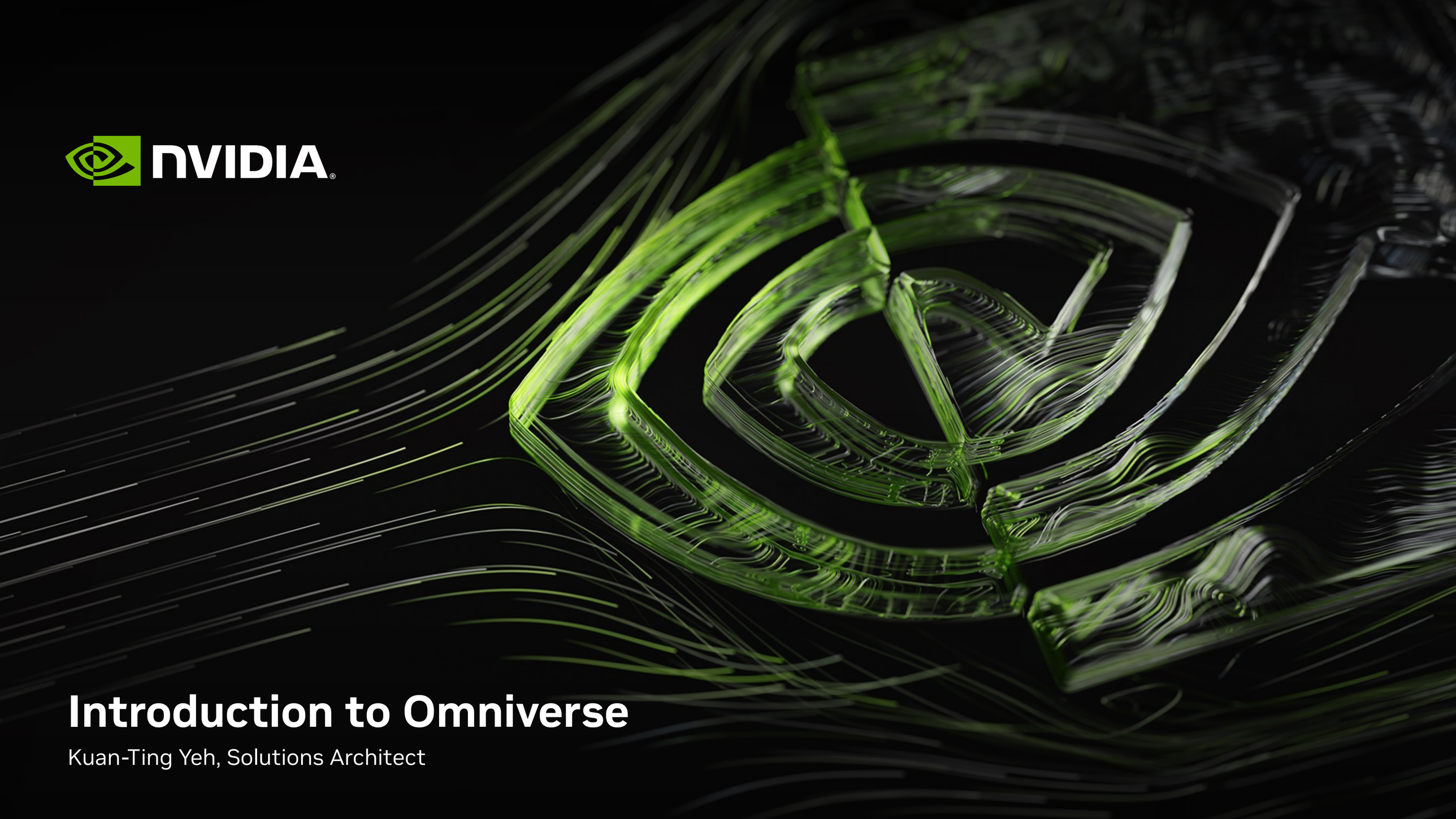


Introduction to Omniverse

Kuan-Ting Yeh, Solutions Architect

AGENDA

- Introduction to Nvidia Omniverse
  Modern Collaboration Workflows & Universal Scene Description (USD)
- Five Foundational Components (Demo)
  Nucleus, Connect, Kit, Simulation, RTX Renderer
- Building the Digital Twin (Hands-On)
  Visualizing weather prediction on Omniverse
- **Unified Industrial Digitalization**
  Digital twin development & Synthetic data generation
- **Extension & Omni.UI (Hands-On)**
  Building an extension to visualize weather prediction using FourCastNet
- **Getting started**
  Download & Resources

Unified Industrial Digitalization

Digital twin development & Synthetic data generation

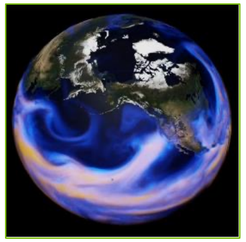
NVIDIA Omniverse

Cloud Native Platform for Connecting, Building and Operating Industrial Metaverse Applications

AI Avatars

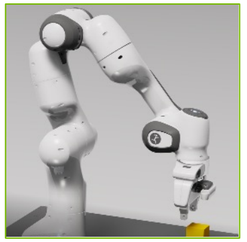
3D Design

Autonomous Vehicles

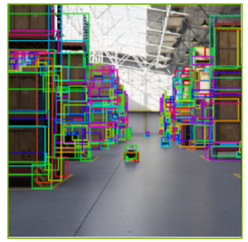


Scientific Digital Twin

Industrial Digital Twin



Robotics



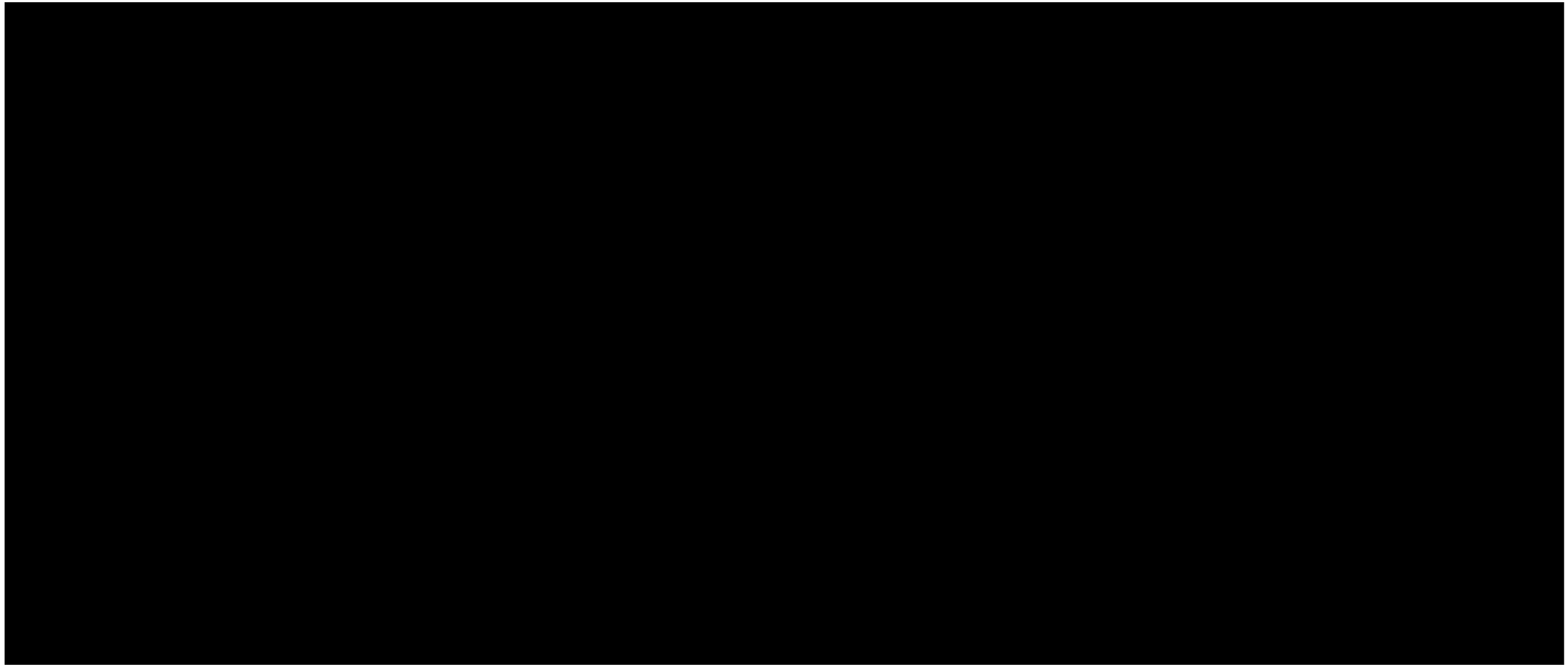
Synthetic Data Generation

Audio2Face | USD Composer | DRIVE Sim | Isaac Sim | Replicator | USD-GDN Publisher

NVIDIA Omniverse

Before Industrial Digital Twin

Designing a System



The Founder 'Speedy System' Featurette (2017)

"With Omniverse and AI, we set up new factories faster and more efficiently"

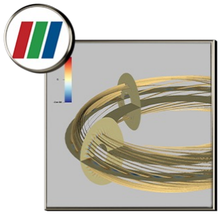
“It all starts with planning - a complex process in which we need to connect many tools, datasets and specialists around the world. Traditionally we are limited as data is managed separately in a variety of systems and tools. Today we are developing custom Omniverse applications to connect our existing tools, know-how, and teams in a unified view.”

