/Omniverse skilled personal to transfer their legacy pipeline over to USD (13), and pipeline users must learn a new set of skills (13).

And as a new technology there is still ongoing evolvement of its feature set, so all parties integrating to USD must actively keep eyes on its development to guarantee forward compatibilities and as feature set is not completed yet, it might still be missing features that were provided by propriety or application-specific file formats. This is currently pushed by NVIDIA to allow full Metaverse capabilities inside USD (21).

2.6 The near future of Universal Scene Descriptor

Since USD was released open-source initiative, anyone has had ability to contribute on the USD standardization and on the source code. It was still maintained and controlled by Pixar's USD team to guide its development and roadmap plans for future development (3,4). Alliance for OpenUSD

(AOUSD) was established shortly before SIGGRAPH 2023 event as a non-profit organization dedicated on developing, promoting and improving the standardization of USD (39).
Another big player in development of USD is NVIDIA, as they have created large ecosystem, Omniverse, top of existing USD and it will be their interest to influence its development to fit their vision (21).

3 OVERVIEW OF NVIDIA OMNIVERSE ECOSYSTEM

Before proceeding further, it is necessary to clarify what NVIDIA Omniverse is. As most adopters in later chapter have multiple mentions of it and likely work with it in at least some capabilities. In 2020 NVIDIA announced Omniverse platform as early access program and its suite of apps (11). Current licensing system allows up to 2 users collaborating without additional license requirements for free, but if project consist of 3 or more users, it will require enterprise license (24).
In January 2024, NVIDIA announced in change of licensing model, there is now limitation of maximum of 2 GPUs on individual license and enterprise licenses are bought by per GPU (37).

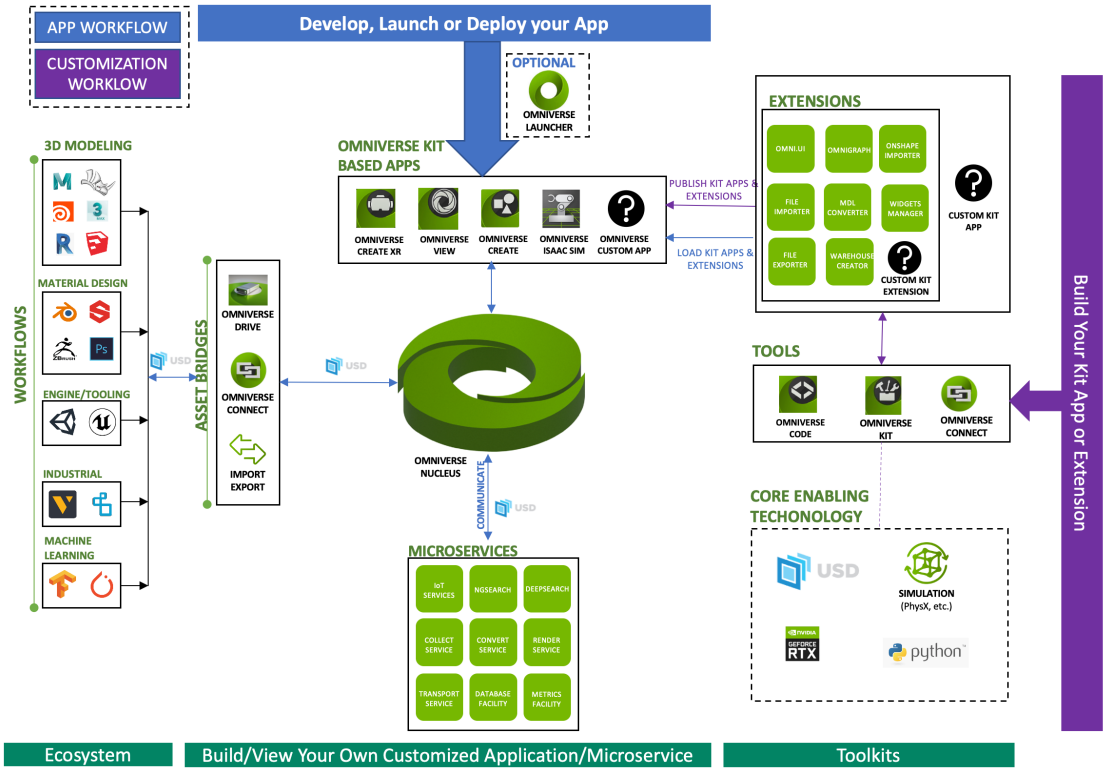

FIGURE 3. An illustration of Omniverse ecosystem overview (38)

Omniverse is a platform designed from extensions and microservices first approach, for example Omniverse Create -- recently rebranded as Omniverse USD Composer(23) -- is a single app composed of 300 extensions (22), and those extensions are Python scripts allowing almost total customization of the app (22).
With this microservice design, it is possible to create custom extensions, or a new kit/app can be created from scratch by selecting only the necessary extensions as shown on the Figure 3 (11, 22).

For this type of development, they provide dedicated tool for extension development called NVIDIA Omniverse CODE.

Omniverse provides multiple features out of the box for their ecosystem, example they provide onsite Nucleus server.
Nucleus works like a shared disk/database, and it enables ability to activate live sync across all connected clients to Nucleus and allowing real-time collaboration.

In this thesis, the focus will particularly be on Omniverse Code, as it will serve as Integrated Development Environment (IDE). But also grasping some core technologies and software, including Live sync feature for collaboration, USD Composer for scene creation, Nucleus as file storage, and core enabling technology focusing on USD and Python. And its Extension and microservice approach.

3.1 Omniverse USD Composer

NVIDIA's Omniverse USD Composer is their main tool for content creation and collaboration of 3D scenarios, known previously as Omniverse Create (23), this can still be seen in most branding footage.
Its primary purpose is to provide a versatile and efficient workspace, enabling artists, designers, and developers to build, edit, and collaborate on complex 3D scenes and projects.

Using it reminds other 3D world manipulation tools for scene creation, such as Unity and Unreal Engine. It also has integrated NVIDIA 3D model and PBR material libraries, so picking it up and creating scene is very straight forward. It also has features such RTX Rendering, Animations, Simulations, and effect system. But when comparing from other game engines, the biggest differences Composer are micro-service design, and stage and layer hierarchy of USD.

3.2 Exploring available Omniverse connectors

NVIDIA has developed multiple connectors alone and with support from software vendors for existing tools and software's that are shown on the above figure. Support for de-facto industry tools like Autodesk Revit, 3ds Max, Maya, Blender, Unity, and Unreal Engine is available (11).
Some of these Connectors are still early access/developers only.

