

抜粋版

はじめてのOpenUSD学習・開発キット

～3Dを使ったアニメ、メタバースの基礎となるUSDを習得～

学習・開発編



スペクトラム・テクノロジー株式会社

<https://spectrum-tech.co.jp>

sales1@spectrum-tech.co.jp

開発編 目次

- OpenUSD学習・開発キット 全体像 [3](#)
- ハード・ソフトウェア概要
 - 1. ハード概要 [4](#)
 - 2. ソフト概要
 - ① ソフト一覧 [5](#)
 - ② プログラム一覧 [6](#)
- 1. OpenUSDとは [8](#)
- 2. OpenUSD学習 [12](#)
- 3. OpenUSD学習ビデオ
 - ① Nvidia openusd learning コース1-7 [36](#)
 - ⑧ omniverse Physics Extension - Kit104 (推奨) [53](#)
 - ⑩ その他: openusd by pixar [60](#)
 - ⑪ その他: omniverse by Nvidia [61](#)
- 4. OpenUSD コード例 [62](#)
- 5. Omniverse応用 [98](#)
 - A) Application [99](#)
 - B) Extension [100](#)
 - C) Service [105](#)
 - D) Connector [108](#)
 - E) Industrial Digital Twins (推奨) [109](#)
 - F) Embedded Web Viewer [123](#)
 - G) 住宅事例 [124](#)
 - H) ダビンチ事例 [125](#)
- 6. 生成AIを活用したOpenusd事例 [126](#)

抜粋版のため、ページは一致しません

OpenUSD学習・開発キット 全体像

ハードウェア

CPU



16GB RAM

GPU



+

USBメモリ



OS



windowsのみで記載

AI連携 (NIM)



USD系



プログラム言語

python



開発ツール



Visual Studio Code

お客様準備

弊社提供

1. ハード概要

①. 必要なハードウェア仕様

開発キットと必要なハードウェアの概要です。

品名		仕様	備考
USBメモリ (windows版)		USB3.0 64GB Openusd, omniverseプログラム一式	
必要なハードウェア: お客様準備			
PC本体	cpu	Intel i7 Gen 5 AMD Ryzen	アニメなどは、高スペックのマシンが必要
	メモリ	16GB以上	
	Disk	250GB以上	
	GPU	RTX3070以上	

2. ソフト概要

①ソフトウェア一覧

本PCにインストールするソフトウェアの概要です。

区分	ソフト名	バージョン	備考
お客様準備: 弊社インストール・サービスにより提供可能			
OS	windows	10/11	
	ubuntu	20.04/22.04 LTS	手順書には記述していません
GPU driver		537.70	rtx
USBメモリにより提供: 手順書に基づきお客様でインストールして頂きます			
開発ツール	Visual studio	2022	msvc要
	Visual studio code	1.94	
	cmake	V3.24	
プログラム言語	python3	3.8以上	
	C++	V14.3	
openusd	openusd	V24.08	
omniverse	Usd composer他	2023.2.5	Usd view,kitなど多数
	kit-app-template Nvidia nim マイクロサービス		Extension, service, connectorなど多数

2. ソフトウェア概要

② 開発キットプログラム一覧 その1

A: 上級者
B: 普通
C: 入門
X: エラー

id	区分	項目	概要	難易度	コメント
1	Openusd学習	Hello world	Openusdの入口。Helloworldで体験	C	
2		Hello world redux	汎用primsを使ったHelloworldで体験	C	
3		プロパティの検査と作成	Helloworldで生成したプロパティの検査と変更を行います	B	
4		レイア参照	二つのレイヤ: refSphere、refSphere2を生成、色違いで表示	B	
5		レイア様式変換	usda(text), usdc(binary), usd間の変換	B	
6		ステージ横断	UsdStage 上の合成プリムを横断するためのAPIについて説明します	B	
7		Authoring Variants	Primの色(青、赤、緑)を変数化し、選択できるようにする	B	
8		アニメーション例	駒の回転アニメ例	A	
9		ロゴ表示例	ロゴ表示例	B	
10		End to End Example	球とテーブルのアニメ例	A	
11		事例: kitchen set	キッチンの例	A	
1	OpenUSD学習ビデオ	コース1: Learning About Stages, Prims and Attributes	Stages, Prims and Attributesの学習	B	ビデオとJupyter notebook併用
2		コース2: Working With Prims and Default Schemas	Prims and Default Schemasの動作	B	
3		コース3: Using Attributes:属性	材質、重量などの属性を追加	B	
4		コース4: Traversing Stages:移動	ステージのシーングラフを移動	B	
5		コース5: Understanding Model Kinds:モデル種類	assembly, group, componentのモデル種類の学習	B	
6		コース6: Setting Up Basic Animations:アニメの基本	TimeCodeの学習	C	
7		コース7: An Introduction to Strength Ordering:強調	レイヤ、強調、参照について学習	B	
8		omniverse Physics Extension - Kit104	アニメ: 物体落下など、使い方を細かく説明。Step1-23まで段階的に説明	B	非常に有益
9		その他	OpenUSD と Omniverse を使用した 3D 製品コンフィギュレーターの構築	B	
10		その他: opensudビデオ	Openusdの立ち上げのpixar作成ビデオ	B	
11		その他: omniverseビデオ	Nvidia作成のomniverseビデオ	B	

2. ソフトウェア概要

② 開発キットプログラム一覧 その2

A: 上級者
B: 普通
C: 入門
X: エラー

id	区分	項目	概要	難易度	コメント
0	OpenUSD コード例	USD explore使い方	USD exploreの画面操作方法	C	
1		Specialize:特化	特徴を加える	B	Pythonを使った事例
2		Visibility: 可視化	primsを表示または非表示にすることができません。	C	
3		References & Payloads:参照と荷物	Payloadとreferenceは同じように参照。	B, X	一部表示がでない。
4		Prims	primの状態を確認。基本構成要素。	B	
5		Paths: パス	パスを使ってprimsを連結	B	
6		Variant Sets:変数設定	変数を作成し、選択	B	
7		Cameras:カメラ	カメラの遠近設定など	B	
8		Transforms:変換	Primsのローカル、ワールドの変換	B	
9		Stage:ステージ	Stageの単位、軸などの設定	B	
10		Materials:材質	MDL、表面の材質を設定	B	
11		Inherits:継承	継承を追加	B	
12		Layers:レイア	サブレイヤの追加	B	
13		Hierarchy & Traversal:階層と横断	階層になっているprimの名前、タイプ、子を検索	B	
14		Data Types:データ型	データをnumpyなどの配列に変換	B	
15		Properties:プロパティ	関係、既存のプロパティの検証、属性設定	B	定義が必要と思われる
1	Omniverse応用	Application	Omniverse Kit SDKを使用してカスタムアプリケーションを作成	B	VS codeで操作
2		Extensions	拡張機能として、simple scene, api connectを作成	A	
3		service	Web系のサービス生成	B	使わなくてもよい。
4		connector	作成したusdの検証	B	Python3.8以上が要
5		Industrial Digital Twins	工場内にベルトコンベヤ、ボックスを配置し、動作させる事例。細かく解説	B	学習必須
6		Embedded Web Viewer	Webクライアントからusd操作	B	紹介のみ
7		住宅事例	住宅内のロボットの事例	B	複雑
8		ダビンチ事例	モニリザの絵画の展示例	B	67GBでgpu:10GB以上のメモリが必要。重い
1	生成AIを活用したOpenusd事例	USD検索	NVIDIA NIM マイクロサービスとしてクラウドで提供。簡単にusd事例検索	C	
2		USD検証	作成したusdの検証。Jsonで条件設定が必要	B	初期はエラー
3		USDcode	Chatでusdプログラムを生成。	A	プロンプトに工夫が必要

1. OpenUSDとは

①概要

- Pixar によって開発された Universal Scene Description (OpenUSD) は、高度なコラボレーションワークフローを促進しながら、さまざまなアセット、ソース、アニメーションで構成される 3D シーンを堅牢かつスケーラブルに交換できる初のオープンソースソフトウェアです。
- 2023年pixar,nvidia, apple,autodeskなどによりopenusdアライアンスが設立
- <https://www.pixar.com/openusd>
- <https://openusd.org/release/index.html>
- 特徴
 - 共通言語化
 - 3D データの定義、パッケージ化、組み立て、編集のための豊富な共通言語を提供し、複数のデジタルコンテンツ作成アプリケーションの使用を容易にします。
 - 共同作業
 - 複数のアーティストが同じアセットやシーンで共同作業できるようにする。
 - 反復利用
 - レイテンシを最小限に抑えて芸術的な反復を最大化。