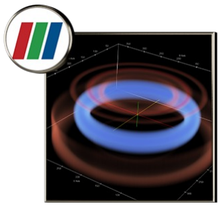
Dr Milan Nedeljković, Member of the Board of Management of BMW AG

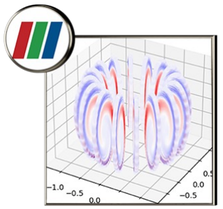
BMW GROUP

NVIDIA Omniverse for Scientific Computing

Connecting Complex HPC 3D and Simulation Workflows







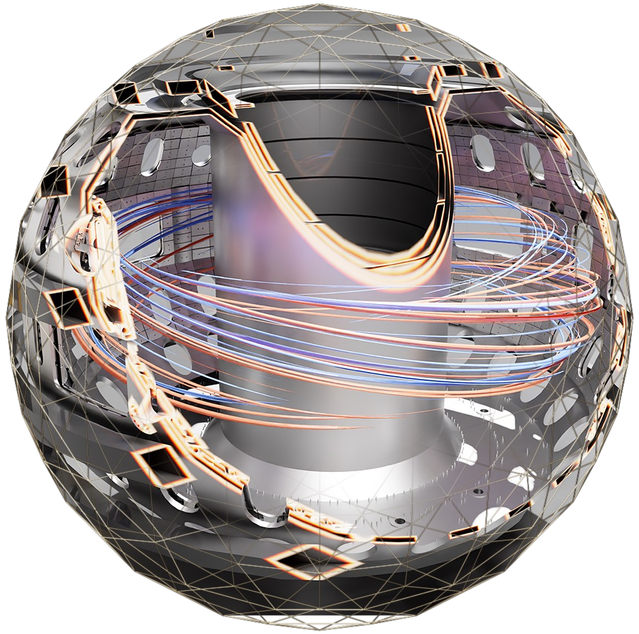
USD

USD

USD

USD

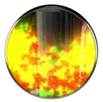
Omniverse Nucleus
Central Database



Renderer

Portal

RTX
Rendering



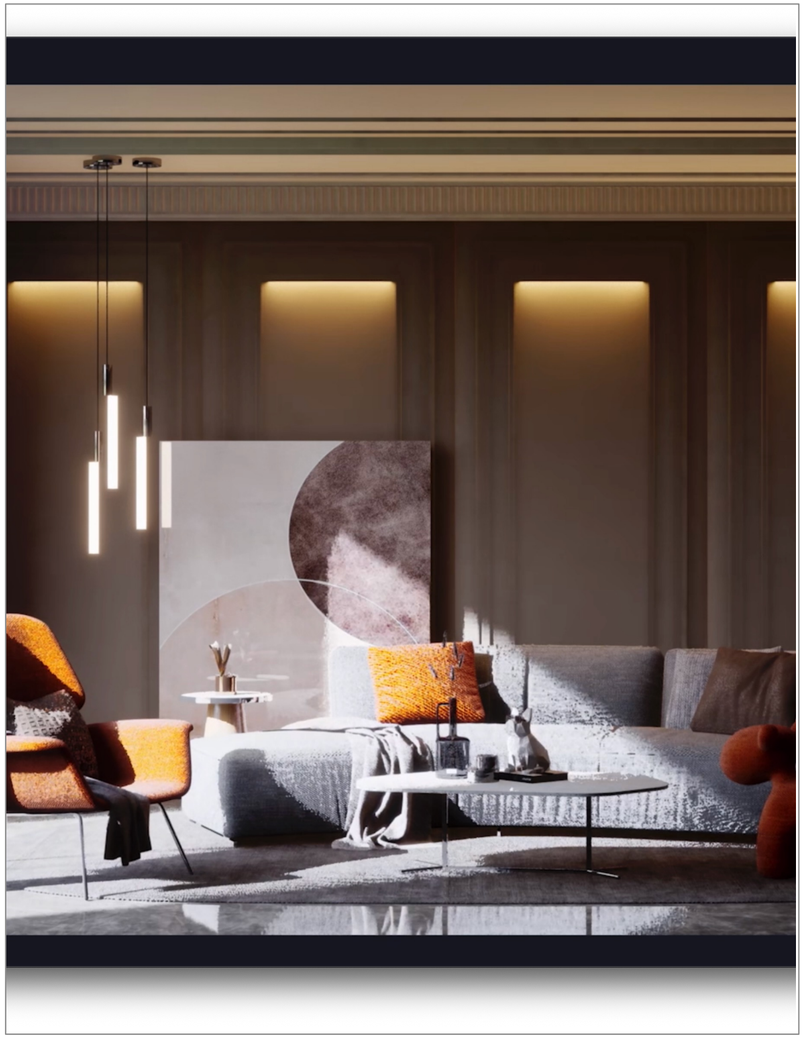
IndeX
Rendering

NeuralVDB

PhysX

AI &
Modulus

最佳化建築、工程、營造與營運工作流程



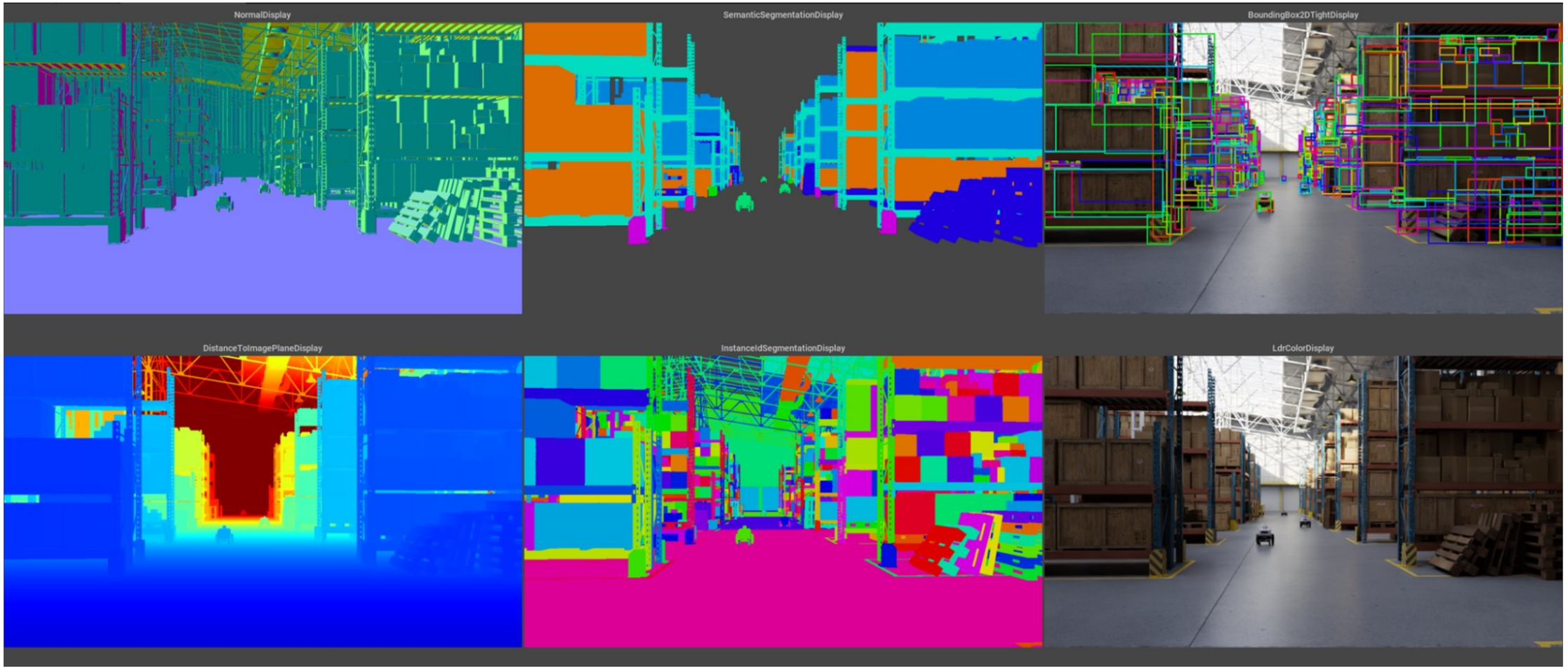
合成資料生成 (Synthetic Data Generation)

使用 Omniverse Replicator 在倉庫場景中加強感測器標註

Normal Display

Semantic Segmentation Display

Bounding Box2D Tight Display



Distance ToImage Display

Instncel dSegmentation Display

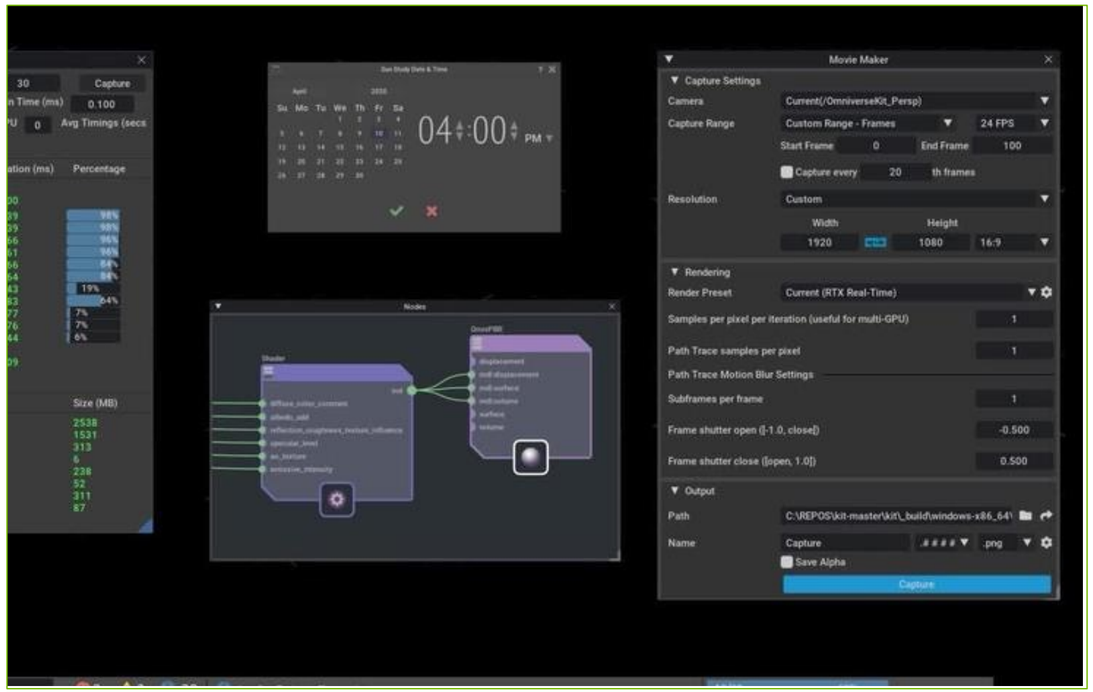
Ldr Color Display

Extension & Omni.UI (Hands-On)