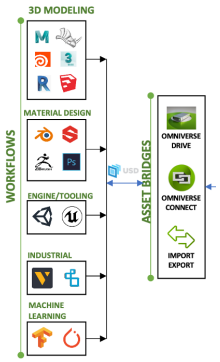

FIGURE 4. A screenshot of some available connectors (38).

The purpose of Connector is to access, either by importing or exporting from/to Nucleus Server. There are couple of asset bridges available based on how the Connector has been developed.

First there are omni-directional connectors, most capable connector method as these omni-directional connectors can load and save data on Nucleus and support OmniLive for live collaboration. For example, a scene can be modified in real-time by adding objects, editing material, and transferring objects either in 3ds Max or Unreal Engine, or from USD Composer.

Connectors can be uni-directional, for example, with Revit, where data can only be transferred from Revit to Nucleus and USD/Nucleus data cannot be loaded into the Revit software. These connectors also support OmniLive, but as described earlier, changes must be made in the Connector tool, and changes made in other tools will not be reflected in the Connector.

The Import/Export Connector, currently relevant for Blender, differs from uni-directional or omni-directional connectors in that there is no OmniLive support, and the import and export processes are more static. Blender, along with several other 3D modeling software such as 3ds Max, has supported USD for years before Omniverse, enabling the importing and exporting of USD files.

Connectors have been developed by vendors as well, for example Siemens Tecnomatix Process Simulate has received add-on for uni-directional connector (31). This add-on is provided with installer binary meaning that file exchange logic is hidden behind encryption.

3.3 Integrated Development Environment with Omniverse Code

Code is an app develop on Omniverse platform, specialized as development environment designed for creating Omniverse extensions and applications. In this thesis it serves as main development tool. It offers a streamlined interface, built-in debugging, Visual Studio Code linking, and seamless integration with Omniverse's features, making it an ideal choice for our development work.

3.4 Virtual Reality interaction with CreateXR

CreateXR is Omniverse tool dedicated for VR/XR experience on USD files.
This tool can be combined with other Omniverse tools to create interactive experience for extensive demonstrations. CreateXR uses OpenXR and it requires SteamVR for Varjo-XR3 experience.
In 2023, NVIDIA announced that CreateXR will be discontinued as standalone app and all features are going to be integrated to USD Composer (29). This was possible due to micro-service architecture that allows any feature to be moved between Omniverse kits and apps.

3.5 Omniverse real-time collaboration with Live sync

In the hearth of USD is real-time collaboration and in Omniverse it is integrated as feature called OmniLive, but it is more commonly called as Live Sync or just Live. Live sync creates new layer on the USD. This allows multiple users to join the same session, modifications are stored in session layer that can be either discard or save those modifications to main layer after their live sync session (27).
Live Sync is extension that is enabled on most NVIDIA Tools, including Omniverse USD Composer and USD Presenter tools, and some connectors have added LiveSync feature to allow live collaboration, some connectors that have this capability are Unreal Engine, 3ds Max and Unity.

To demonstrate OmniLive, same demonstration scene is opened as used on Figures 1 and 2 demonstrate USD hierarchy and activated Live mode.

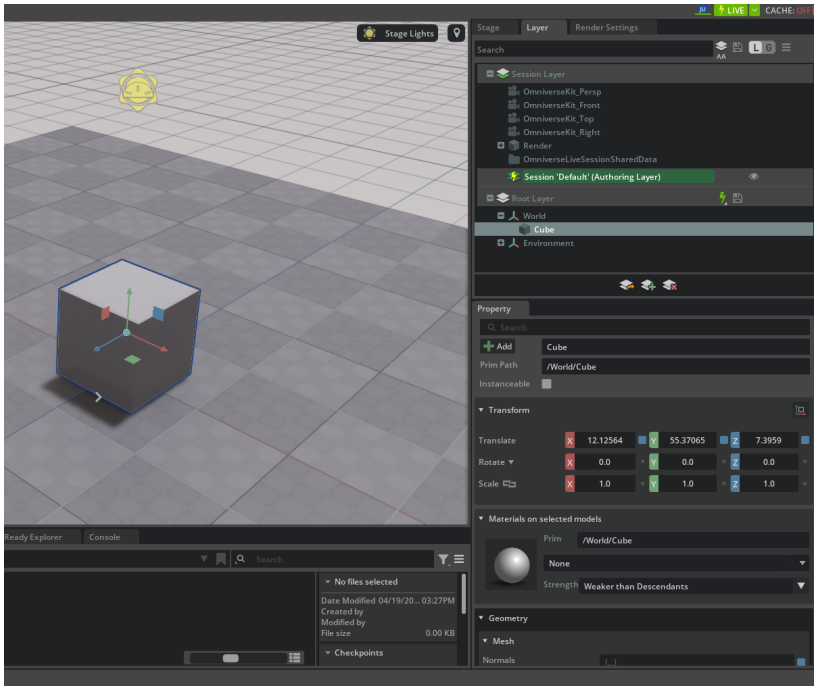

FIGURE 5. A screenshot of Omniverse USD Composer with demonstration scene open and OmniLive feature connected.

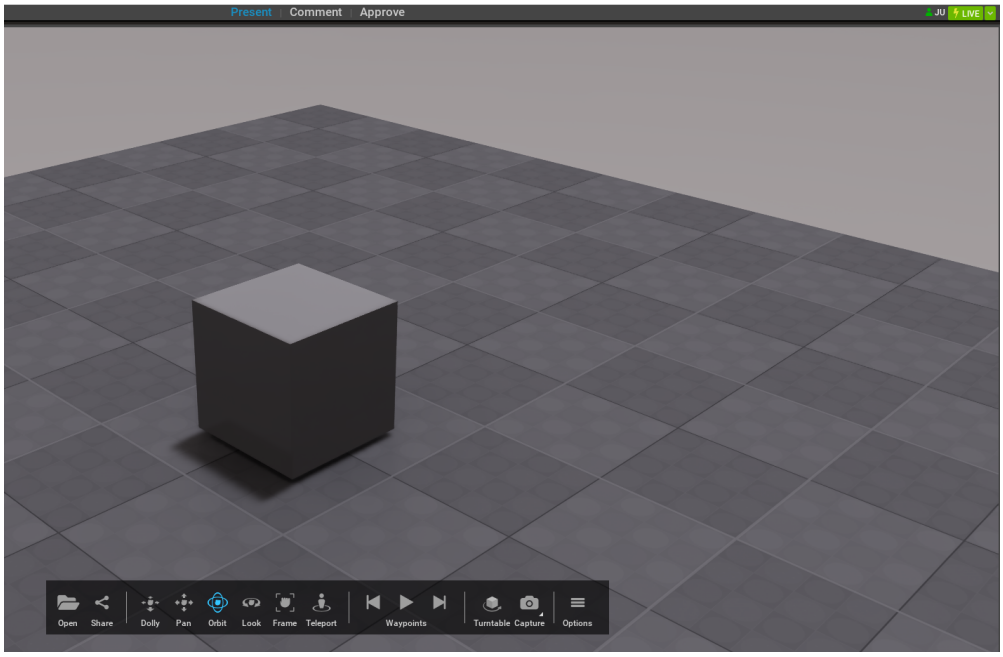

FIGURE 6. Screenshot of Omniverse Presenter with demonstration scene open and OmniLive feature connected.