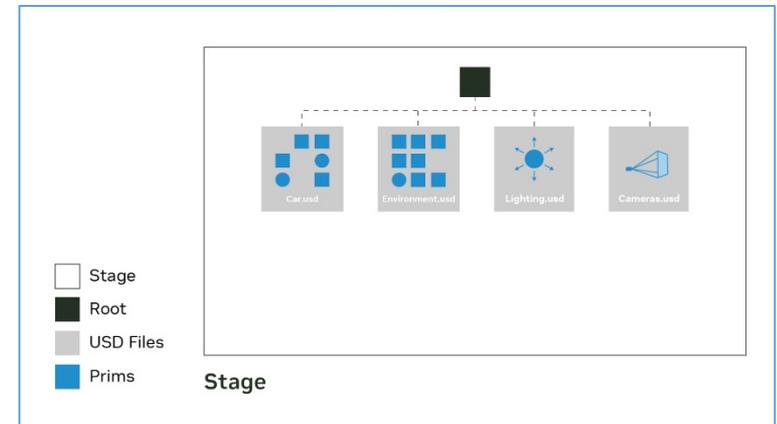
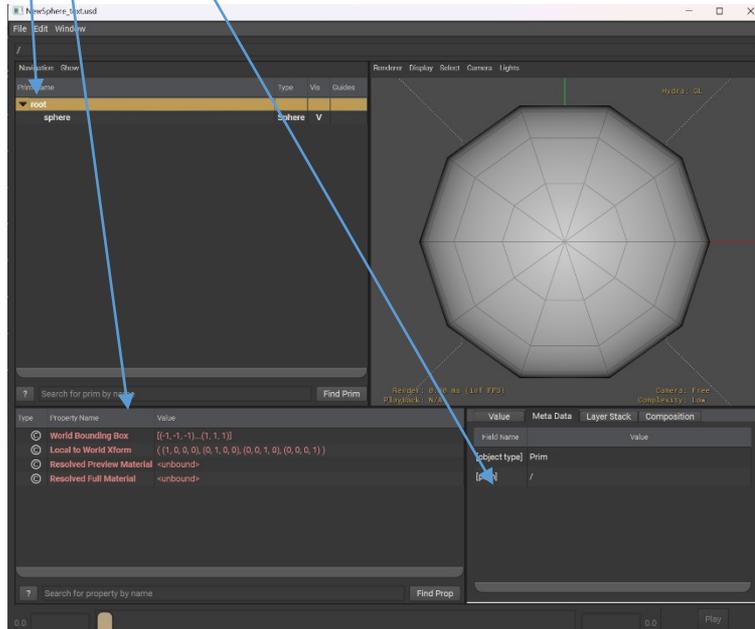
1. OpenUSDとは

②構成

• <https://openusd.org/release/glossary.html>

• Usdを構成する代表的な項目を紹介。

- Prim:primitiveの略。Prim は USD の主要なコンテナオブジェクトです。Prim は他の Prim を含む(および順序付ける)ことができ、ステージ上に「名前空間階層」を作成します。また、Prim は意味のあるデータを保持するプロパティを含む(および順序付ける)こともできます。
- Path:パスは名前空間内の場所です。
- Property: プロパティは USD の別の種類のネームスペースオブジェクトです(最初のもはプリムです)。プリムは合成されたシーンの構成とインデックスを提供しますが、プロパティには「実際のデータ」が含まれます。プロパティには、属性とリレーションシップの2つのタイプがあります。
- Layer:レイヤーは、USD のシーン記述のアトミックな永続コンテナです。レイヤーには0個以上の PrimSpec が含まれており、これらはプロパティとメタデータの値を記述します。
- Stage:ステージは、ルート USD ファイルから派生したシーングラフと、それが構成するすべての参照/レイヤーファイルに対する USD 抽象化です。
- Schema:USD では、スキーマを、何らかの UsdObject から構造化データを作成および取得することを目的とするオブジェクトとして定義しています。コアにあるほとんどのスキーマは「プリムスキーマ」であり、これはさらに IsA スキーマと API スキーマに洗練されており、USD ディストリビューションでは、独自のスキーマを作成するためのコード生成ツールを提供しています。



stageの構成

1. OpenUSDとは

②構成

- <https://openusd.org/release/glossary.html>
- Prim:primitiveの略。Prim は USD の主要なコンテナ オブジェクトです。Prim は他の Prim を含む (および順序付ける) ことができ、ステージ上に「名前空間階層」を作成します。また、Prim は意味のあるデータを保持するプロパティを含む (および順序付ける) こともできます。
 - Primは、プロパティ、属性、メタデータ、関係などの情報を有する
 - また画像、非画像の情報も有している



ST Spectrum Technology

2. OpenUSD 学習

OpenUSDの基礎を学習します。

https://openusd.org/release/tut_helloworld.html#

① Hello world

• usda生成

- X64 native tool 又は visual studio の terminal 使用

➢ cd

C:\Users\skyri\Desktop\masa_legion\omniverse_soft\OpenUSD\extras\usd\tutorials\helloWorld

➢ python helloWorld.py

- HelloWorld.usda が生成

• Usda表示

➢ usdview HelloWorld.usda

- 又は omniverse > ライブラリ > usdview でも表示可能

コマンド入力

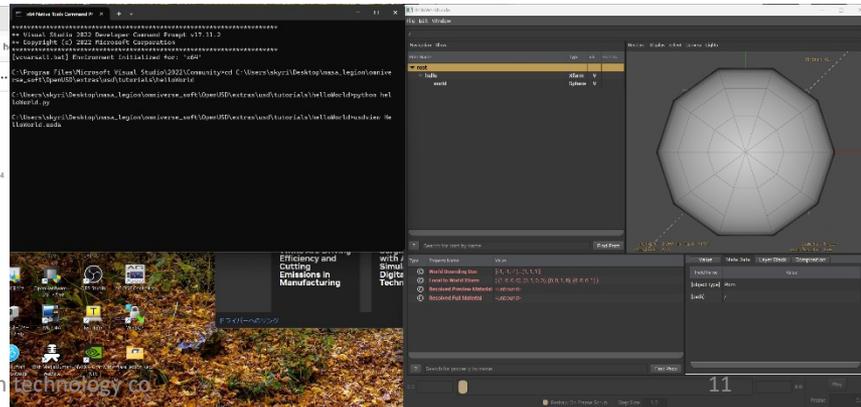
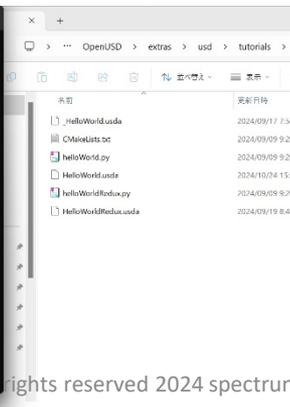
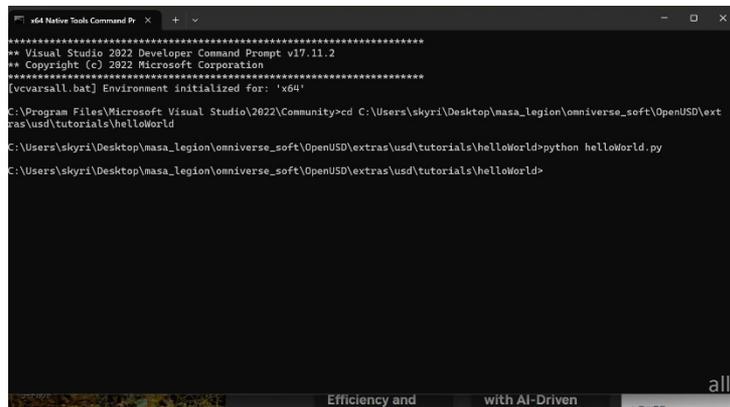
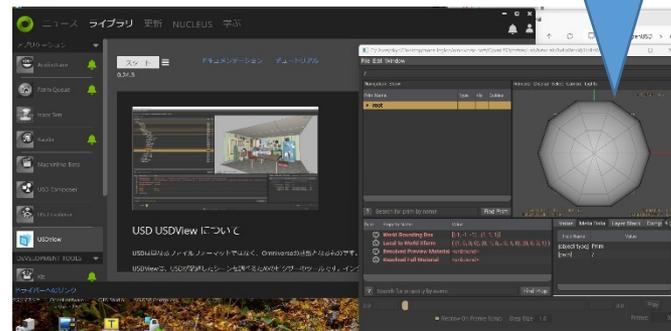
> cd

C:\Users\skyri\Desktop\masa_legion\omniverse_soft\OpenUSD\extras\usd\tutorials\helloWorld

> python helloWorld.py

> usdview HelloWorld.usda

円が表示されない場合は、
display>shading
mode>wireframe:defaultを
wireframeonsurfaceで表示



2. OpenUSD 学習

OpenUSDの基礎を学習します。

https://openusd.org/release/tutorial_helloWorld.html

① Hello world

• usda生成

- X64 native tool又はvisual studioのterminal使用

➢ cd

```
C:\Users\skyri\Desktop\masa_legion\omniverse_soft\OpenUSD\extras\usd\tutorials\helloWorld
```

➢ python helloWorld.py

- HelloWorld.usdaが生成

• Usda表示

➢ usdview HelloWorld.usda

- 又はomniverse>ライブラリ>usdviewでも表示可能

➢ usdcat HelloWorld.usda

➢ usdedit HelloWorld.usda

端末に表示
txtファイルに表示

コマンド入力

> cd

```
C:\Users\skyri\Desktop\masa_legion\omniverse_soft\OpenUSD\extras\usd\tutorials\helloWorld
```

> python helloWorld.py

> usdview HelloWorld.usda

> usdcat HelloWorld.usda

> usdedit HelloWorld.usda

情報は少ないですが、defaultの値があるので、図が表示されます。

```
x64 Native Tools Command Pr
** Visual Studio 2022 Developer Command Prompt v17.11.2
** Copyright (c) 2022 Microsoft Corporation
[vcvarsall.bat] Environment initialized for: 'x64'
C:\Program Files\Microsoft Visual Studio\2022\Community>cd C:\Users\skyri\Desktop\masa_legion\omniverse_soft\OpenUSD\extras\usd\tutorials\helloWorld
C:\Users\skyri\Desktop\masa_legion\omniverse_soft\OpenUSD\extras\usd\tutorials\helloWorld>python helloWorld.py
C:\Users\skyri\Desktop\masa_legion\omniverse_soft\OpenUSD\extras\usd\tutorials\helloWorld>usdview HelloWorld.usda
C:\Users\skyri\Desktop\masa_legion\omniverse_soft\OpenUSD\extras\usd\tutorials\helloWorld>usdcat HelloWorld.usda
#usda 1.0
def Xform "hello"
{
  def Sphere "world"
  {
  }
}
```

```
x64 Native Tools Command Pr
** Visual Studio 2022 Developer Command Prompt v17.11.2
** Copyright (c) 2022 Microsoft Corporation
[vcvarsall.bat] Environment initialized for: 'x64'
C:\Program Files\Microsoft Visual Studio\2022\Community>cd C:\Users\skyri\Desktop\masa_legion\omniverse_soft\OpenUSD\extras\usd\tutorials\helloWorld
C:\Users\skyri\Desktop\masa_legion\omniverse_soft\OpenUSD\extras\usd\tutorials\helloWorld>python helloWorld.py
C:\Users\skyri\Desktop\masa_legion\omniverse_soft\OpenUSD\extras\usd\tutorials\helloWorld>usdview HelloWorld.usda
C:\Users\skyri\Desktop\masa_legion\omniverse_soft\OpenUSD\extras\usd\tutorials\helloWorld>usdcat HelloWorld.usda
#usda 1.0
def Xform "hello"
{
  def Sphere "world"
  {
  }
}
```