Building an extension to visualize weather prediction using FourCastNet

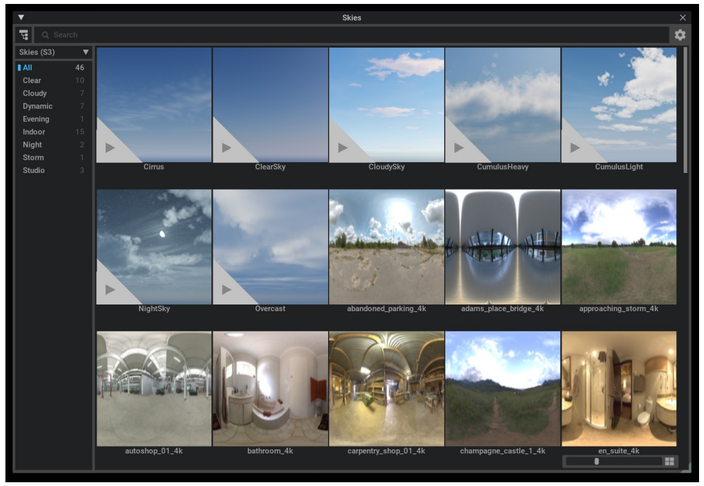
Custom Extensions

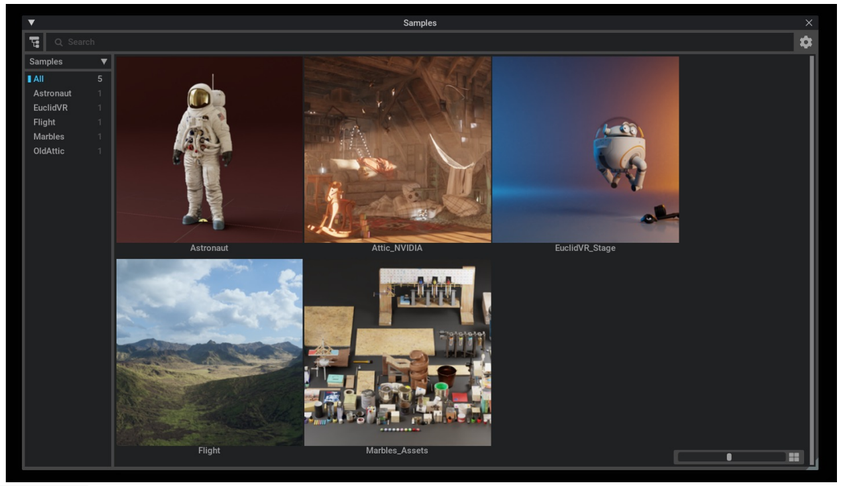
Omniverse is a development platform for building tools