In the Figure 5, we can notice that Layer window has added Session layer. In it we can see that camera detail prims are added, and any changes done only in Live mode is presented. Any part that has no changes are no details stored in Session layer.

In Figure 6, Live mode is activated as seen on top right corner with list of all other users connected to the Live session. Any changed done in Figure 5 are presented in real time to our Presenter tool on Figure 6.

4 DEVELOPED PROJECTS WITH OMNIVERSE

4.1 Omniverse to Teams integration

Before going over extension development, one of the projects that was developed is NVIDIA Omniverse Team integration. Worked on this project while being part of small team at Accenture Helsinki office which successfully developed an integration for Microsoft Teams and NVIDIA Omniverse.

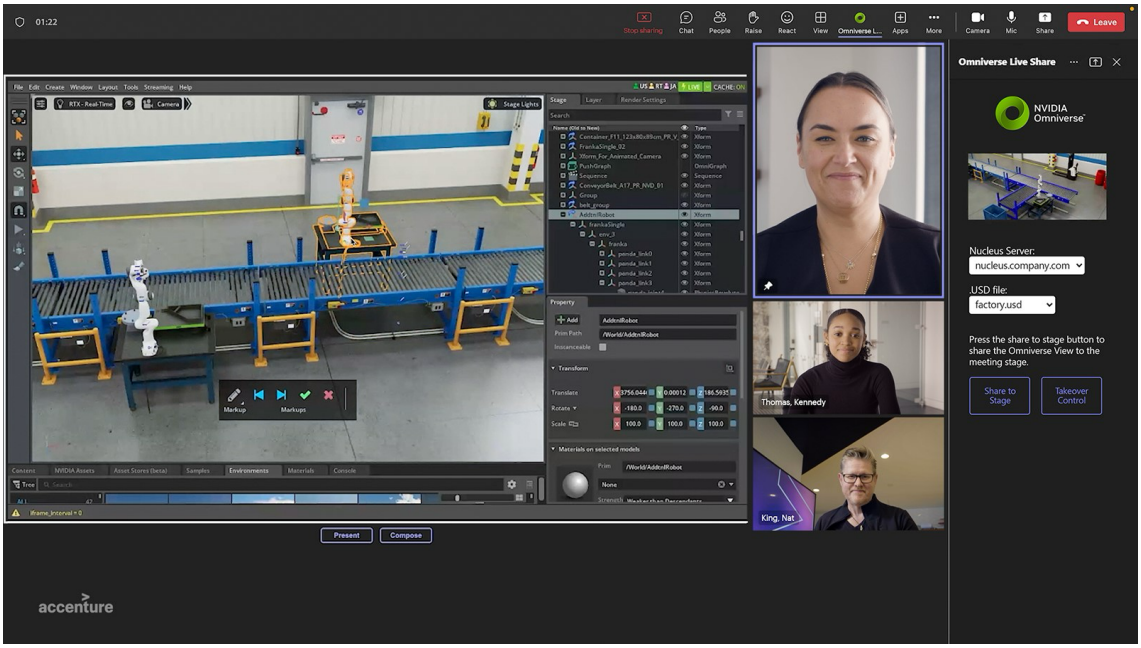

FIGURE 7. Screenshot of Omniverse Team integration (18). Integration developed by me and Accenture Helsinki Innovation Lab.

This solution enabled users to access Omniverse's capabilities from non-RTX capable devices (18, 20) with cloud rendering. The team achievement was also recognized during the NVIDIA GTC 2023 Keynote, where NVIDIA CEO Jensen Huang personally showcased solution (19). This remarkable acknowledgment highlights the potential impact of integration in expanding Omniverse's accessibility to a wider range of devices and users, further promoting collaboration and innovation in the 3D graphics industry.

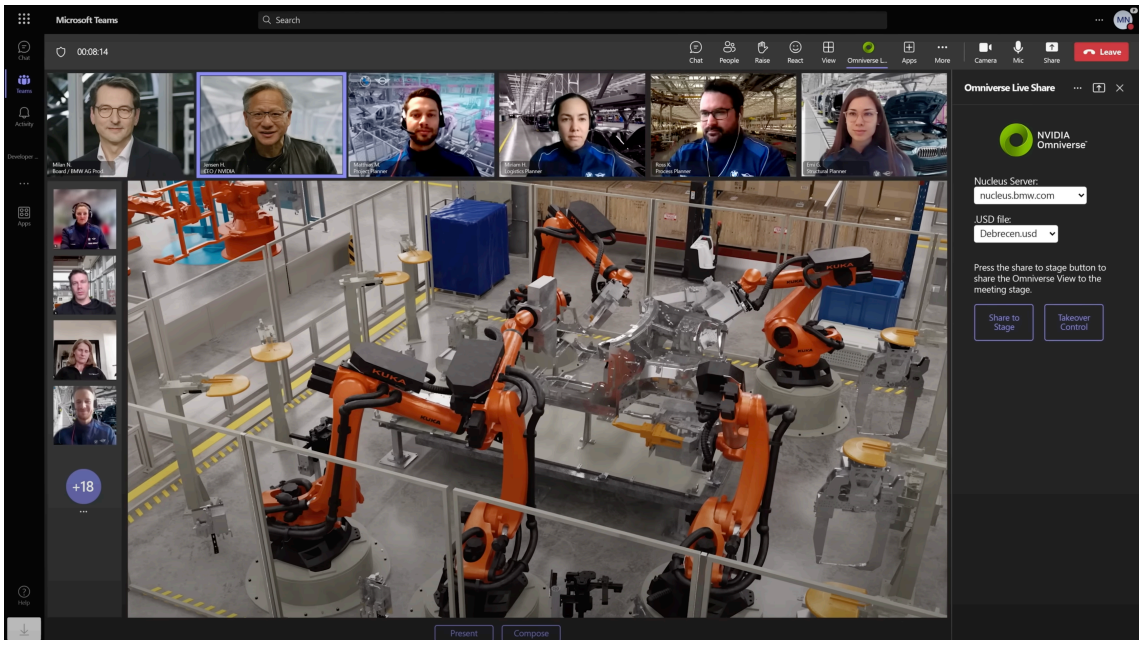

FIGURE 8. Screenshot from NVIDIA GTC Keynote 2023.