2. OpenUSD 学習

Openusdの基礎を学習します。

https://openusd.org/release/tut_helloWorld.html#

① Hello world

• usda生成

- X64 native tool又はvisual studioのterminal使用

➢ cd C:\Users\skyri\Desktop\masa_legion\omniverse_soft\OpenUSD\extras\usd\tutorials\helloWorld

➢ python helloWorld.py

- HelloWorld.usdaが生成

• Usda api操作

➢ usdview HelloWorld.usda

- Python interpreterで操作

- Windows>interpreter:右側へ

➢ usdviewApi.prim

- Usd.Prim(</hello/world>)

➢ usdviewApi.prim.GetTypeName()

- 'Sphere'

➢ usdviewApi.prim.GetAttribute('radius').Get()

- 1.0

コマンド入力

> cd

C:\Users\skyri\Desktop\masa_legion\omniverse_soft\OpenUSD\extras\usd\tutorials\helloWorld

> python helloWorld.py

> usdview HelloWorld.usda

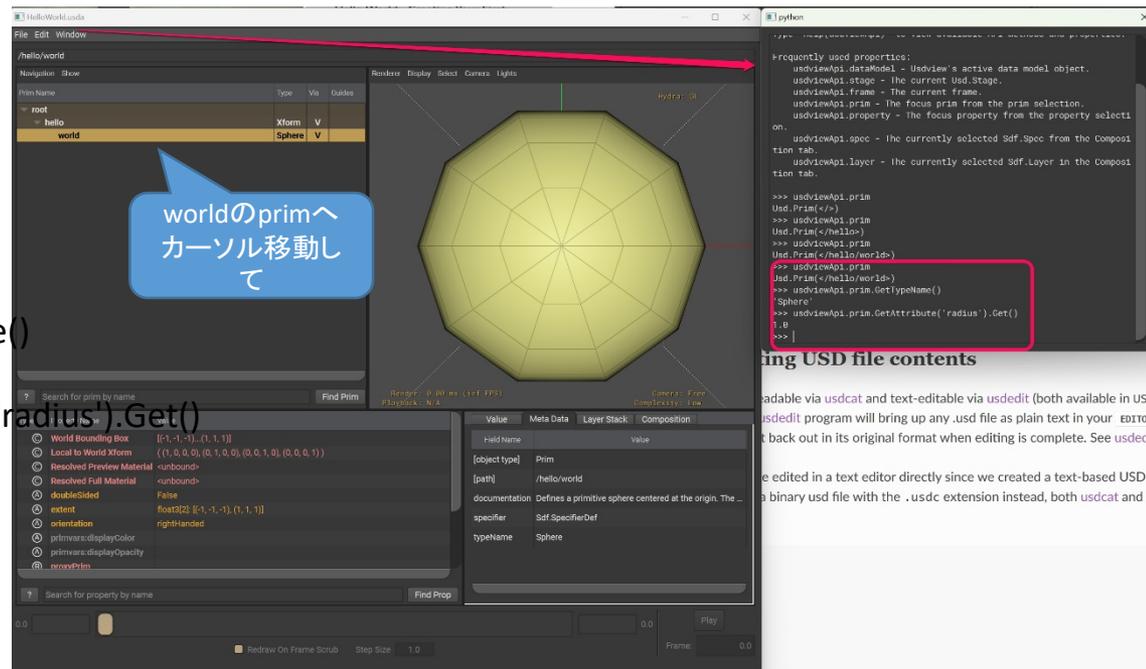
> usdcat HelloWorld.usda

> usdedit HelloWorld.usda

> usdviewApi.prim

> usdviewApi.prim.GetTypeName()

> usdviewApi.prim.GetAttribute('radius').Get()



2. OpenUSD 学習

OpenUSDの基礎を学習します。

https://openusd.org/release/tut_converting_between_layer_formats.html

コマンド入力

```
> cd
```

```
C:\Users\skyri\Desktop\masa_legion\omniverse_soft\OpenUSD\extras\usd\tutorials\convertingLayerFormats
```

```
> usdcat -o NewSphere.usdc Sphere.usda
```

```
> usddiff Sphere.usda NewSphere.usdc
```

```
> usdcat -o NewSphere_binary.usd --usdFormat usdc Sphere.usd
```

```
> usdview Sphere.usd
```

⑤ レイア様式変換

- usda(text), usdc(binary), usd間の変換。基本は、usdのtextファイル

- Usda生成

- X64 native tool又はvisual studioのterminal使用

- cd

- C:\Users\skyri\Desktop\masa_legion\omniverse_soft\OpenUSD\extras\usd\tutorials\convertingLayerFormats

- usdcat -o NewSphere.usdc Sphere.usda ファイル変換:usdaからusdcへ

- usddiff Sphere.usda NewSphere.usdc 差分比較

- usdcat -o NewSphere_binary.usd --usdFormat usdc Sphere.usd usdからusdcフォーマット指定で変換(同じusdファイルで中身のフォーマットを変更)

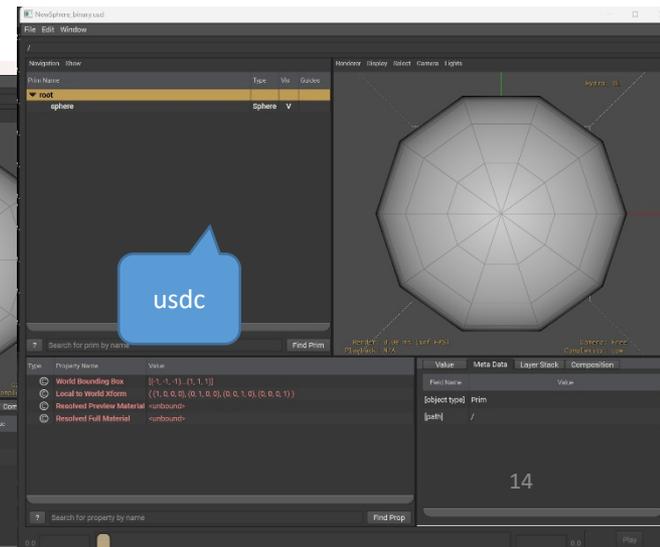
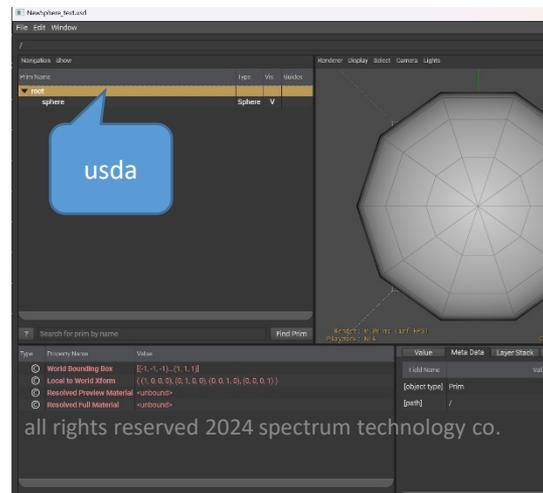
- Usda表示

- usdview Sphere.usd

- usdview Sphere.usda

- usdview NewSphere.usdc

Usdviewで表示させると同じになります



2. OpenUSD 学習

Openusdの基礎を学習します。

https://openusd.org/release/tut_xforms.html

⑧ アニメーション例

- 駒の回転アニメ例。

- Usda生成

- X64 native tool又はvisual studioのterminal使用

➢ cd

C:\Users\skyri\Desktop\masa_legion\omniverse_soft\OpenUSD\extras\usd\tutorials\animatedTop

➢ python generate_examples.py

- Step1からsetp6.usdaが生成

- Usda表示

➢ usdview top.geom.usd 元のコマ

➢ usdview Step1.usda 下のplayでアニメ、何も表示されな

➢ usdview Step4.usda playでコマ回転

➢ usdview Step4A.usda 軸が揺れて回転、spin,tilt

➢ usdview Step5.usda 軸がずれながら揺れて回転

➢ usdview Step6.usda コマ3個で回転

コマンド入力

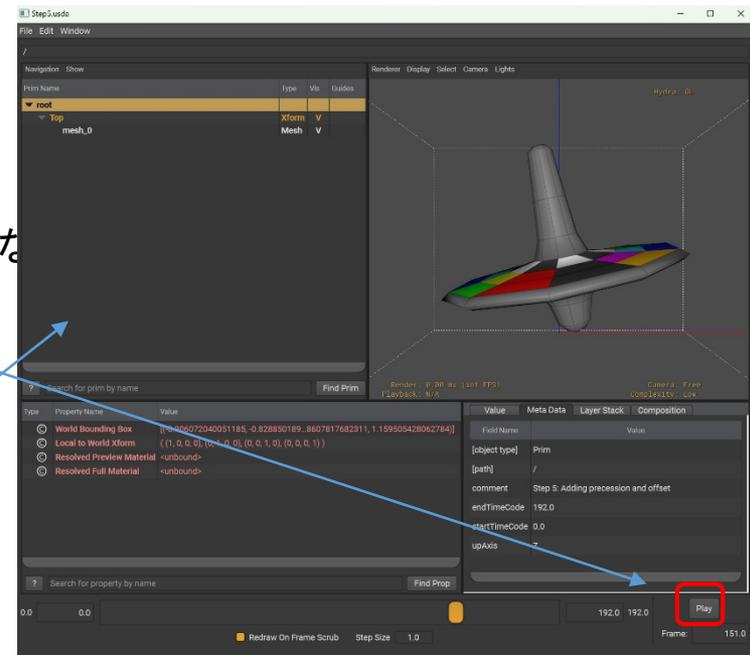
> cd

C:\Users\skyri\Desktop\masa_legion\omniverse_soft\OpenUSD\extras\usd\tutorials\animatedTop