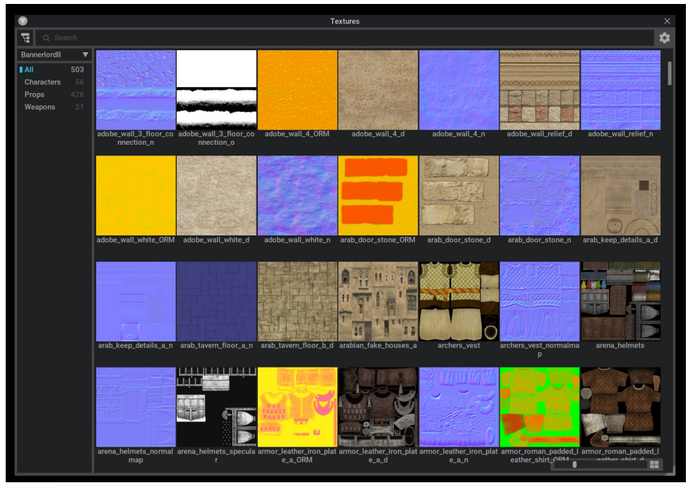


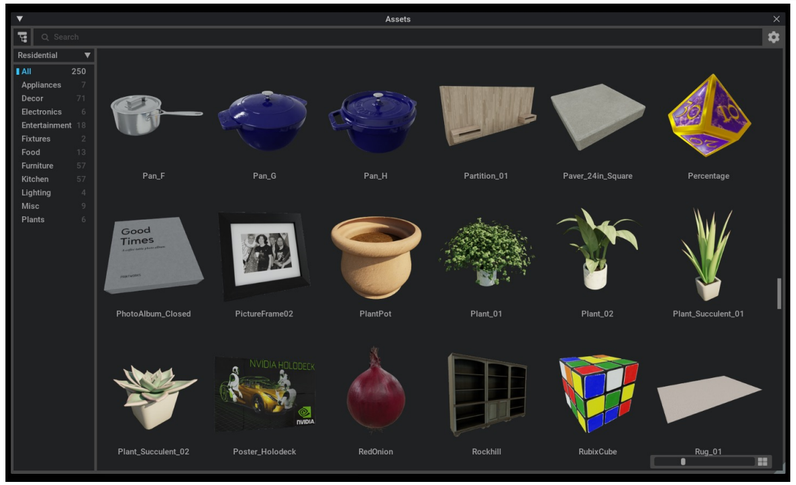
Extensions

Browser extension for creating any browser



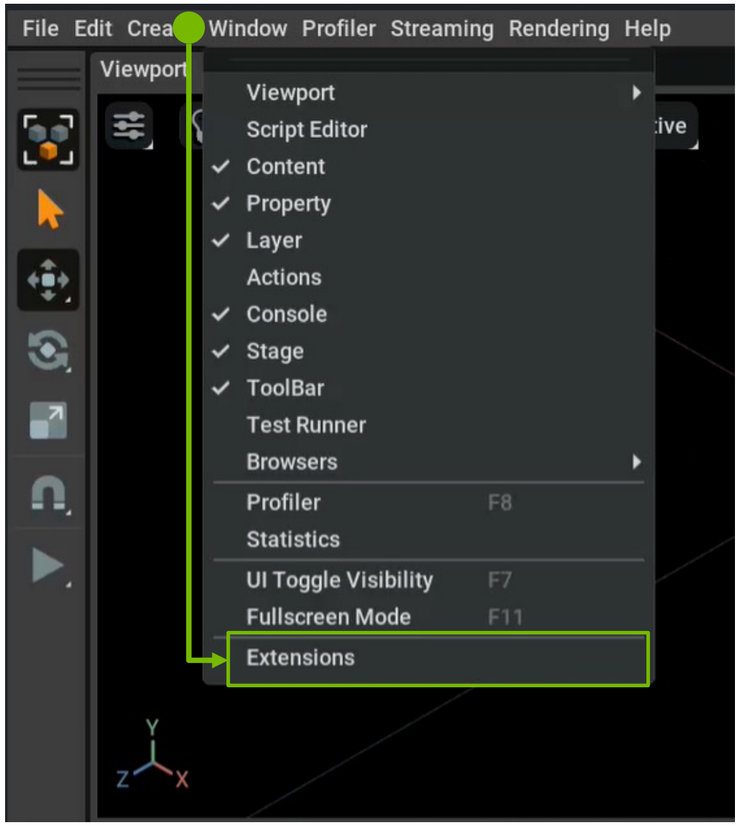




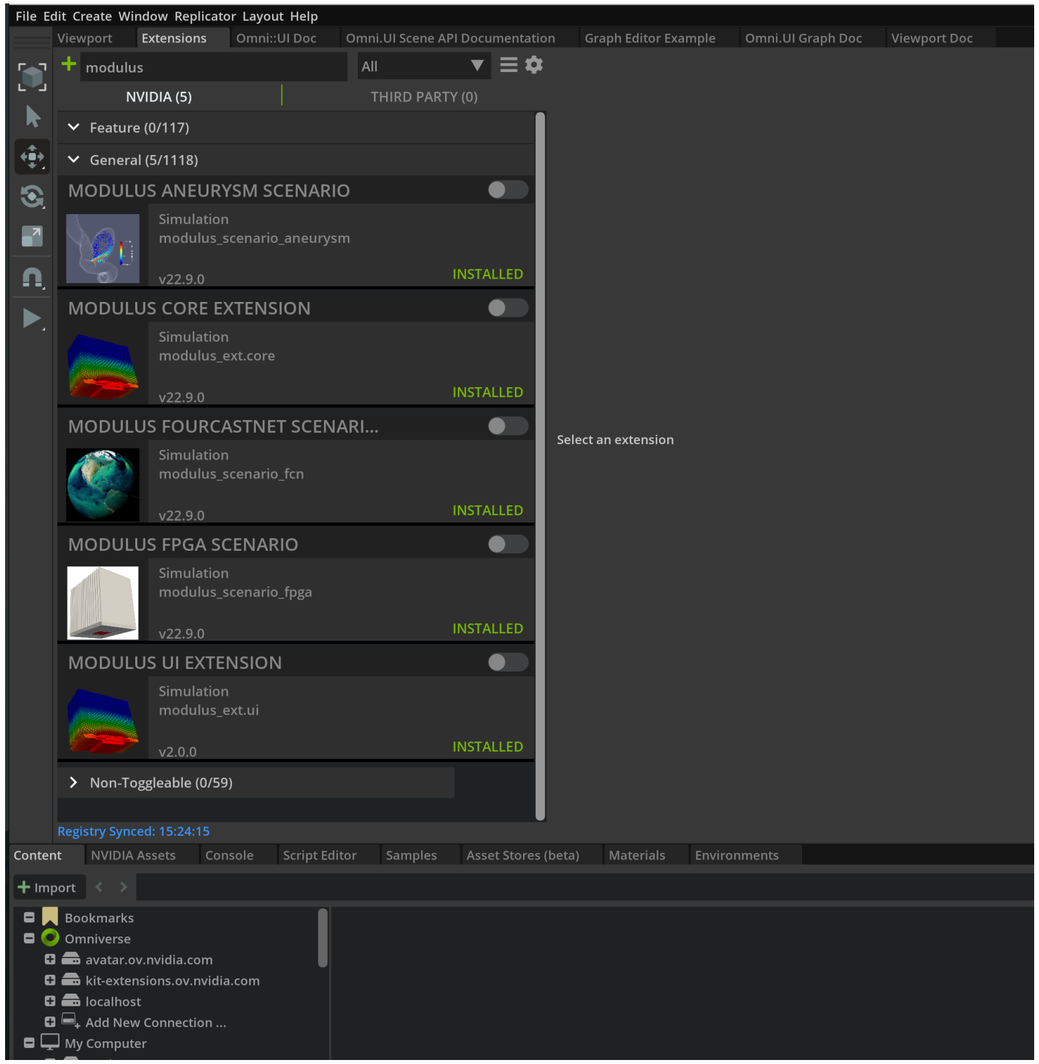


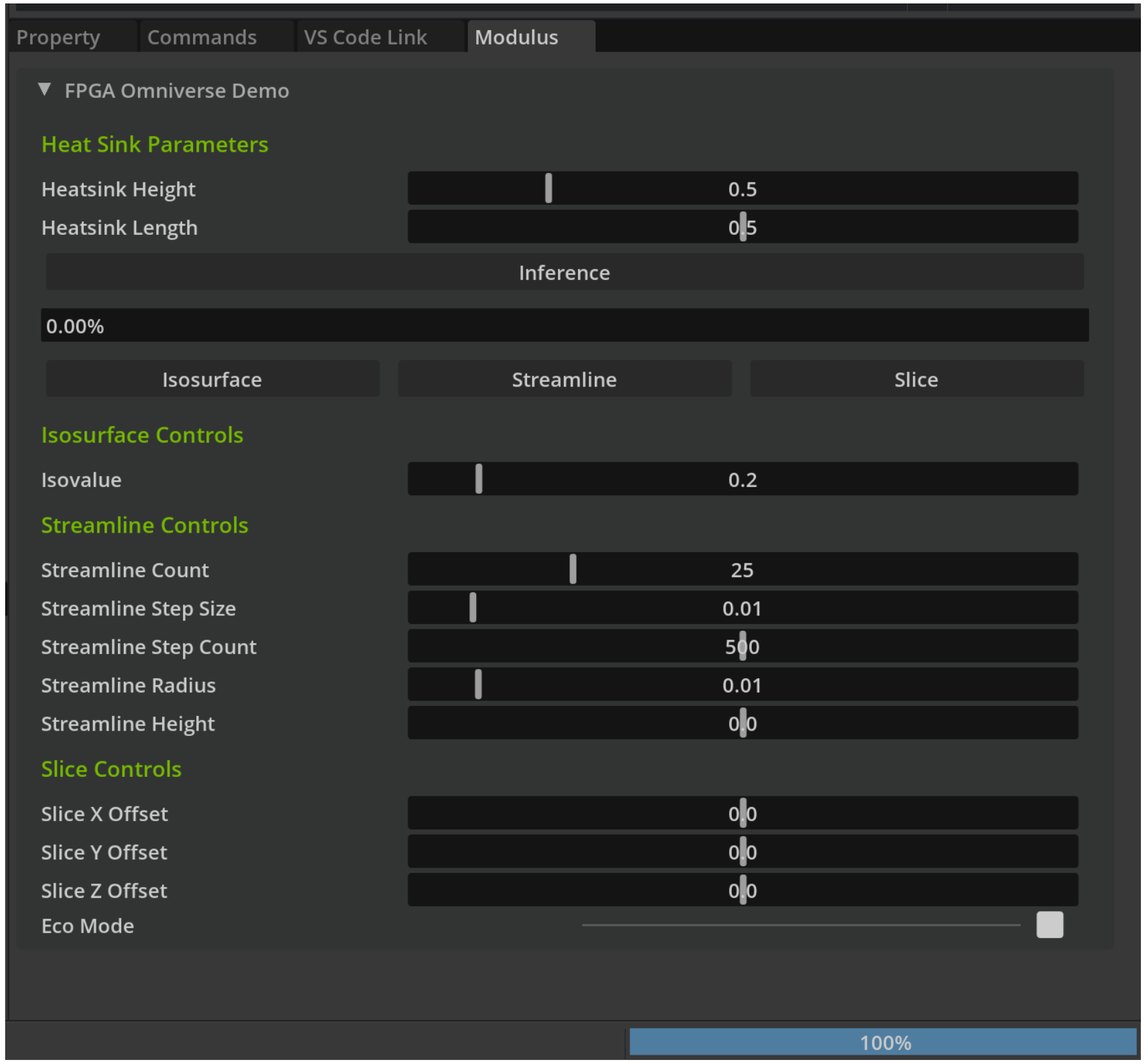
擴充套件管理 (Extension Manager)

NVIDIA提供了數百個由NVIDIA開發的擴充套件給開發者免費使用, 開發者可以自由的去編輯、修改、或是整合到自己的擴充套件或是應用程式中。不需要從0開始， 即可快速完成開發流程。