Teams Liveshare integration has received quite an amount of publicity on Omniverse field. Above is screenshot from GTC 2023 keynote. In it BMW Member of the Board of Management Dr. Milan Nedeljković and NVIDIA CEO Jensen Huang discussing of Teams and Omniverse integration and possibilities for planning new factories (19. 1:09:22). Keynote is also featuring BMW factory design team.

World Economy Forum released their Insight Report March 2024, as main topic it had industrial metaverse. On the report is mentioned on how Eclipse automation is integrating Dassault Solid-Works with Omniverse and Omniverse Teams Liveshare (33).

Teams Liveshare integration team has also reaped success in Accenture's own V360 Awards, on Company & people category for our market unit and will be presented in global V360 awards in 2024. Couple of development team members also participated GTC in 2024 with latest version of Teams Liveshare demonstration.

4.2 Omniverse Replicator for synthetic data generation

Recently was also created an Omniverse Replicator project. Replicator can create synthetic data for AI object detection training.

The role and task involved developing required scripts and functions for replicator and generate requested data for Yolo V7 trainer.

Using the Nucleus server enabled seamless collaboration between two engineers. As objects were classified for annotations and code was improved and tested, 3D modeling was concurrently being done. Issues with material visualization on Blender, which integrates Omniverse, were addressed by fixing materials in Code or Create, with updates received instantly. Synthetic data was utilized to enhance a major industrial client's AI detection model by merging it with real data. Consequently, the accuracy of the AI model, when trained on a combination of synthetic and real images, improved by large margin compared to training solely on real images. Omniverse Replicator provides capabilities to view scenarios from non-human view as shown in the next figure.

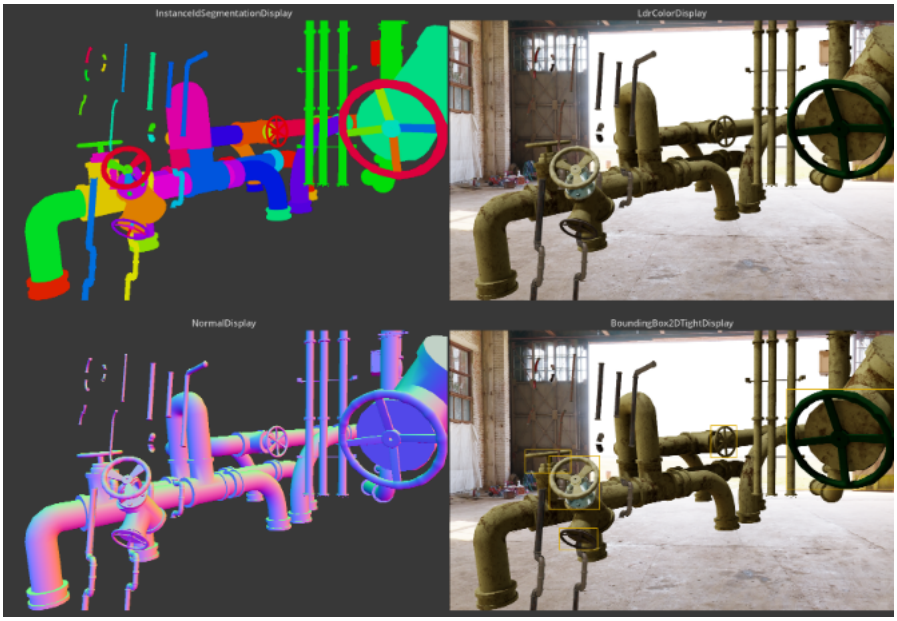

FIGURE 9. Screenshot from Omniverse Code.

In the above screenshot taken from Omniverse Code is shown sensor output layers. In top left is semantic output, top right is RGB layer, this is what regular camera would see. On bottom left we have normal angles layer and on bottom right we have most important output for training object detection, BoundingBox2d view. The bounding box shows live view of annotated objects in the scene.

RGB layer image was written to folder with bounding boxes to be used with Yolo V7 model, all data was then used for AI model training.

5 DEVELOPING CUSTOM TOOLS FOR OMNIVERSE

This chapter demonstrates creating Omniverse extension that can leverage USD capabilities. With Omniverse, amount of work required is decreased as instead of developing software from USD source code. This allows starting working on the features of extension instead of creating boilerplate code USD compatibility.
NVIDIA currently allows free and paid distribution of extensions, this allows creation extension and selling them. NVIDIA's focus has been on selling Omniverse on per GPU license format for teams of 2 or larger. There is marketplace for Omniverse extensions in NVIDIA forums, Extension Exchange and for industrial tools private distribution is currently done by vendors directory. Currently Omniverse community is mainly on NVIDIA forums, at forums.developer.nvidia.com/c/omniverse and on Discord (32).

5.1 The goal, purpose, and methods of development work

The goal of this development work is custom extension to Omniverse platform. By creating Omniverse extension instead of building USD tool from scratch, it saves multiple hours of work on mundane work on the boilerplate USD support. Extension can be then installed on any Omniverse platform app, such as Isaac Sim, Create, View or Replicator. The main purpose is this to work as proof of concept and to allow any further development, and not to release it as a functional product.

The purpose of this extension is to establish a connection between Discord webhook and Omniverse platform, enabling transfer of custom messages derived USD data. By creating this link, it allows to facilitate real-time communication and collaboration between team members whose primary tools are Discord and Omniverse. sending custom messages defined in Omniverse of selected USD object data.

Development of the extension relies on a well-defined set of methods and tools. Development stack includes Omniverse CODE, Visual Studio Code and powerful NVIDIA RTX GPU.
As Omniverse main feature is Ray trace rendering visualization and it is NVIDIA's product, RTX 4080 is used as development platform GPU.

5.2 Omniverse extension structure

An Omniverse extension is a modular component designed to add functionality, tools, or features. In the simplest form, extension is just folder with omniverse.toml file. This toml file is found by Extension system and the system will do anything that the file tells it to do, this can be Python modules, Carbonite plugins, shared libraries or applying settings for example. The concept for the extensions was designed to be easy for integration, management, and distribution.

NVIDIA provides a template extension to easily start development in their GitHub or it can be created from Extensions window (26, 2). In the Figure 10 is "+" button that starts flow to create new extension. This flow is used to create new extension based on the template and some config details are updated, example namespace name and structure.