> python generate_examples.py

> usdview top.geom.usd

> usdview Step1.usda



2. OpenUSD 学習

OpenUSDの基礎を学習します。

https://openusd.org/release/tut_end_to_end.html

⑩ End to End Example

- 球とテーブルのアニメ例。
- Usda生成
 - X64 native tool又はvisual studioのterminal使用
 - cd
C:\Users\skyri\Desktop\masa_legion\omniverse_soft\OpenUSD\extras\usd\tutorials\endToE
nd

● Simple shot work: 玉突きのアニメ

- python tutorial_scripts/add_set_to_s00.py
- usdview shots/s00/s00.usd

- python tutorial_scripts/layout_shot_s00_01.py
- python tutorial_scripts/anim_shot_s00_01.py
- usdview shots/s00_01/s00_01.usd
- playボタンでアニメ動作

コマンド入力

> cd

C:\Users\skyri\Desktop\masa_legion\omniverse_soft\OpenUSD\extras\usd\tutorials\endToE

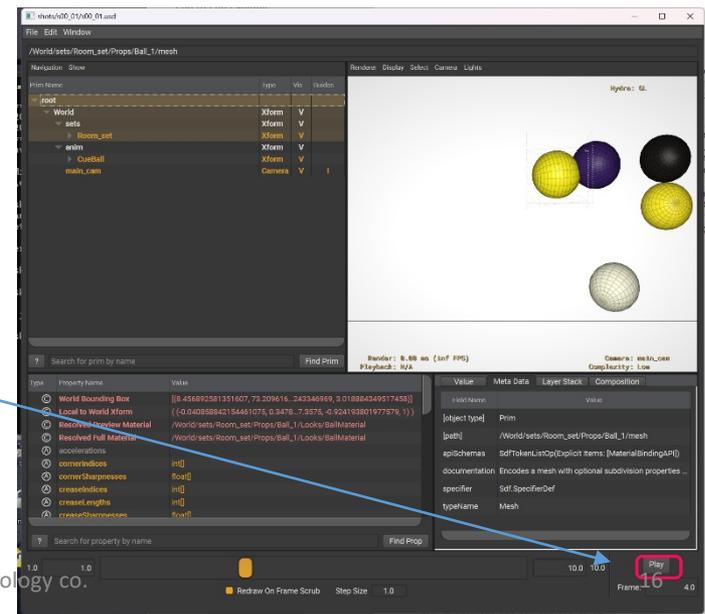
> python tutorial_scripts/add_set_to_s00.py

> usdview shots/s00/s00.usd

> python tutorial_scripts/layout_shot_s00_01.py

> python tutorial_scripts/anim_shot_s00_01.py

> usdview shots/s00_01/s00_01.usd



2. OpenUSD 学習

OpenUSDの基礎を学習します。

https://openusd.org/release/usd_products.html

⑫ 製品例

- Adobe, Autodesk, apple, nvidiaなどのusdを使った製品一覧
- Nvidia omniverseは個別に説明。

Products Using USD

USD has support in many 3D Content Creation Applications and Ecosystems.

This list is maintained by the community, and is not meant to be an exhaustive or complete list of resources, nor is it an endorsement of any of the linked material. We haven't checked the linked materials for accuracy, nor to see if they represent best practices. Please enjoy this material in the spirit intended, of celebrating community and industry achievements around the use and advancement of USD.

- 3Delight
- Adobe
 - Substance 3D Painter
 - Substance 3D Modeler
 - Substance 3D Stager
- AMD ProRender
- Apple
 - Preview and QuickLook
 - AR Creation Tools
 - Rendering Engines
 - ModellIO
 - Motion
- ArcGIS CityEngine
- Autodesk
 - 3ds Max
 - Arnold
 - Bifrost
 - Fusion 360
 - Maya
- Revit
- Blender Foundation
 - Blender
 - Cycles
- Chaos V-Ray
- E-on
 - Plant Factory
 - Vue
- Gaffer
- Golaem
- Guerilla Render
- Foundry
 - Katana
 - Mari
 - Modo
 - Nuke
- Intel OSPRay
- Dreamworks Moonray
- J Cube
- Multiverse for Maya
- Muse
- Left Angle Autograph
- Maxon
 - Cinema 4D
 - Redshift
 - ZBrush
- NVIDIA Omniverse
- Procreate
- Shapr3D
- SideFX Houdini
- SynthEyes
- Tilt Brush
- Unity
- Unreal Engine
- Usdtweak
- Vicon Shogun
- Wizard

3. OpenUSD 学習ビデオ

OpenUSDの基礎をビデオとjupyter notebookで学習します。

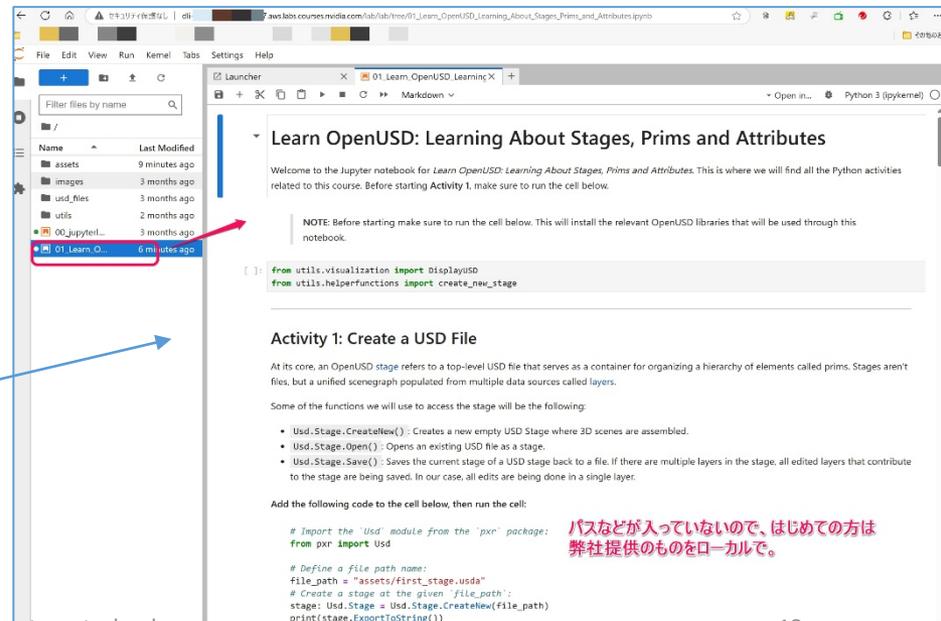
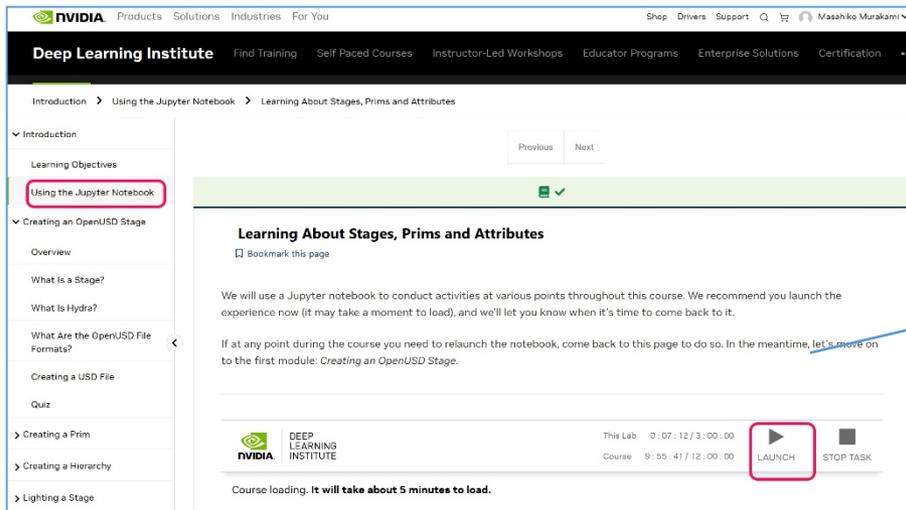
nvidia作成したもので基礎をわかりやすく解説

<https://www.nvidia.com/en-us/learn/learning-path/openusd/>

① コース1: Learning About Stages, Prims and Attributes

- Nvidiaへの登録が必要
- nvidia提供のjupyter notebook起動。

01_Learn_OpenUSD_Learning_About_Stages_Prims_and_Attributesをクリックで起動



3. OpenUSD 学習ビデオ

OpenUSDの基礎をビデオとjupyter notebookで学習します。

nvidia作成したもので基礎をわかりやすく解説

<https://www.nvidia.com/en-us/learn/learning-path/openusd/>

① コース1: Learning About Stages, Prims and Attributes

• Creating an OpenUSD Stage

- What Is a Stage? データを1か所に集約し、さまざまなアプリケーションやデータセット間でのコラボレーションが可能になります。
- What Is Hydra? 複雑な 3D シーンを効率的かつ柔軟にレンダリングできるようになります。
- What Are the OpenUSD File Formats?
 - udda(text), usdc(3d scene), usdz(zip), usd
- Creating a USD File
 - 上記のjupyter notebookで学習

各自でビデオで
学習、最終回に
テストあり

The screenshot shows the NVIDIA Deep Learning Institute website. The main content area displays a course titled "Creating an OpenUSD Stage" with a table of contents on the left. The table of contents includes sections like "Introduction", "Learning Objectives", "Using the Jupyter Notebook", "Creating an OpenUSD Stage", "Overview", "What is a Stage?", "What is Hydra?", "What Are the OpenUSD File Formats?", "Creating a USD File", "Creating a Prim", "Creating a Hierarchy", "Lighting a Stage", and "Defining Attributes". The "What is a Stage?" section is currently selected. On the right, there is a diagram of a USD file structure. The diagram shows a "Root" node (black square) connected to three "USD Files" nodes (grey squares): "Geometry.usd", "Materials.usd", and "Lights.usd". A legend below the diagram identifies the symbols: a black square for "Stage", a grey square for "Root", a grey square for "USD Files", and a blue square for "Prims". The diagram is labeled "Scenegraph".