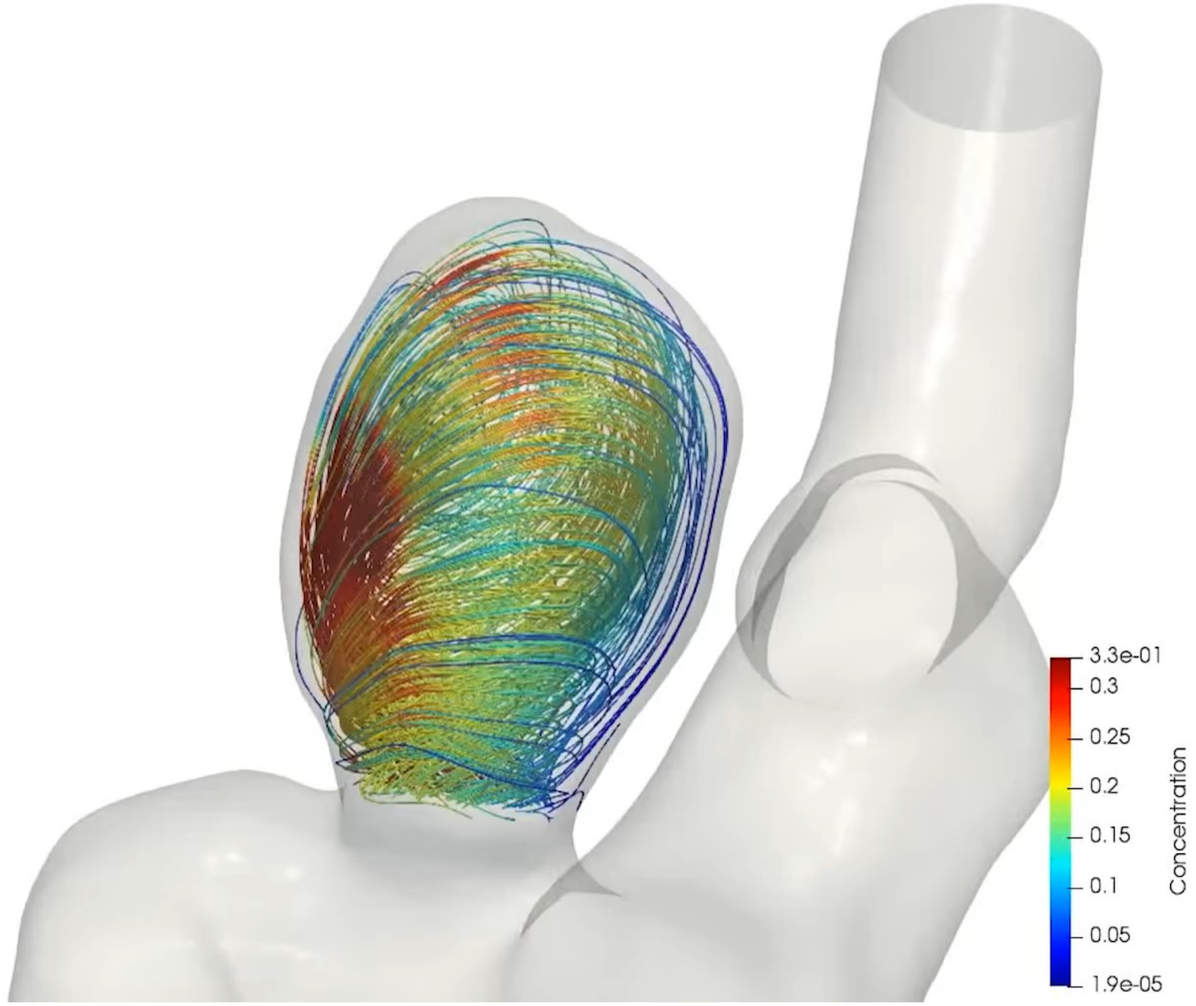


Modulus Extension



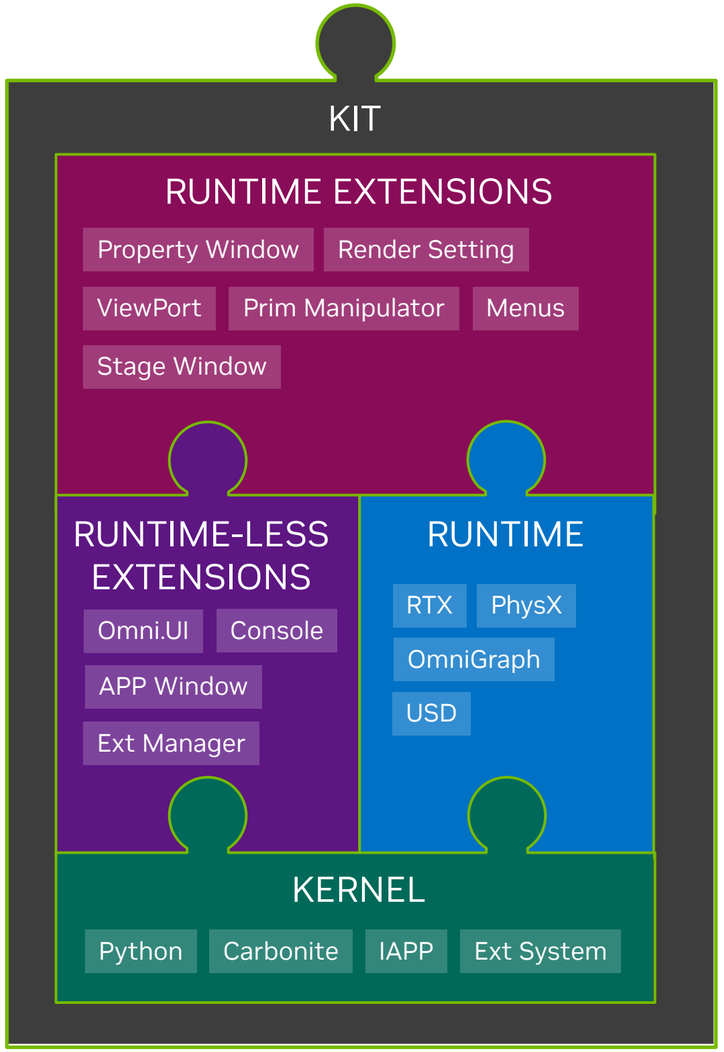


Modulus Extension



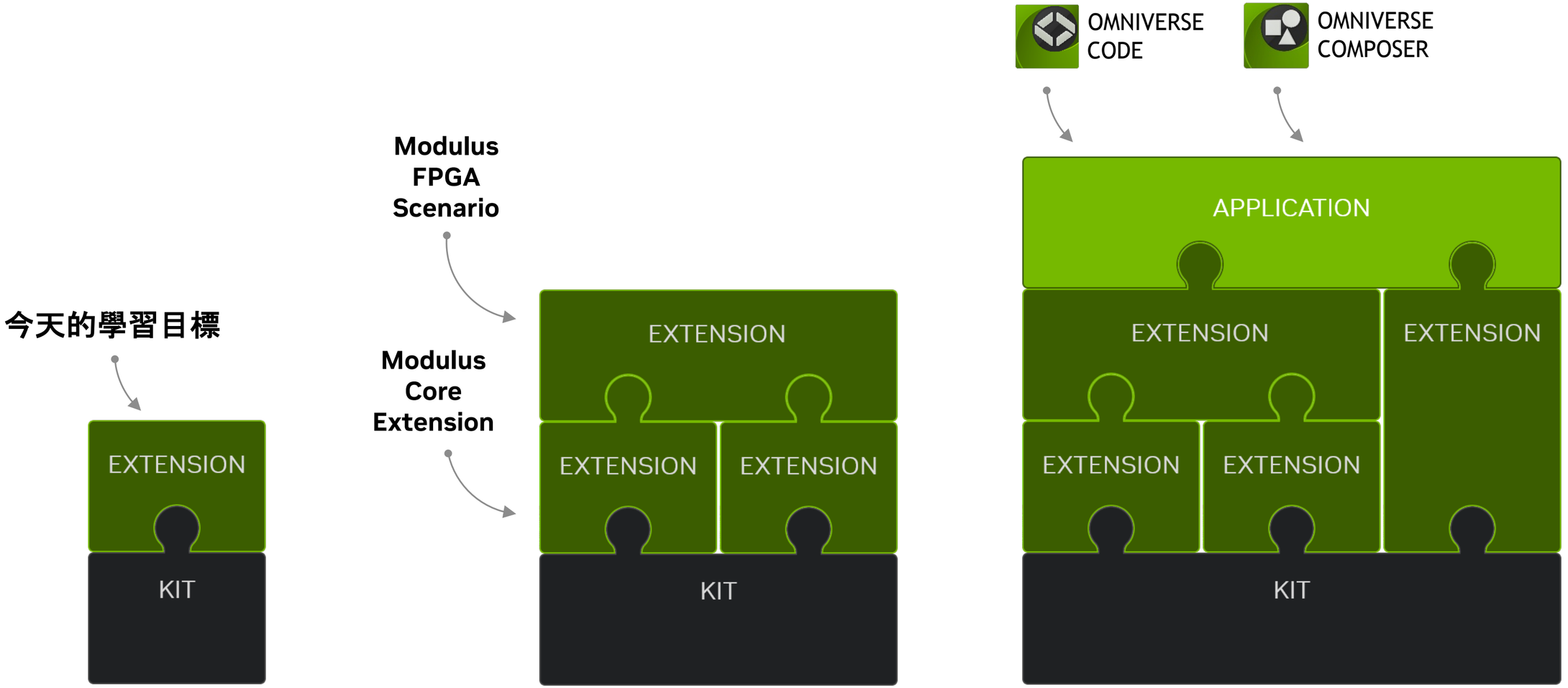
Kit Technology Stack

Kit is a modular stack of Runtimes and Extensions



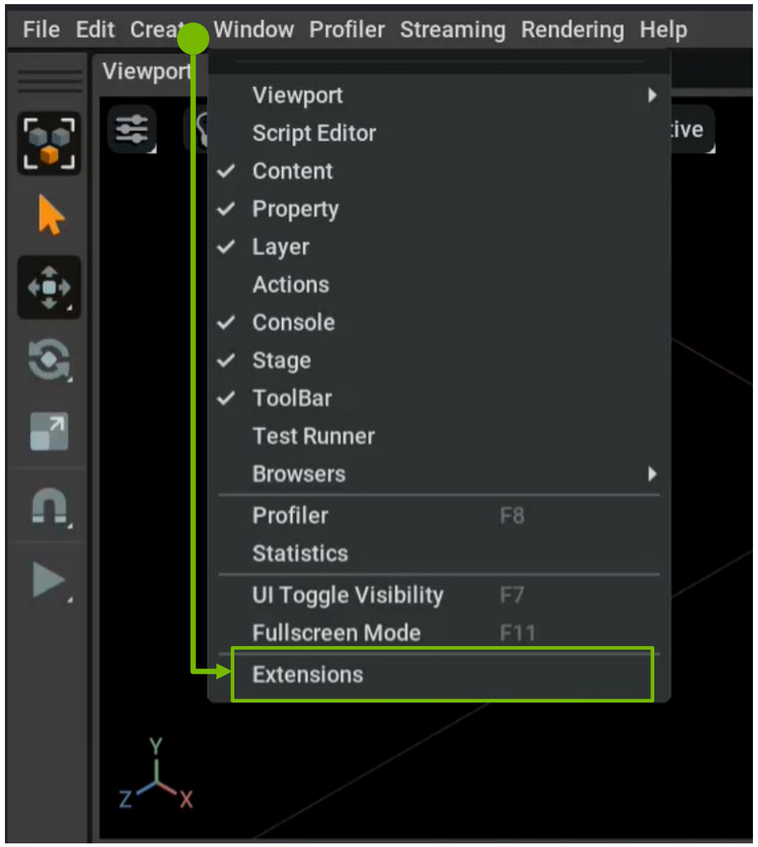
Extreme Modularity with Kit

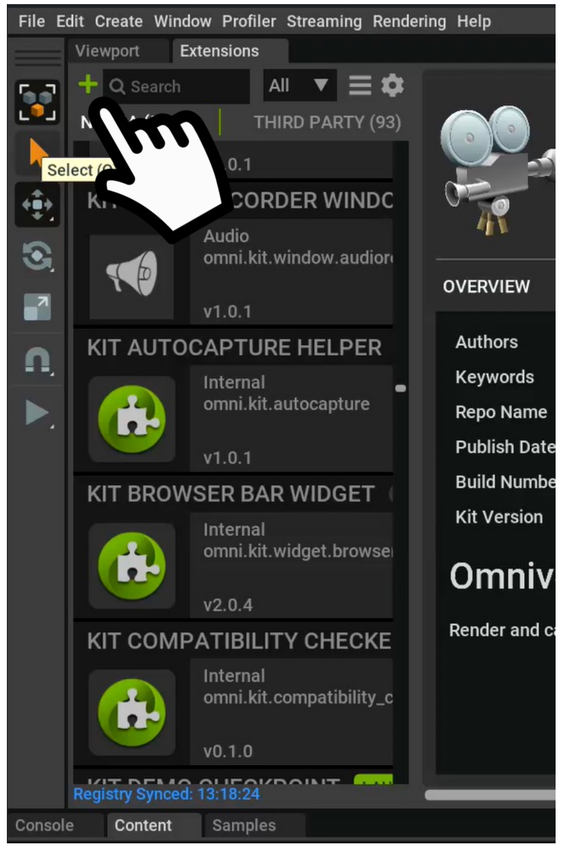
Solutions are stacks of Extensions



Hands-On (5分鐘)

新增一個Extension





- 選「New Extension Project Template」
- 選「/workspace/python/source_code/extension」
- Project Name: omniverse-project-navier-stokes
- ext_id: omniverse-project-navier-stokes
- 回到JupyterLab開啟extension.py檔案，路徑如下：

source_code -> extension -> omniverse-project-navier-stokes -> exts -> omniverse.extension.navierstokes -> omniverse -> extension -> navierstokes -> extension.py

omni.ui

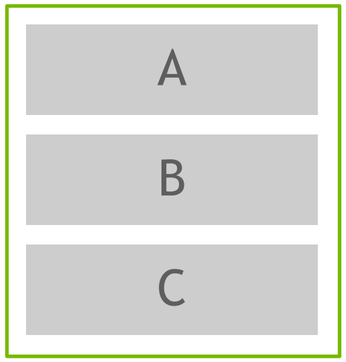
Building the User Interface

```
class OmniverseExtensionNavierstokesExtension(omni.ext.IExt):

  def on_startup(self, ext_id):
    print("[omniverse.extension.navierstokes] ... startup")

    self._window = ui.Window("My Window", width=300, height=300)
    with self._window.frame:
      with ui.VStack():
        label = ui.Label("empty")

        with ui.HStack():
          ui.Button("Add")
          ui.Button("Reset")

  def on_shutdown(self):
    print("[omniverse.extension.navierstokes] ... shutdown")
```