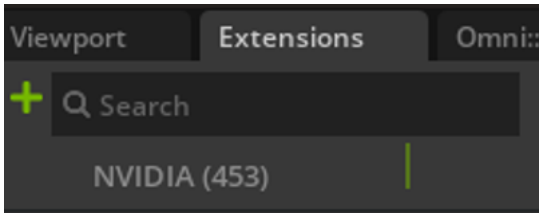

FIGURE 10. Screenshot from Extension window.

In the next figure extension template is extracted. Omniverse extension typically consists of specific directory structure to ensure compatibility and maintainability. This extension template includes placeholder graphics, config, changelog, testing scripts and readme file.
Also on the same figure, folder path is following namespace naming.

```
PS C:\Users\juho.vatanen\Documents\ov\kit-extension\exts> tree /F
Folder PATH listing for volume Windows
Volume serial number is 36D0-B24D
C:.
└───omni.hello.world
    ├───config
    │       extension.toml
    │
    ├───data
    │       icon.png
    │       preview.png
    │
    ├───docs
    │       CHANGELOG.md
    │       README.md
    │
    └───omni
        └───hello
            └───world
                │   extension.py
                │   __init__.py
                │
                └───tests
                        test_hello_world.py
                        __init__.py
```

FIGURE 11. Omniverse extension template visualization with tree /F command on terminal.

5.3 Exploring Extensions window

Extensions window is main tool in Omniverse to control all the active and enabled extensions on the app.

To add and enable additional extensions, settings wheel button is pressed to open settings view and in this view, adding and removing extensions is done by editing Extension Search Paths list.

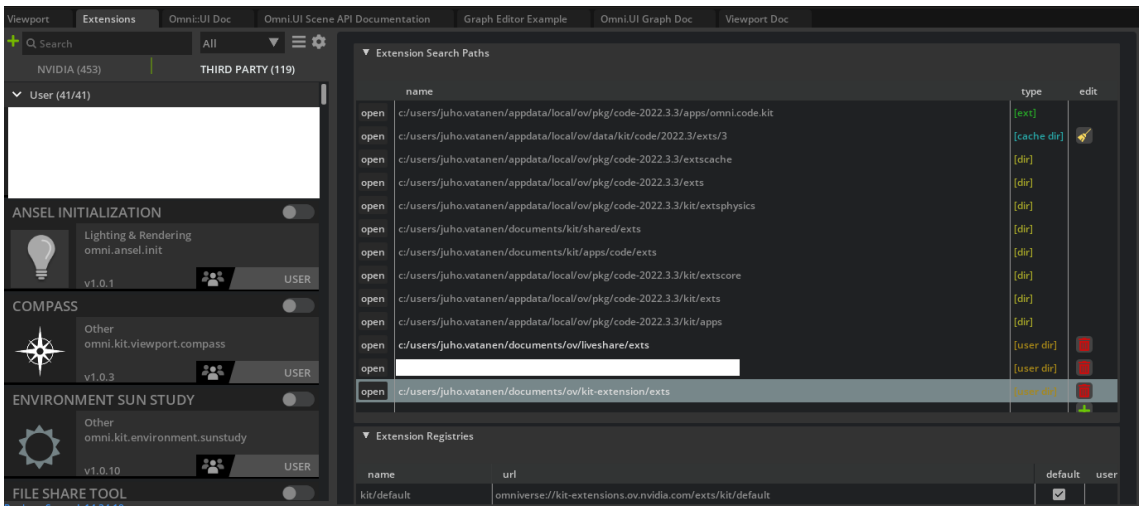

FIGURE 12. Screenshot from Extensions window. In "Extensions Search Paths" setting can be defined paths for more directories. And custom extension registry sources.

Adding extension registries is also possible, as shown bottom of figure 12. This can provide great way to handle propriety extensions in enterprise landscape to allow easy access for in-house developed extensions.

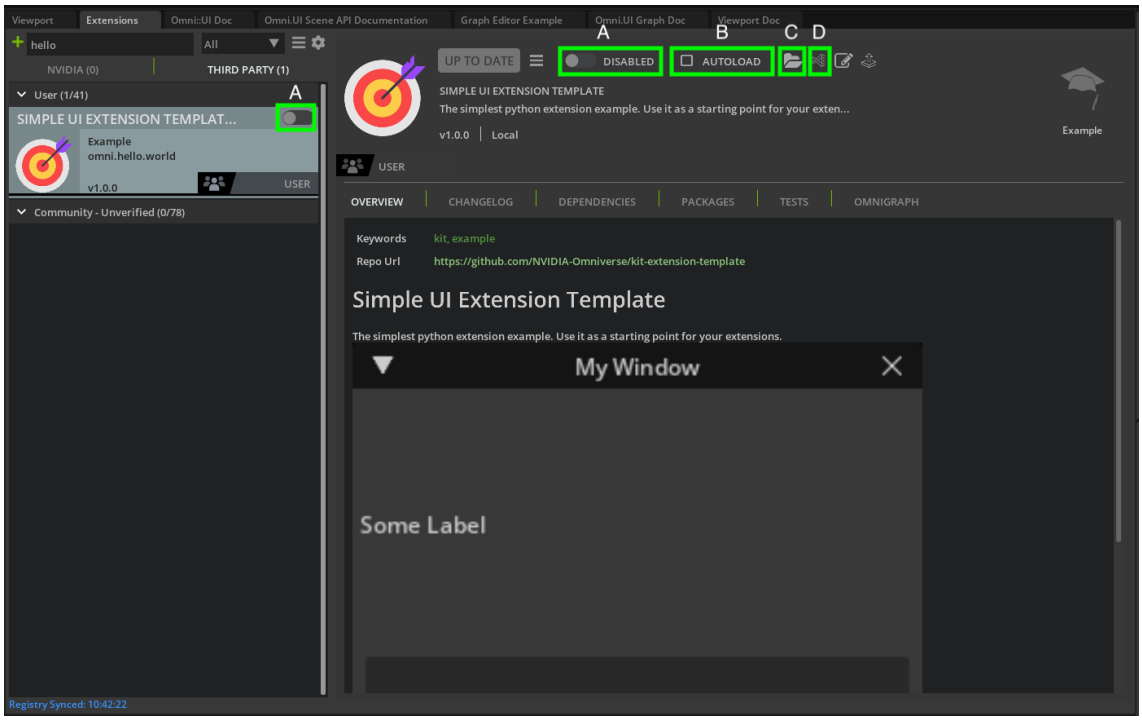

FIGURE 13. Screenshot from Extensions window when template example is selected.