3. OpenUSD 学習ビデオ

OpenUSDの基礎をビデオとjupyter notebookで学習します。

nvidia作成したもので基礎をわかりやすく解説

<https://www.nvidia.com/en-us/learn/learning-path/openusd/>

① コース1: Learning About Stages, Prims and Attributes

- Creating an OpenUSD Stage

- Creating a Prim

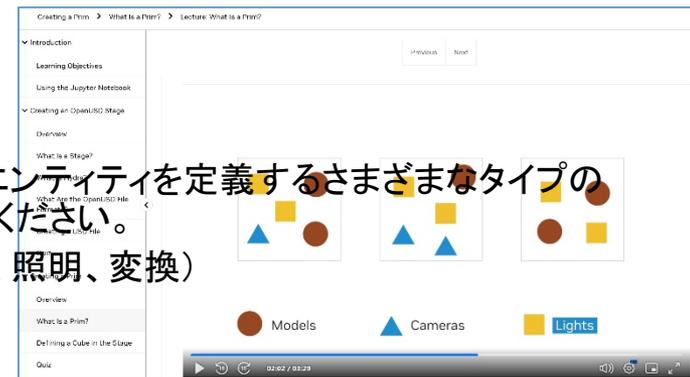
- What Is a Prim?

- Primitiveの略、プリムは、シーン内のオブジェクトまたはエンティティを定義するさまざまなタイプのデータ、属性、および関係を保持するコンテナと考えてください。

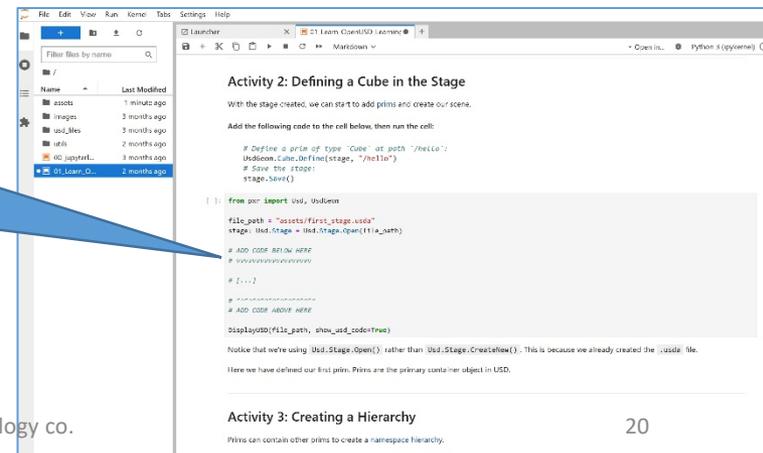
- usdフレームワークのコア・コンポーネント(メッシュ、材料、照明、変換)

- Defining a Cube in the Stage

- Ipythonで操作;activity2 cube,helloのusd作成



Jupyter notebookの四角内にコードをコピーすると動作します。指示どおりに行ってください。



3. OpenUSD 学習ビデオ

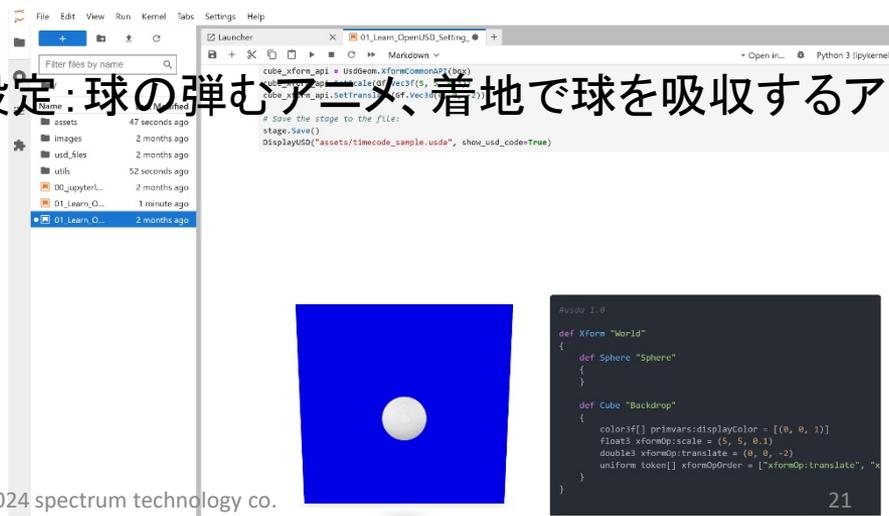
OpenUSDの基礎をビデオとjupyter notebookで学習します。

nvidia作成したもので基礎をわかりやすく解説

<https://www.nvidia.com/en-us/learn/learning-path/openusd/>

⑥ コース6: Setting Up Basic Animations: アニメの基本

- Experimenting With TimeCode and TimeSamples
- What Are TimeCode and TimeSamples?
 - TimeCode は、単位が割り当てられていない時間点です。これらは、ステージから単位が派生したフレームと考えることができます。TimeSample は、USD の属性に関連付けられた個々の時間変動値を指します。各属性には、timeCode を属性のデータ型値にマップする timeSample のコレクションを含めることができ、時間の経過に伴うアニメーションが可能になります。
 - usdシーンのアニメとシミュレーションを行う重要な要素
- ipynb act1 timecode設定1-60
- ipynb act2 timecode設定と値設定: 球の弾むアニメ、着地で球を吸収するアニメ



4. OpenUSD コード例

OpenUSDのcode sampleで基礎を学習します。

https://docs.omniverse.nvidia.com/dev-guide/latest/programmer_ref/usd.html#

② Visibility: 可視化

- https://docs.omniverse.nvidia.com/dev-guide/latest/programmer_ref/usd/visibility/show-hide-prim.html
- 可視性属性を継承または非表示に設定することで、プリムを表示または非表示にすることができます。

➤ cd C:\Users\skyri\Desktop\masa_legion\omniverse_soft\OpenUSD-Code-Samples\sample

➤ python visible.py

➤ usdview visible.usda

cubeが表示、visをiにすると非表示又は右クリックで

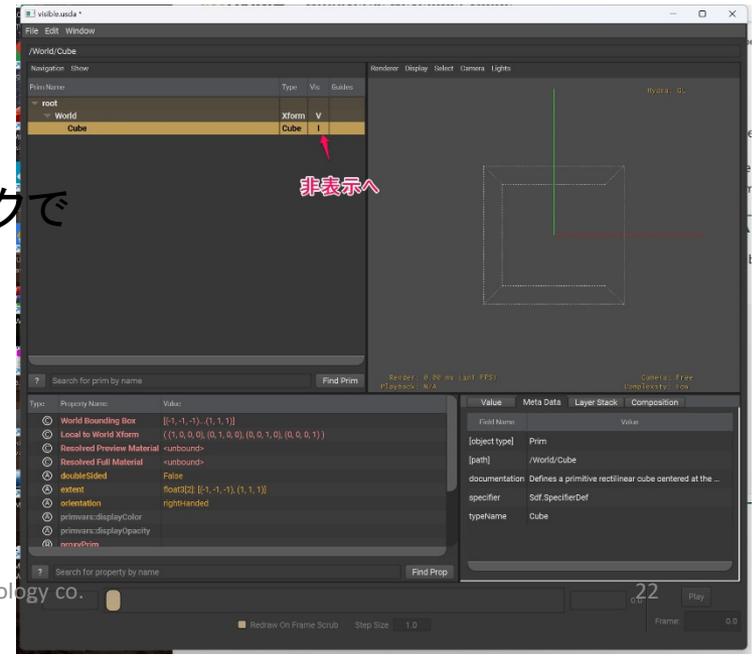
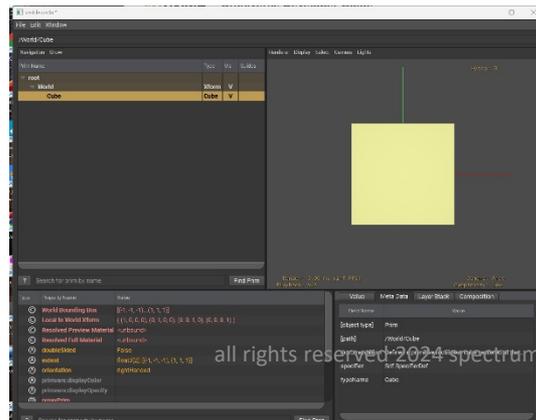
コマンド入力