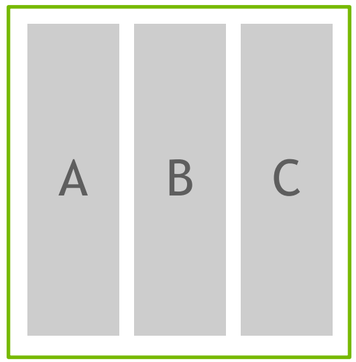
omni.ui

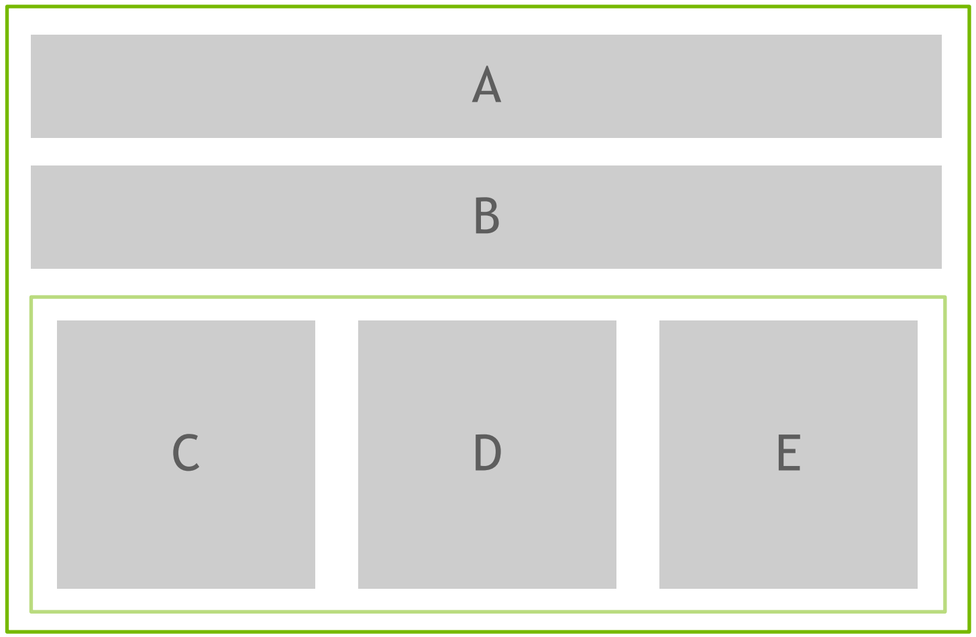
ui.HStack() & ui.Vstack()

```
with ui.VStack():
  ui.Button("A")
  ui.Button("B")
  ui.Button("C")
```




```
with ui.HStack():
  ui.Button("A")
  ui.Button("B")
  ui.Button("C")
```



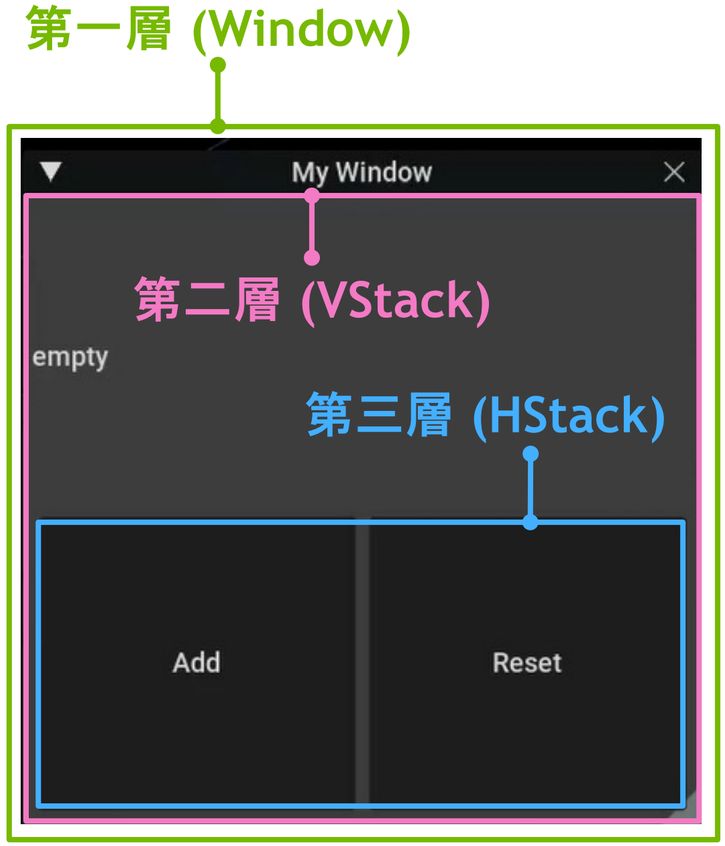



```
# A 2-layer UI structure
with ui.VStack():
  ui.Button("A")
  ui.Button("B")
  with ui.HStack():
    ui.Button("C")
    ui.Button("D")
    ui.Button("E")
```

omni.ui

Building the User Interface

```
class OmniverseExtensionNavierstokesExtension(omni.ext.IExt):

  def on_startup(self, ext_id):
    print("[omniverse.extension.navierstokes] ... startup")

    self._window = ui.Window("My Window", width=300, height=300)
    with self._window.frame:
      with ui.VStack():
        label = ui.Label("empty")

        with ui.HStack():
          ui.Button("Add")
          ui.Button("Reset")

  def on_shutdown(self):
    print("[omniverse.extension.navierstokes] ... shutdown")
```



omni.ui

More Components

"Button": {"stack_direction"}

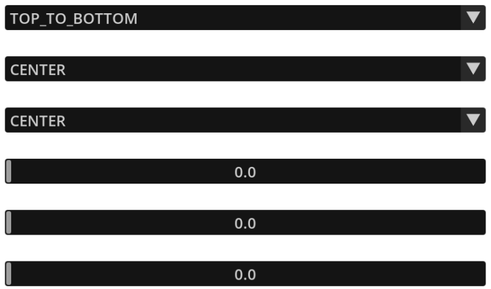
"Button.Image": {"alignment"}

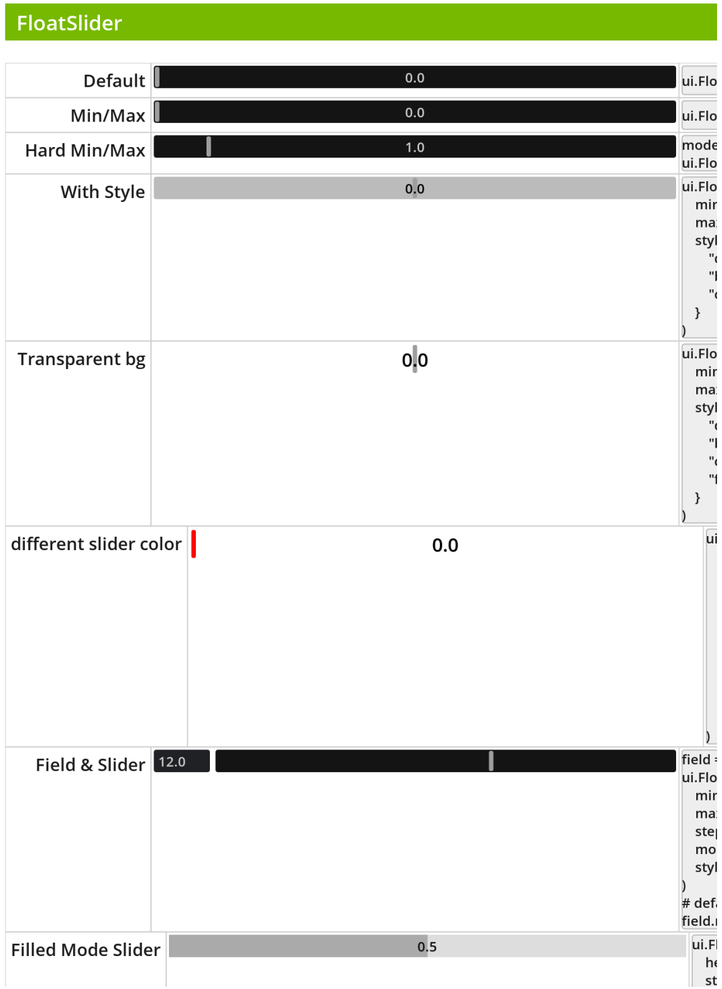
"Button.Label": {"alignment"}

padding

margin

Button.spacing

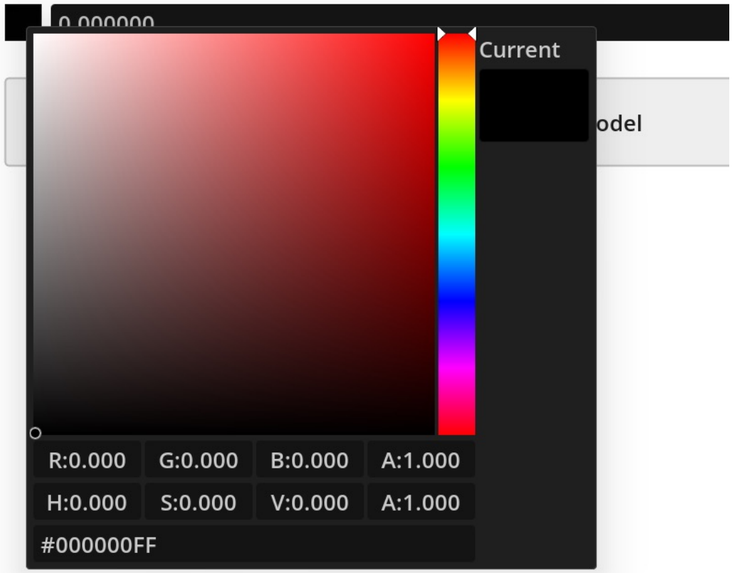




Drags connected to the color model


```
with ui.HStack(spacing=SPACING):
    color_model = ui.ColorWidget(0.125, 0.25, 0.5, width=0, height=0
    for item in color_model.get_item_children():
        component = color_model.get_item_value_model(item)
        ui.FloatDrag(component)
```

ComboBox connected to the color model.