After adding extension path to Extensions Search Paths, Omniverse adds it to extensions list. Using search bar extension can be located and it can be selected. In this view there is many important features, first we got enable/disable toggle button to active this extension (FIGURE 13, A). Enable autoload toggle (FIGURE 13, B). Autoload can be used to automatically enable extension on startup. There are two more buttons especially for developers. The first one is to open extension file directory to locate source code (FIGURE 13, C), and second one is to open extension source code as visual studio code project (FIGURE 13, D).

5.4 Visual Studio Code Link

For easy debugging inside Visual Studio Code, the link between Omniverse CODE and Visual Studio Code needs to be established. Linking allows use of breakpoints, checking variables data and other common debugging features.

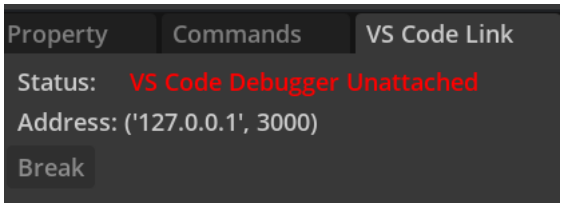

FIGURE 14. By default, VS Code Link window will show unattached as connection to visual studio code is not established.

To connect, inside .vscode folder is launch.json file and it must be configured based on information on VS Code Link window. Connection port for Omniverse should be by default 3000 and that serves debug information to IDE.

```
.vscode > {} launch.json > ...
1  {
2      "version": "0.2.0",
3      "configurations": [
4          {
5              "name": "Python: Attach (windows-x86_64/linux-x86_64)",
6              "type": "python",
7              "request": "attach",
8              "port": 3000,
9              "host": "localhost"
10         }
11     ]
12 }
```

FIGURE 15. Launch.json config updated to match our connection details.

Version included in the extension template used to connect debugger to Omniverse Code does not match our requirements, so on figure 15 we have working version of the configuration file.

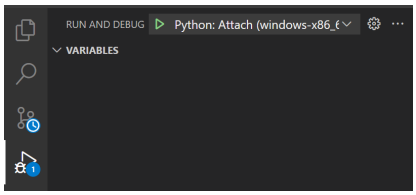

FIGURE 16. Run and debug menu inside vscode, name that is defined inside launch.json is shown as the name and other configurations are under dropdown.

After configuration file has been configured correctly, inside VS Code is a "Run and Debug" menu as shown on figure 16. When green "Play" button is pressed thread information becomes visible

and within Omniverse VS Code Link window indicates that it has been attached to VS Code. This connection confirmation status can be seen on figure 17.

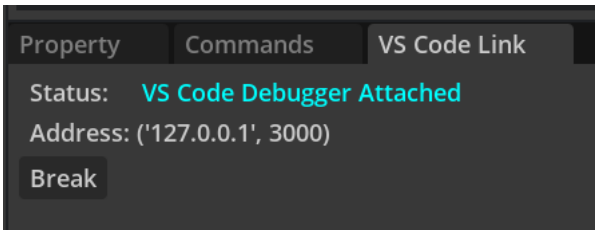

FIGURE 17. VS Code Link window is now also displaying that we have been attached. We are now ready to start debugging our extension.

After these steps, use of breakpoints is working and be used for developing extensions. And also, on Python scripts that are attached to prims as components, that can execute code on specific prims using BehaviourScript based Python scripts.

5.5 Software development languages

Omniverse currently supports developing extensions with Python and C++, but in this case, we only focus for Python development. And in current state there's major security issue with Omniverse that I want to address before going deeper on exact development workflow (34).

As of 2023 August, all Python scripts are executed on system user level, meaning that if you download project from some other party that includes script components, they could include malicious behavior. For example, installing malware. That is why at its current state, you should not open projects from untrusted sources.

5.6 Webhook integration example project

To create webhook, integration platform where webhook is in must be chosen. In this demonstration Discord Webhook is used as it is modern chat platform and is free to use.
Extension base was created using extension window as used in chapter 5.2.

Top of the extension template, Discord integration using Python requests library from PyPi was added. Entire source code for this extension is available on figure 18.

Rest of the code is broken to 4 parts.

Line 20 we define webhook URL parameter.

Lines 21-28 user interface (UI) is defined using omni.ui library and get_selected_name function is set to be called on button press.

Lines 30-44 selected prim details are acquired from stage and we also log if user has not selected prim in the Omniverse scene.

Lines 46-50 webhook is sent to Discord with selected prim name.

```
1   import omni.ext
2   import omni.ui as ui
3   import omni.usd
4   import omni.kit.pipapi
5
6   omni.kit.pipapi.install(
7       package="requests",
8       version="2.31.0",
9       module="requests",
10      ignore_import_check=False,
11      ignore_cache=False,
12      use_online_index=True,
13      surpress_output=False,
14      extra_args=[]
15  )
16
17  import requests
18
19  class SolidjuhoOamkDiscordExtension(omni.ext.IExt):
20      url = "https://discord.com/api/webhooks/1231
21      def on_startup(self, ext_id):
22          self._window = ui.Window("Discord integration", width=300, height=300)
23          with self._window.frame:
24              with ui.VStack():
25                  label = ui.Label("First choose item in stage window")
26
27                  with ui.HStack():
28                      ui.Button("Send Message", clicked_fn=self.get_selected_name)
29
30      def get_selected_name(self):
31          #Get USD Stage
32          stage = omni.usd.get_context().get_stage()
33          #Get selection
34          selection = omni.usd.get_context().get_selection()
35          #Get paths from selectiom
36          selected_paths = selection.get_selected_prim_paths()
37
38          if selected_paths:
39              selected_prim = stage.GetPrimAtPath(selected_paths[0])
40              selection_name = selected_prim.GetName()
41              print(f"Selected : {selection_name}")
42              self.discord_webhook(selection_name)
43          else:
44              print("Please first choose an item.")
45
46      def discord_webhook(self, msg):
47          data = {"content" : f"Selected : {msg}",
48                  "username": "Omniverse Integration"
49          }
50          requests.post(self.url,json=data)
```

FIGURE 18. A screenshot of Discord Webhook integration source code.

In the Figure 19 Discord integration extension user interface is shown on bottom right corner and in stage view HelloWorldCube prim is selected.