> cd

C:\Users\skyri\Desktop\masa_legion\omniverse_soft\OpenUSD-Code-Samples\sample

> python visible.py

> usdview visible.usda



4. OpenUSD コード例

OpenUSDのcode sampleで基礎を学習します。

https://docs.omniverse.nvidia.com/dev-guide/latest/programmer_ref/usd.html#

コマンド入力

> cd

C:\Users\skyri\Desktop\masa_legion\omniverse_soft\OpenUSD-Code-Samples\sample

> python prim_parent_path.py

> usdview prim_parent_path.usda

⑤ Paths

C) Get the Parent Path for a Prim Path: 親のパス取得

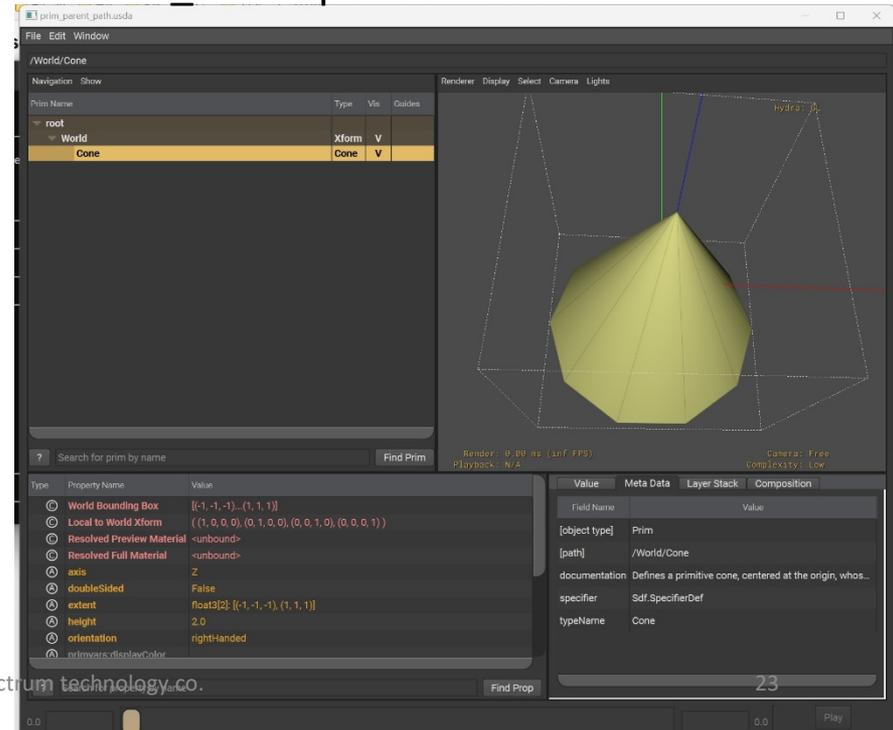
- https://docs.omniverse.nvidia.com/dev-guide/latest/programmer_ref/usd/paths/get-parent-path.html

- プリムパスが必要で、その親のプリムパスを取得したい場合は、Sdf.Path.GetParentPath() を使用できます。

- cd C:\Users\skyri\Desktop\masa_legion\omniverse_soft\OpenUSD-Code-Samples\sample

- python prim_parent_path.py cone生成

- usdview prim_parent_path.usda



4. OpenUSD コード例

OpenUSDのcode sampleで基礎を学習します。

コマンド入力

> cd

C:\Users\skyri\Desktop\masa_legion\omniverse_soft\OpenUSD-Code-Samples\sample

> python transf_loc.py

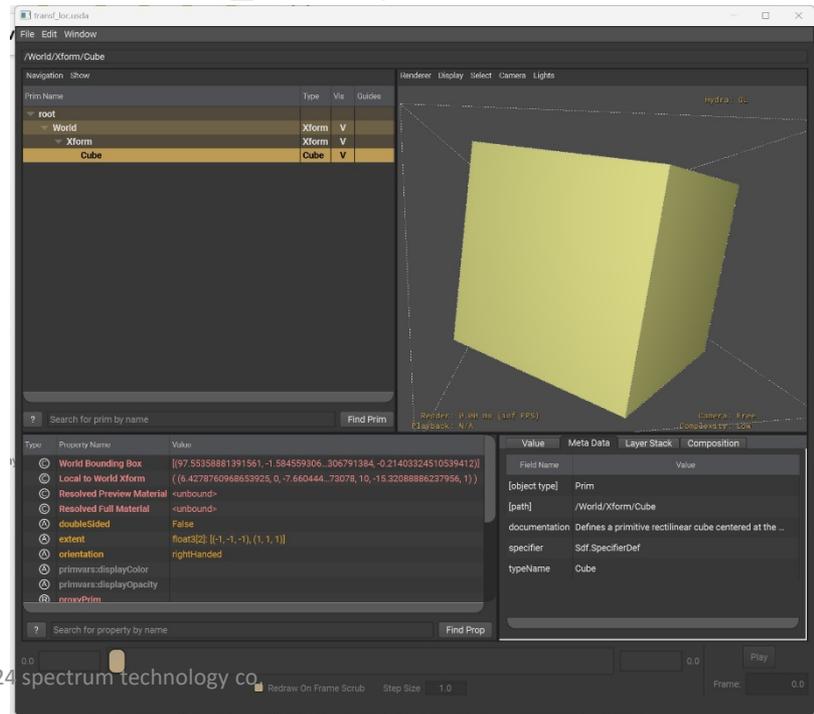
> usdview transf_loc.usda

https://docs.omniverse.nvidia.com/dev-guide/latest/programmer_ref/usd.html#

⑧ Transforms:変換

A) Get the Local Space Transforms for a Prim: ローカル変換

- https://docs.omniverse.nvidia.com/dev-guide/latest/programmer_ref/usd/transforms/get-local-transforms.html
- プリムのローカル変換を取得する必要がある場合 (つまり、祖先プリムの変換を考慮しない場合)、キットの便利な関数 `omni.usd.get_local_transform_SRT()` を使用できます。
- cd C:\Users\skyri\Desktop\masa_legion\omniverse_soft\OpenUSD-Code-Samples\sample
- python transf_loc.py
- usdview transf_loc.usda



4. OpenUSD コード例

Openusdのcode sampleで基礎を学習します。

https://docs.omniverse.nvidia.com/dev-guide/latest/programmer_ref/usd.html#

⑧ Transforms:変換

B) Get the World Space Transforms for a Prim: ワールド変換

- https://docs.omniverse.nvidia.com/dev-guide/latest/programmer_ref/usd/transforms/get-world-transforms.html
- ワールド空間でのプリムの変換を取得する必要がある場合 (つまり、祖先プリムの変換を考慮に入れる場合)、キットの便利な関数 `omni.usd.get_world_transform_matrix()` を使用できます。

➤ `cd C:\Users\skyri\Desktop\masa_legion\omniverse_soft\OpenUSD-Code-Samples\sample`

➤ `python transf_gl.py`

➤ `usdview transf_gl.usda`

- 見た目はローカルと同じ

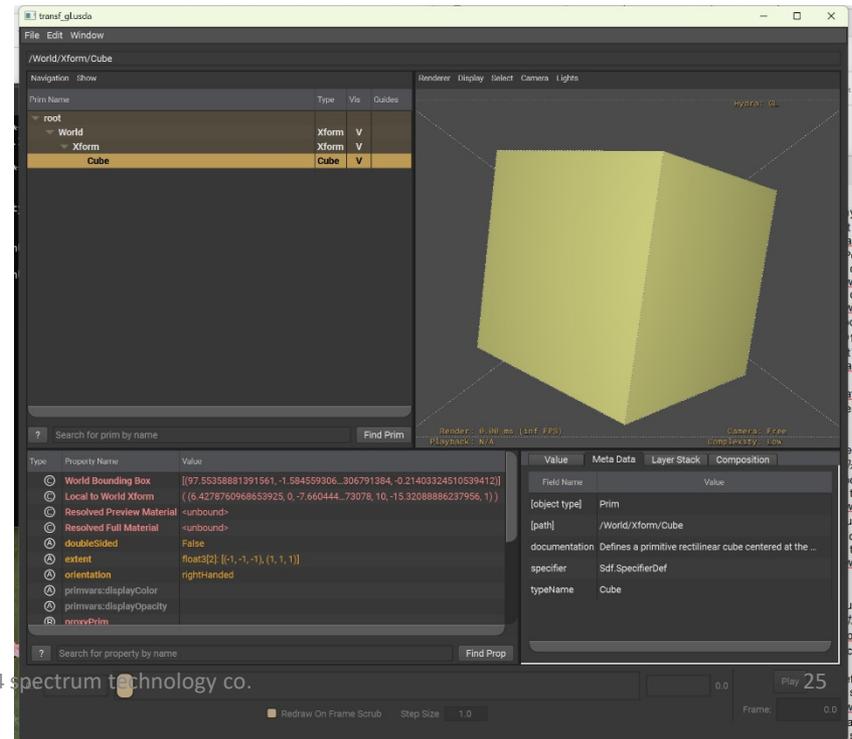
コマンド入力

> cd

C:\Users\skyri\Desktop\masa_legion\omniverse_soft\OpenUSD-Code-Samples\sample

> python transf_gl.py

> usdview transf_gl.usda



4. OpenUSD コード例

OpenUSDのcode sampleで基礎を学習します。

https://docs.omniverse.nvidia.com/dev-guide/latest/programmer_ref/usd.html#

⑭ Data Types: データ型

A) Convert Between VtArray and Numpy Array: 配列変換

- https://docs.omniverse.nvidia.com/dev-guide/latest/programmer_ref/usd/data-types/convert-vtarray-numpy.html

- データを配列への変換を行います。

- cd C:\Users\skyri\Desktop\masa_legion\omniverse_soft\OpenUSD-Code-Samples\sample