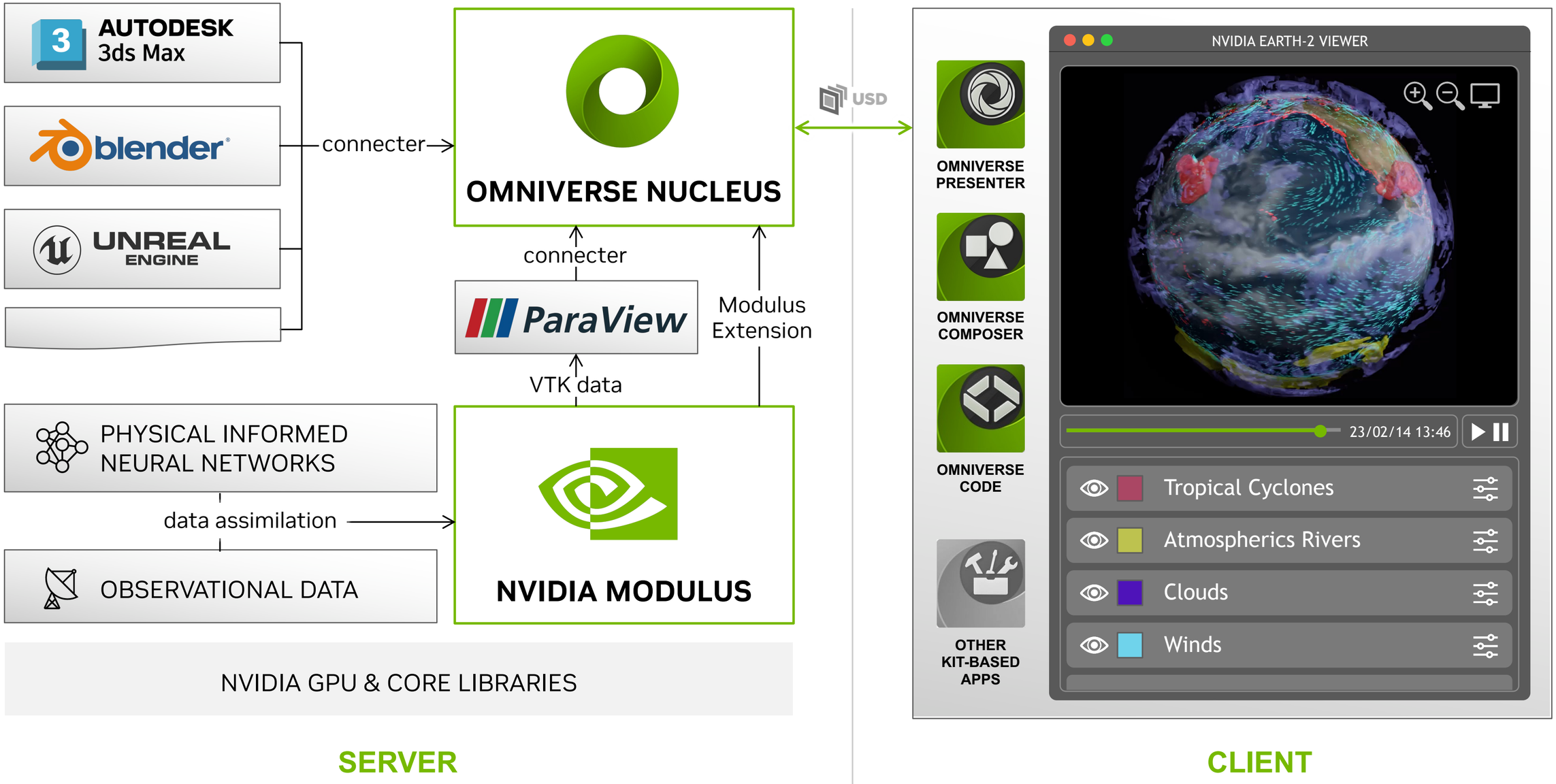


omni.ext

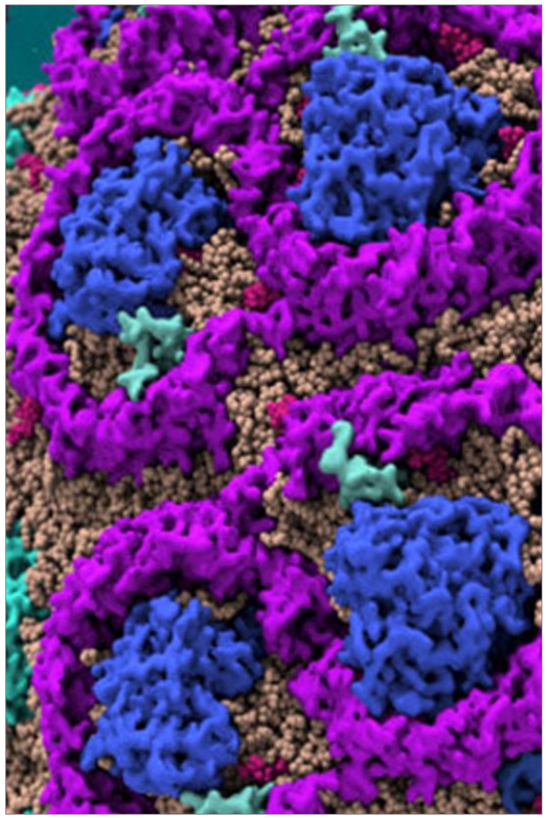
Adding functions to your UI

```
def on_startup(self, ext_id):
  print("[omniverse.extension.navierstokes] omniverse extension navierstokes startup")

  self._count = 0

  self._window = ui.Window("My Window", width=300, height=300)
  with self._window.frame:
    with ui.VStack():
      label = ui.Label("")

      def on_click():
        self._count += 1
        label.text = f"count: {self._count}"

      def on_reset():
        self._count = 0
        label.text = "empty"

      on_reset()

      with ui.HStack():
        ui.Button("Add", clicked_fn=on_click)
        ui.Button("Reset", clicked_fn=on_reset)
```

EARTH-2 Reference Framework

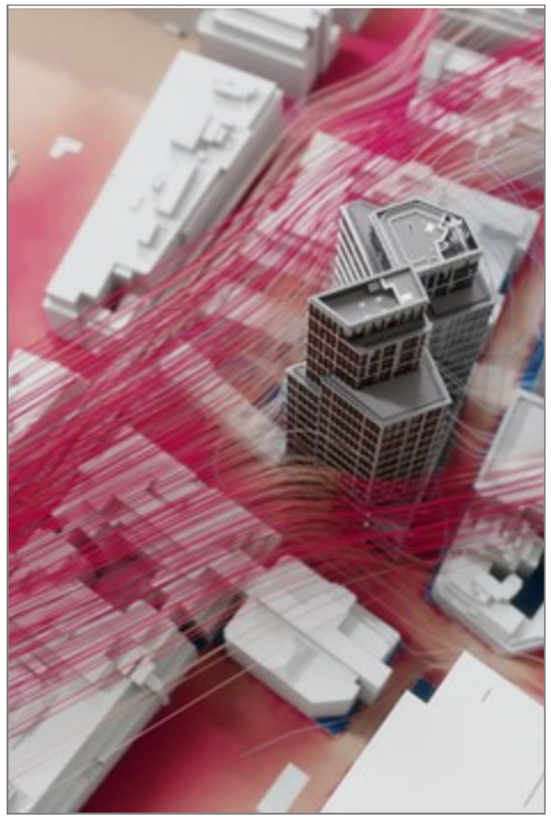


Digital Twins Can be Built at Every Scale

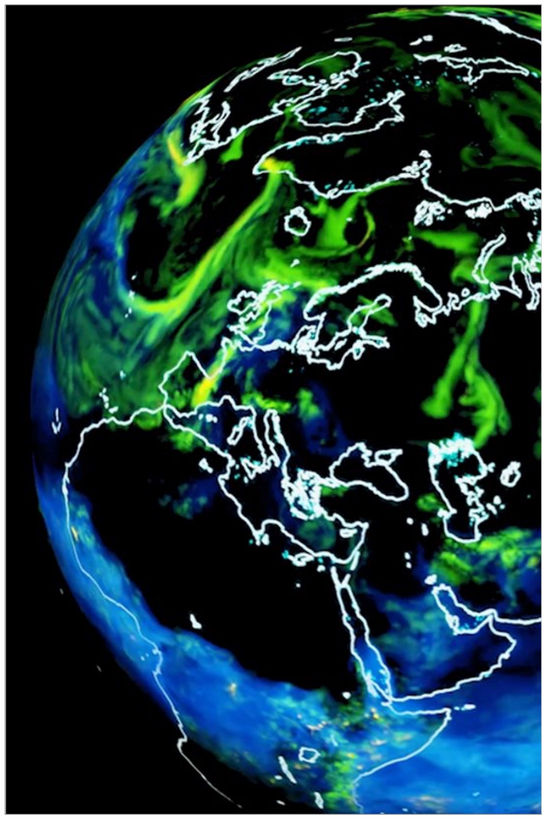


MOLECULAR

INDUSTRIAL



CITY

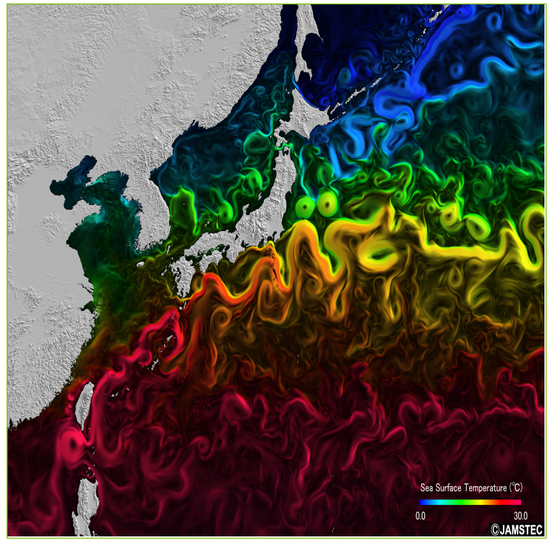


PLANETARY

Model Coupling to Answer What-If Questions

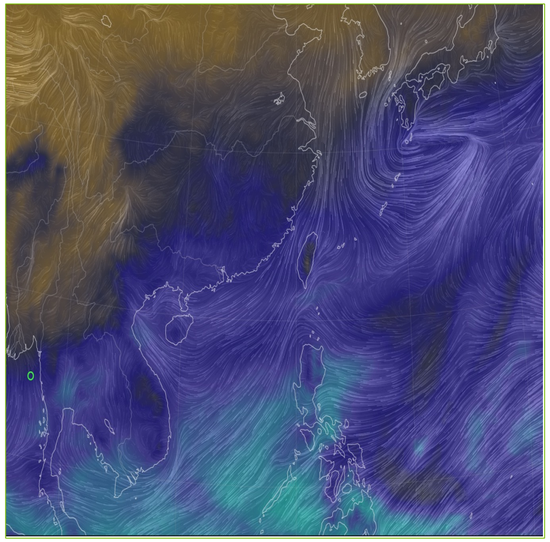
Coupling fast data-driven models together enables us to explore their interactions