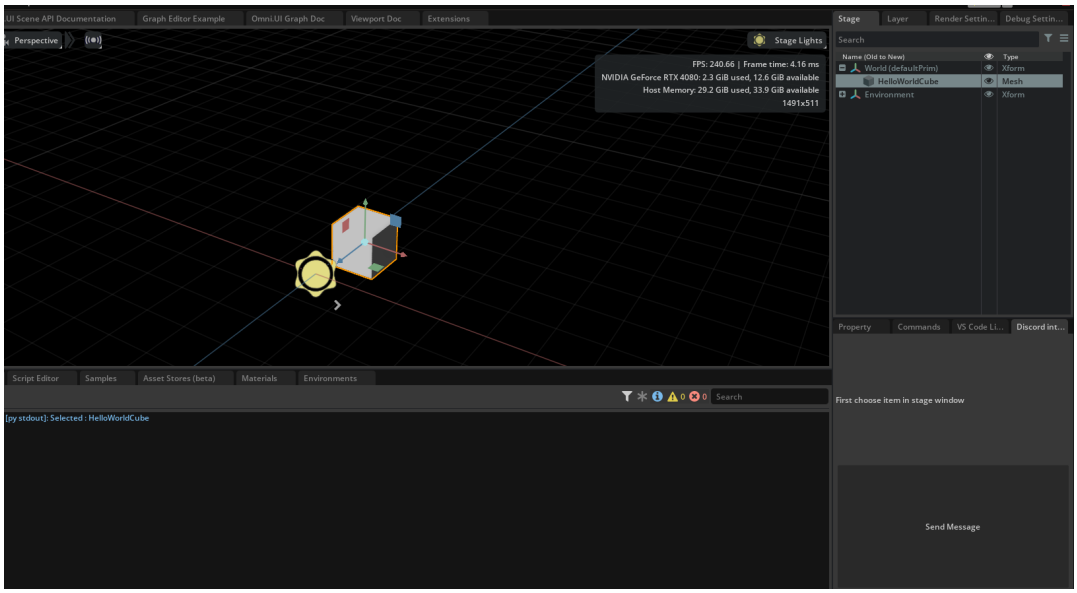

FIGURE 19. A screenshot from Omniverse Code

When Send Message button is pressed in Figure 19, Webhook is activated with content. This is shown in Figure 20 that is printing selected prim name as we defined it in line 47.

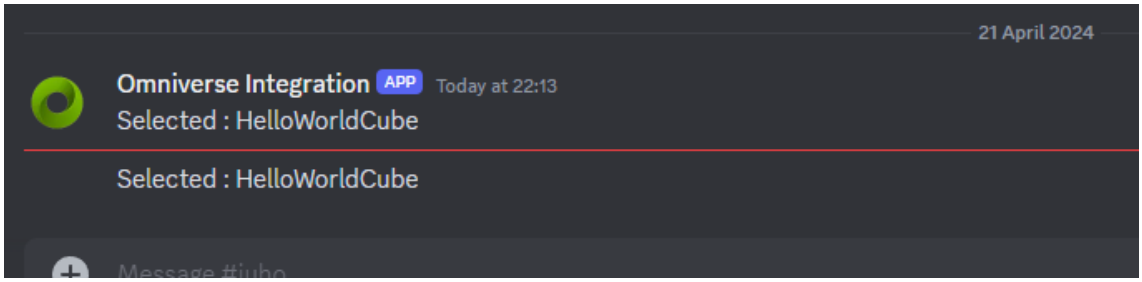

FIGURE 20. A screenshot from Discord

This expansion can be run on any Kit apps, for example Omniverse USD Presenter, Omniverse USD Composer and Omniverse Isaac Sim applications.

In a relatively small amount of source code, it is possible to integrate Discord Webhooks, Omniverse Ecosystem and USD scenario as they have well designed architecture and designed to be easily expandable.

6 CONCULSION

Further suggestions based on findings and discussions in this thesis, several further suggestions can be made to improve implementation and usability of USD and NVIDIA Omniverse in real-time collaborative environments. These suggestions address limitations and increase potentiality for future development and integration.

Main resource of USD data is currently other tools already integrated as discussed on chapter 3.2. Big limitations of these tools is that there has been multiple approaches for integration implementation, from virtual disk to omni-directional. Based on the experiences on these tools, only omni-directional connectors have shown full potential Omniverse, so uni-directional connectors should not be developed and invest full in omni-directional feature set. This allows much deeper collaboration. Another limitation in practice has been couple of closed-source libraries in NVIDIA Omniverse, example authentication to nucleus server is behind closed-source library. This has created some headache from developer side as reverse engineering of it has been prohibited by license.

Even with this limitation extension development could be said to be like other game engines, but instead of creating builds for game you are building a kit with your own extensions.

Another interesting possibility could be in Defense industry, as layering model would allow only with specific access to view details. Example, there could be a military base and contractors who are working on adding new ethernet cables around structure. These structures could see building BIM data in 3D view and how to install new cable flow are not able to view any details outside of area they are working on and details of unrelated, example details of electricity cables and AC are restricted and shown as blocks, so no unnecessary details are shared to contractors and allow better sensitive data control.

Also, defense, space and medical fields are using distributed interactive simulations (DIS) standard for interoperability of different simulations. Omniverse could create 3D visualization of these simulations as most simulations are data focused without 3D visuals. DIS data does not include model data on its payloads, so loading models from it would not be possible, but this standard is very robust and trusted by critical parties whose field has only small error of margin allowed.

As for future vision for USD and Omniverse, USD has already been a game changer on movie industry and only recently tipping toes on other industries, especially industrial and defense sectors are known to be followers. But during past 1 year working this technology, I have seen surprising amount of interest and follow up. In my personal view USD has a bright future ahead, but NVIDIA Omniverse is still looking for its place in the ecosystem. As in the end Omniverse is closed source version of USD with license cost limitations for teams over 2 people. It is easier to expand Omniverse than creating support for existing visualization engine. Development top Omniverse would be cheaper but running it for long term in big teams is going to be more costly as the license is charged per GPU. NVIDIA is in control of the license, and we can see new licensing models in the future. NVIDIA could adopt minimum revenue license model, similar to what Unity game engine has. This would allow smaller companies to adopt Omniverse and pay the cost once they are profitable.