- python data_to_numpy.py

出力: numpyへ

```
[1. 2. 3.]
```

```
[4. 5. 6.]
```

```
[7. 8. 9.]]
```

- python data_from_numpy.py

- 出力: numpyから

```
[(1, 2, 3), (4, 5, 6), (7, 8, 9)]
```

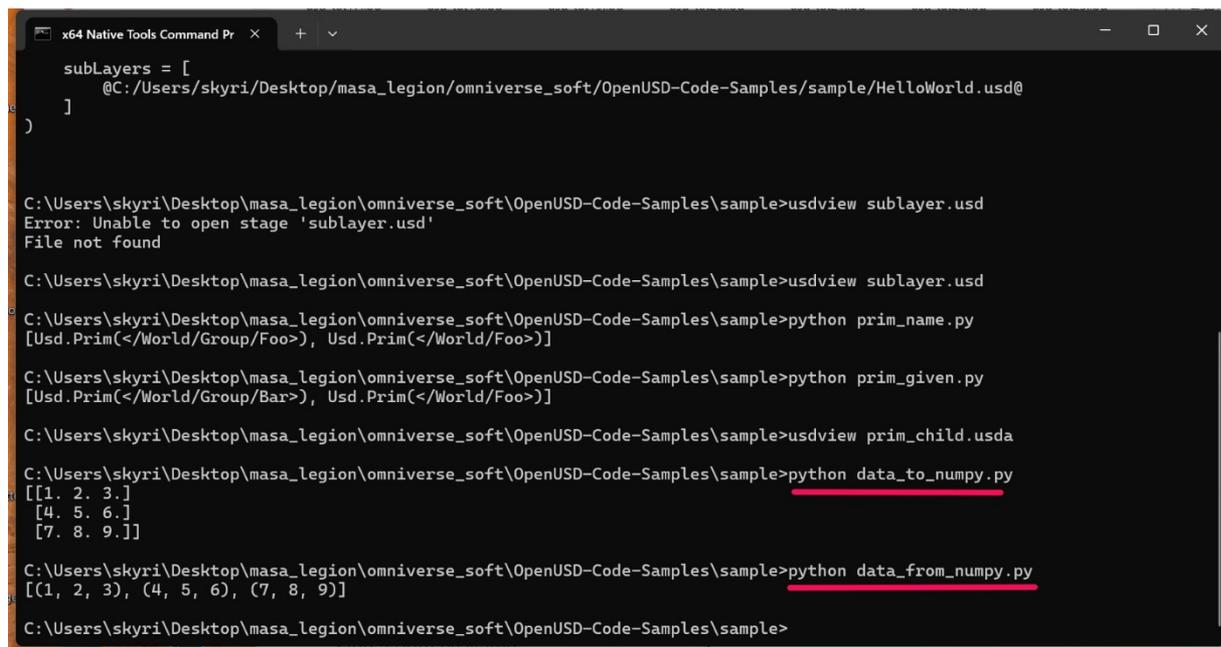
コマンド入力

```
> cd
```

```
C:\Users\skyri\Desktop\masa_legion\omniverse_soft\O  
penUSD-Code-Samples\sample
```

- python data_to_numpy.py

- python data_from_numpy.py



```
x64 Native Tools Command Pr x + v
subLayers = [
  @C:/Users/skyri/Desktop/masa_legion/omniverse_soft/OpenUSD-Code-Samples/sample>HelloWorld.usd@
]
)

C:\Users\skyri\Desktop\masa_legion\omniverse_soft\OpenUSD-Code-Samples\sample>usdview sublayer.usd
Error: Unable to open stage 'sublayer.usd'
File not found

C:\Users\skyri\Desktop\masa_legion\omniverse_soft\OpenUSD-Code-Samples\sample>usdview sublayer.usd

C:\Users\skyri\Desktop\masa_legion\omniverse_soft\OpenUSD-Code-Samples\sample>python prim_name.py
[Usd.Prim(</World/Group/Foo>), Usd.Prim(</World/Foo>)]

C:\Users\skyri\Desktop\masa_legion\omniverse_soft\OpenUSD-Code-Samples\sample>python prim_given.py
[Usd.Prim(</World/Group/Bar>), Usd.Prim(</World/Foo>)]

C:\Users\skyri\Desktop\masa_legion\omniverse_soft\OpenUSD-Code-Samples\sample>usdview prim_child.usda

C:\Users\skyri\Desktop\masa_legion\omniverse_soft\OpenUSD-Code-Samples\sample>python data_to_numpy.py
[[1. 2. 3.]
 [4. 5. 6.]
 [7. 8. 9.]]

C:\Users\skyri\Desktop\masa_legion\omniverse_soft\OpenUSD-Code-Samples\sample>python data_from_numpy.py
[(1, 2, 3), (4, 5, 6), (7, 8, 9)]

C:\Users\skyri\Desktop\masa_legion\omniverse_soft\OpenUSD-Code-Samples\sample>
```

ST Spectrum Technology

5.Omniverse応用

Windowsにomniverse app,extention,serviceを使ったアプリ作成など。

<https://docs.omniverse.nvidia.com/dev-guide/latest/tutorials.html>

SDK kitの各位置付けになります。

A) Application

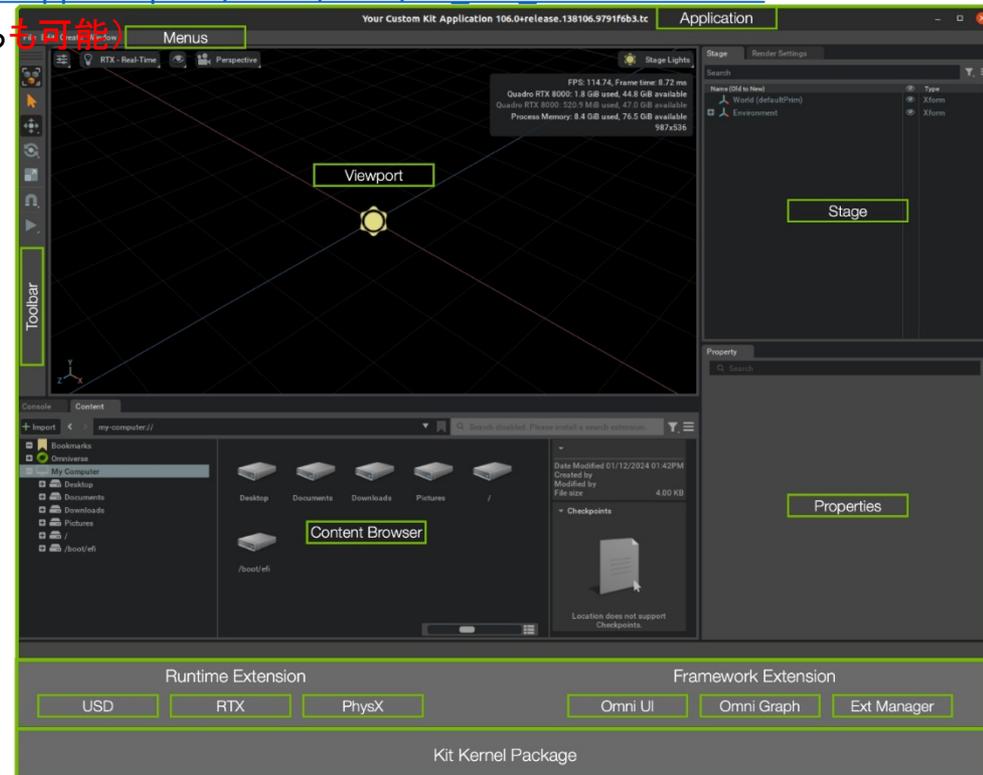
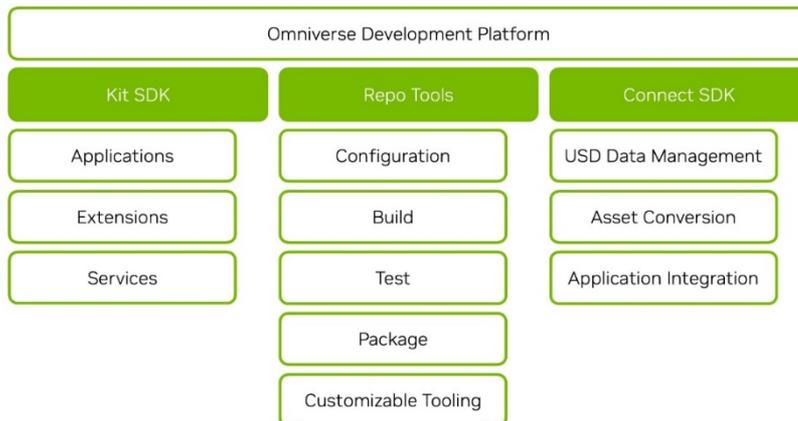
- Omniverse Kit SDK を使用してカスタム アプリケーションを作成するプロセスをユーザーにガイドします。
- https://docs.omniverse.nvidia.com/kit/docs/kit-app-template/latest/docs/kit_sdk_overview.html#
- Usd composerの見方 (omniverse launcherから **も可能**)

B) Extention

- 外部APIとの接続などの拡張機能。

C) service

- 拡張サービスなど



ST Spectrum Technology

5.Omniverse 応用

Windowsにomniverse app,extention,serviceを使った操作。

<https://docs.omniverse.nvidia.com/dev-guide/latest/tutorials.html>

C) service

• インストール

• https://docs.omniverse.nvidia.com/kit/docs/kit-app-template/latest/docs/extending_services.html

• Visual studio codeを使用、new terminal

• テンプレート作成

➢ C:\Users\skryi\Desktop\masa_legion\omniverse_soft\kit-app-template

➢ .\repo.bat template new

• 設定変更: 弊社提供のファイル参照

• 依存追加

• C:\Users\skryi\Desktop\masa_legion\omniverse_soft\kit-app-template\source\extensions\tutorial.service.setup\configの extension.toml
"omni.kit.primitive.mesh" = {}

• C:\Users\skryi\Desktop\masa_legion\omniverse_soft\kit-app-template\source\extensions\tutorial.service.setup\service\setup_service.py

• from pxr import Gf, Sdf, UsdGeom, UsdShade

• service.py修正

Replace the existing CubeDataModel with the following SceneDataModel

• ビルド

➢ .\repo.bat build

• BUILD (RELEASE) SUCCEEDED (Took XX.XX seconds) かなればok

• ランチ

➢ .\repo.bat launch

• tutorial.serviceを選択

• <http://localhost:8011/docs>

• Try it out>excuteでusda出力

➢ cd C:\asset_write_path

➢ usdview scene.usda

usd composerで見た方がよいかも

コマンド入力

> cd C:\Users\skryi\Desktop\masa_legion\omniverse_soft\kit-app-template

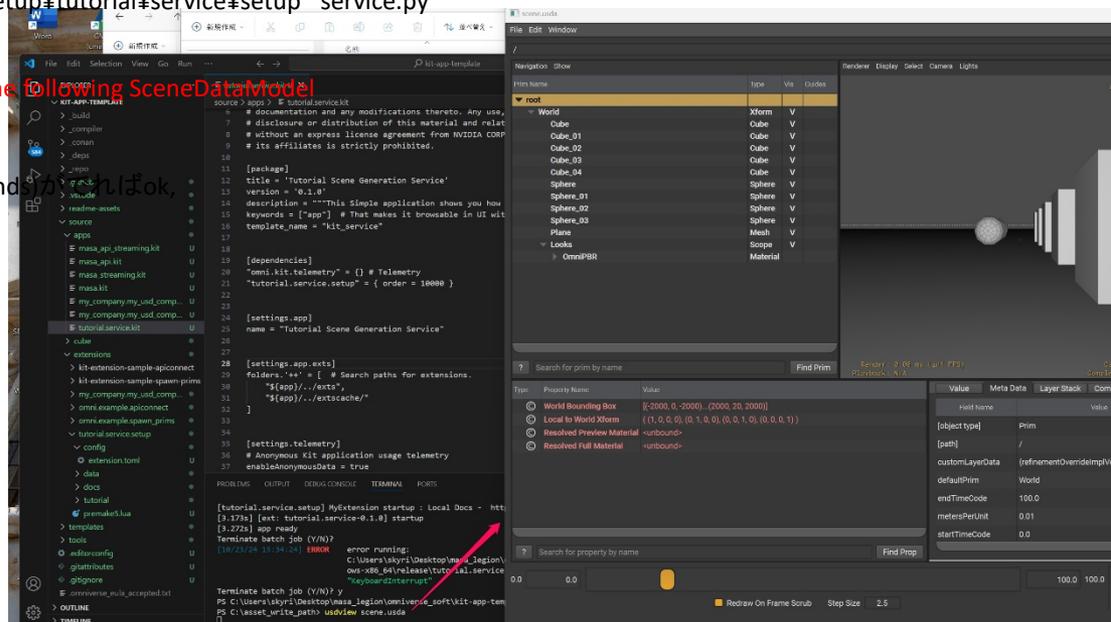
> .\repo.bat template new

> .\repo.bat build

> .\repo.bat launch

> cd C:\asset_write_path

> usdview scene.usda



Omniverseの応用事例。

<https://developer.nvidia.com/omniverse#section-learning-library>

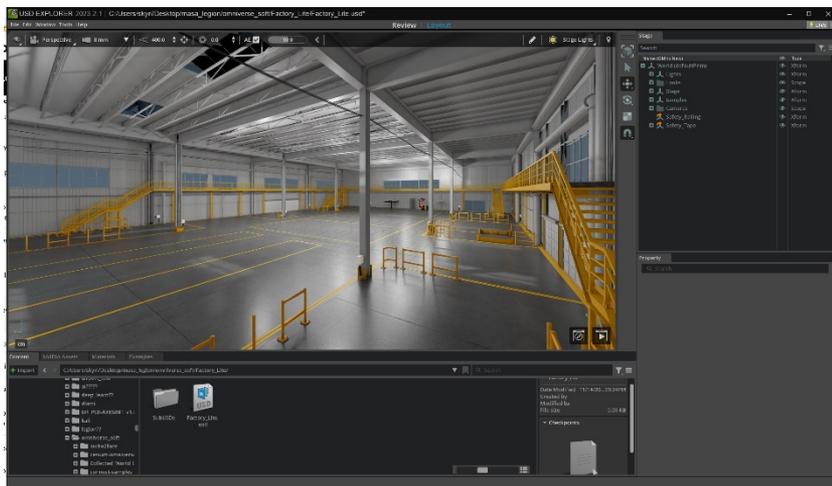
E) Industrial Digital Twins

● 概要

- Omniverseプラットフォームを使用してデジタルツインのアプリケーションを開発します。デジタルツインのワークフローを検討する際の重要な概念を確認し、USD Explorer テンプレートアプリケーションの組み込み機能について説明します。OpenUSD フレームワークを使用して物理的に正確な3Dシーンを作成し、テンプレートアプリケーション内で提供される主要なサンプル拡張機能を活用する方法を学びます。次に、動体拡張機能を追加してアプリケーションをカスタマイズし、シーンに物理を適用します。

● 設定

- cd C:¥Users¥skyri¥Desktop¥masa_legion¥omniverse_soft¥Factory_Lite
- Omniverse launcher>usd composerを起動し、Factory_Lite.usdを開く、元がbinaryなのでFactory_Lite2.usdaで保存
- vscodeでFactory_Lite2.usdaを開く。元のusdはbinaryなので編集できないためusdaに変換



ST Spectrum Technology

5.Omniverse応用

コマンド入力

- cd
- C:\Users\skyri\Desktop\masa_legion\omniverse_soft\Factory_Lite
- Factory_Lite4.usda

Omniverseの応用事例。

<https://developer.nvidia.com/omniverse#section-learning-library>

E) Industrial Digital Twins

- 概要

- 設定

- cd C:\Users\skyri\Desktop\masa_legion\omniverse_soft\Factory_Lite

- Factory_Lite4.usdaにサンプル

- 動体追加

- 動体設定

- beltconveyorに動きを設定:property>add>physics>Rigid Body with Colliders Preset

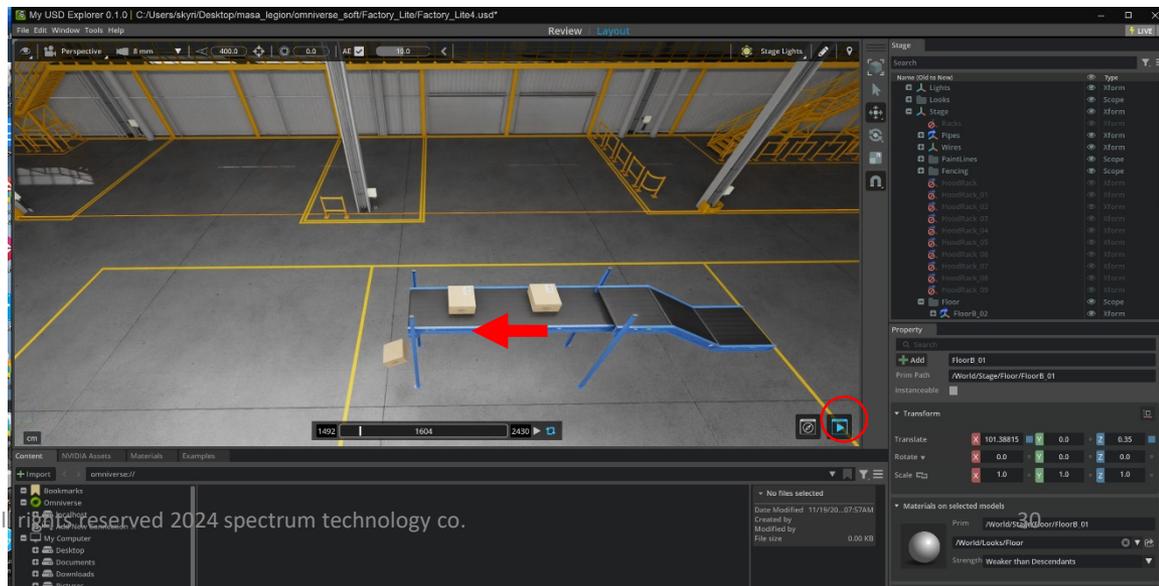
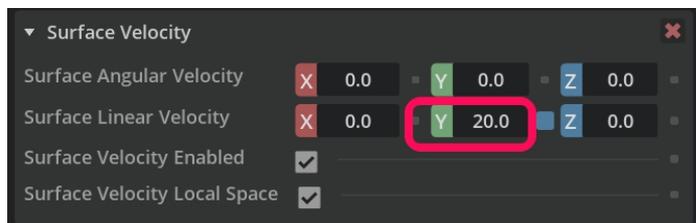
- Kinematic Enabled: On, Velocities in Local Space: On

- property>add>Physics > Surface Velocity

- Surface Linear Velocity の Y: 20

- ファイルを保存

- プレイし、確認、逆の方向に動きます。-20に再設定、もう一つのコンベアも同様に設定すると正常に動きます。



6. 生成AIを活用したOpenusd事例

生成AIを活用したOpenusd事例を紹介します。NVIDIA NIM マイクロサービスとしてクラウドで提供。usdのサンプル検索、作成したusdファイルの検証、USDcodeを生成できるモデル

コマンド入力

- cd
- C:¥Users¥skyri¥Desktop¥masa_legion¥omniverse_soft¥openusd_ai
- WarningSign_A04_PR_NVD_01.usd

A) USD検索

- <https://build.nvidia.com/nvidia/usdsearch>
- nvidia生成のusd検索ツール: ブラウザで利用: usdsearch
- テキスト又は画像から関連のassetsファイル(usd)を紹介
- cd C:¥Users¥skyri¥Desktop¥masa_legion¥omniverse_soft¥openusd_ai
- WarningSign_A04_PR_NVD_01.usdファイルをusd exploreなどで開いて確認