REFERENCES

1. Thacker, Jim, 2013. Pixar's USD system: the new super-Alembic? Search date 24.03.2023. https://www.cgchannel.com/2013/10/pixars-usd-system-the-new-super-alembic/
2. Pixar Animation Studio, 2016. Pixar Animation Studios Open Sources Universal Scene Description. Search date 24.03.2023. https://openusd.org/release/press_opensource_release.html
3. Pixar Animation Studio, 2016. USD License. Search date 24.03.2023. https://github.com/PixarAnimationStudios/USD/blob/release/LICENSE.txt
4. Pixar Animation Studio, 2021. Universal Scene Descriptor. Search date 24.03.2023. https://openusd.org/release/intro.html
5. Farinha, Diego, 2023. Q&A with Remedy Entertainment: Adopting USD into the Game Development Pipeline. Search date 23.03.2023. https://developer.nvidia.com/blog/qa-with-remedy-entertainment-adopting-usd-into-the-game-development-pipeline/
6. Remedy Entertainment, 2023. Book of USD. Search date 23.03.2023. https://remedy-entertainment.github.io/USDBook/
7. Schechter, Sonia, 2020. Essential guide to 3D file formats. Search date 24.03.2023. https://www.marxentlabs.com/3d-file-formats/
8. Burton, Arti, 2016. Pixar Animation Studios Open Sources Universal Scene Description. Search date 24.03.2023 https://80.lv/articles/pixar-animation-studios-open-sources-universal-scene-description/
9. Northcott, Drew, 2016. Search date 27.03.2023. https://qr.ae/przLw7
10. Evan DeFilippis, Stephen Michael, Impink Madison, Singell Jeffrey T., Polzer Raffaella Sadun, 2020. Collaborating during coronavirus: The impact of covid-19 on the nature of work. Search date 27.03.2023 http://www.nber.org/papers/w27612
11. NVIDIA, 2019. Development workflows. https://docs.omniverse.nvidia.com/prod_digital-twins/prod_digital-twins/building-warehouse-digital-twins/dev-workflows.html
12. Discussion with employee of Eclipse, subsidiary of Accenture, March 2023.
13. Nnoli, Ike, 2023. NVIDIA Omniverse Accelerates Game Content Creation With Generative AI Services and Game Engine Connectors. Search date 27.03.2023. https://blogs.nvidia.com/blog/2023/03/21/omniverse-accelerates-game-dev/
14. Vocal informal discussion Remedy employees IGDA Helsinki meetup. 14.03.2023.

15. Pixar. USD Universal Scene Descriptor. Search date 27.03.2023. https://www.pixar.com/usd
16. Unity Technologies, 2018. USD Unity SDK. https://github.com/Unity-Technologies/usd-unity-sdk
17. Martin, Scott, 2023. BMW Group Starts Global Rollout of NVIDIA Omniverse. https://blogs.nvidia.com/blog/2023/03/21/bmw-group-nvidia-omniverse/
18. NVIDIA Press Release, 2023. NVIDIA and Microsoft to Bring the Industrial Metaverse and AI to Hundreds of Millions of Enterprise Users via Azure Cloud. https://nvidianews.nvidia.com/news/nvidia-and-microsoft-to-bring-the-industrial-metaverse-and-ai-to-hundreds-of-millions-of-enterprise-users-via-azure-cloud
19. NVIDIA, 2023. GTC 2023 Keynote with NVIDIA CEO Jensen Huang. https://youtu.be/DiGB5uAYKAg
20. Nordman, Teemu, 2023. https://www.linkedin.com/posts/teemu-nordman_connecting-nvidia-omniverse-to-microsoft-activity-7044046252665126912-Sevy
21. NVIDIA, Press Release, 2022. NVIDIA and Partners Build Out Universal Scene Description to Accelerate Industrial Metaverse and Next Wave of AI. https://nvidianews.nvidia.com/news/nvidia-and-partners-build-out-universal-scene-description-to-accelerate-industrial-metaverse-and-next-wave-of-ai
22. NVIDIA, 2020. Technical requirements. https://docs.omniverse.nvidia.com/plat_omniverse/common/technical-requirements.html
23. NVIDIA, 2023. Renaming of NVIDIA Omniverse Create and NVIDIA Omniverse View. Search Date 24.4.2024. https://forums.developer.nvidia.com/t/renaming-of-nvidia-omniverse-create-and-nvidia-omniverse-view/244817
24. NVIDIA, Omniverse licensing and pricing. Search date 28.03.2023. https://www.nvidia.com/en-us/omniverse/download/
25. Pixar Animation Studio, 2016. FAQ. https://openusd.org/release/usdfaq.html
26. NVIDIA, 2022. Kit extension template repository. Search date 24.4.2024. https://github.com/NVIDIA-Omniverse/kit-extension-template
27. NVIDIA, 2020. OmniLive. Search date 24.4.2024. https://docs.omniverse.nvidia.com/app_create/prod_extensions/ext_live.html#live-workflow
28. NVIDIA, Extensions (Basic). Search date 24.4.2024. https://docs.omniverse.nvidia.com/kit/docs/kit-manual/latest/guide/extensions_basic.html
29. Ricard3D, Omniverse Community Manager. Search date 24.4.2024. https://forums.developer.nvidia.com/t/issues-on-connecting-tablet-ar/229677/5

30. Pixar Animation Studios, Maximizing USD performance, 2021. Search date 24.4.2024. https://openusd.org/release/maxperf.html
31. Siemens, Discover what's new in Tecnomatix 2307 (August 2023). Search date 24.4.2024. https://blogs.sw.siemens.com/tecnomatix/discover-whats-new-in-tecnomatix-august-2023/
32. NVIDIA, 2021, Emendizabal, Omniverse Discord Server is LIVE!. Search date 24.4.2024. https://forums.developer.nvidia.com/t/omniverse-discord-server-is-live/178422
33. WEF, Insight Report, March 2024, Page 41. https://www3.weforum.org/docs/WEF_Navigating_the_Industrial_Metaverse_A_Blueprint_2024.pdf
34. NVIDIA, Getting Started with Extensions. Search date 24.4.2024. https://docs.omniverse.nvidia.com/kit/docs/kit-manual/latest/guide/extensions_basic.html
35. Apple, WWDC22, Explore USD tools and rendering. Search date 24.4.2024. https://developer.apple.com/videos/play/wwdc2022/10141/
36. Apple, Use 3D objects in Keynote on Apple Vision Pro. Search date 24.4.2024. https://support.apple.com/fi-fi/guide/apple-vision-pro/tanef42de60c/visionos
37. NVIDIA, NVIDIA ONNIVERSE LICENSE AGREEMENT. Search date 24.4.2024. https://docs.omniverse.nvidia.com/platform/latest/common/NVIDIA_Omniverse_License_Agreement.html
38. NVIDIA [online image]. Search Date 24.4.2024. https://docs.omniverse.nvidia.com/digital-twins/latest/_images/dt-dev_workflow.png
39. Alliance for OpenUSD, 2023. The Alliance for OpenUSD (AOUSD): Shaping the future of 3D technology. Search date 07.05.2024. https://aousd.org/blog/the-alliance-for-openusd-aousd-shaping-the-future-of-3d-technology/