Climate Twin



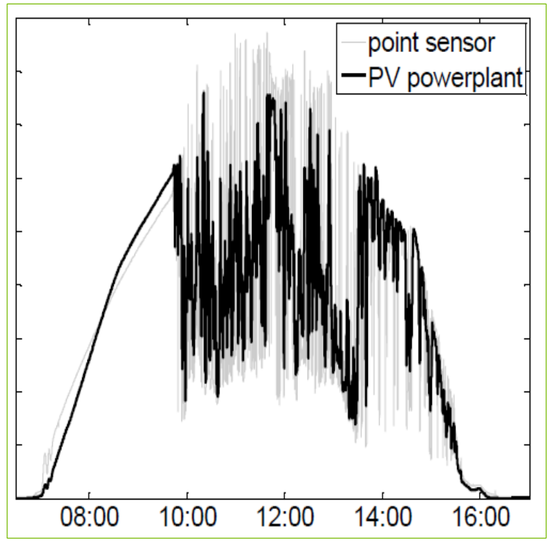
http://www.jamstec.go.jp/gallery/e/simulation/weather/002.html

Weather Twin



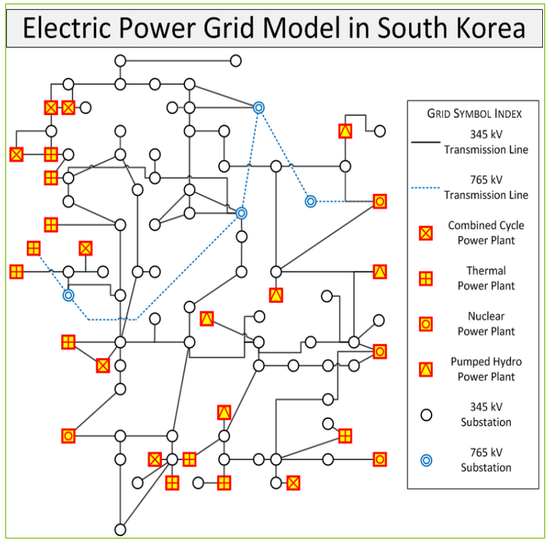
https://earth.nullschool.net/#current/wind/surface/level/overlay=total_precipitable_water/orthographic=-64.71,19.12,2322

Powerplant Twin



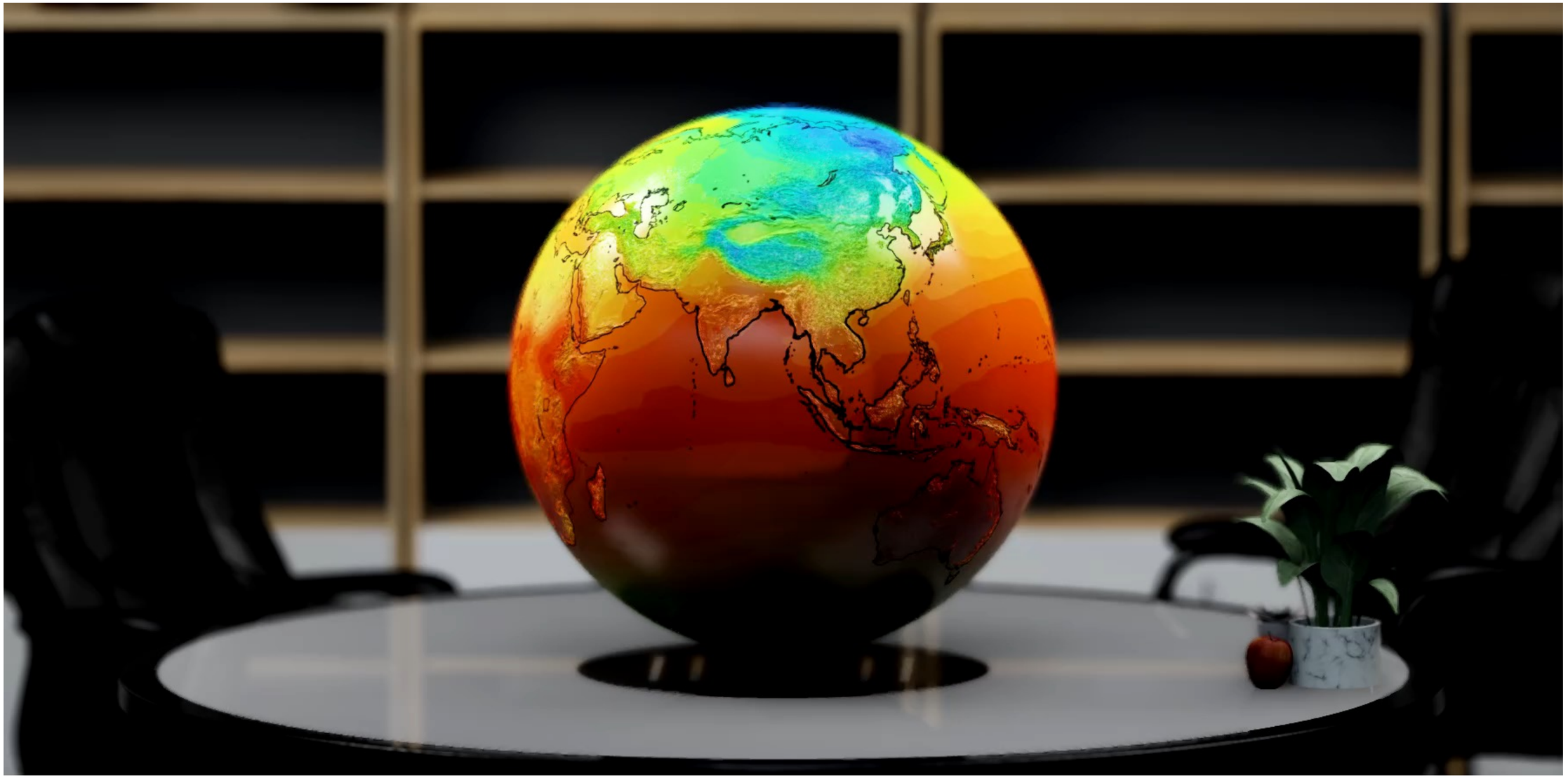
https://www.esig.energy/wiki-main-page/simulating-solar-power-plant-variability-a-review-of-current-methods/

Electric Grid Model



https://www.researchgate.net/figure/Overall-configuration-of-electric-power-grid-modelling_fig2_311217737

Simple Visualization



Get Started

Download & Installation for non-Enterprise customers

1. 到Omniverse官網點擊立即開始

https://www.nvidia.com/zh-tw/omniverse/

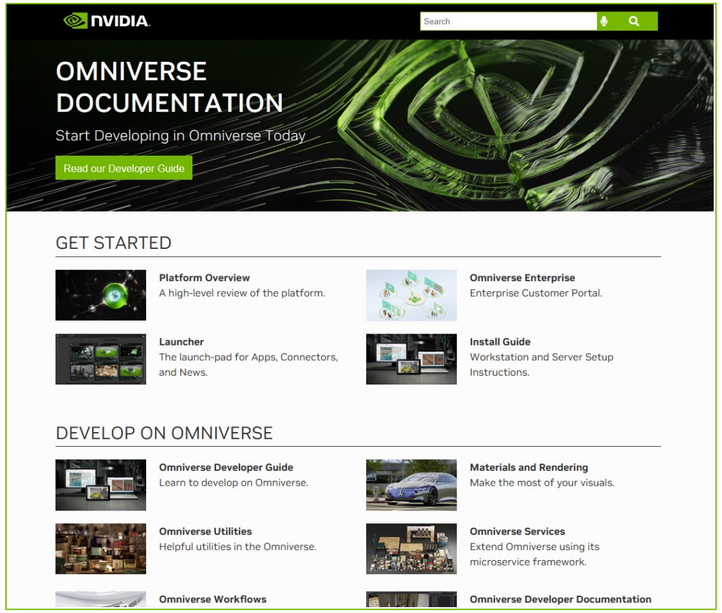
2. 選擇標準版， 並點擊「免費下載」

註冊Nvidia開發者帳號即可免費使用

學習資源

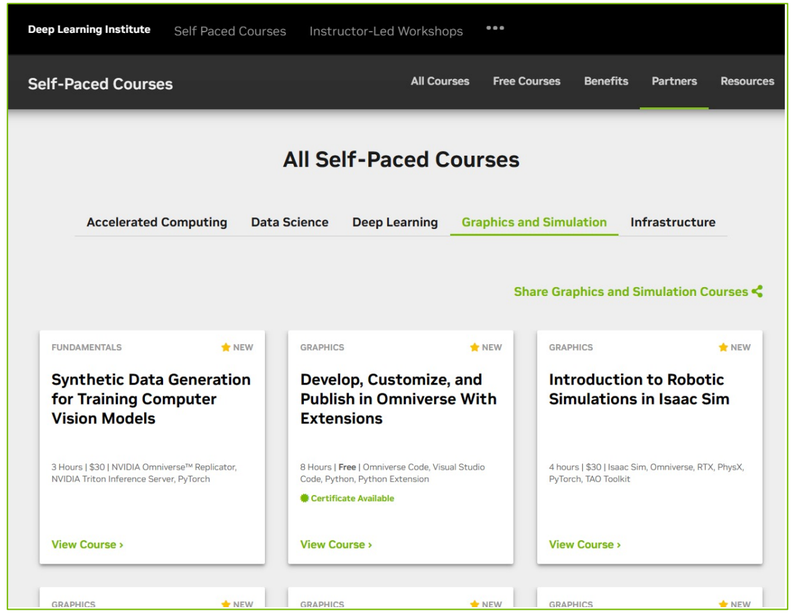
Want to learn more?

Omniverse Document



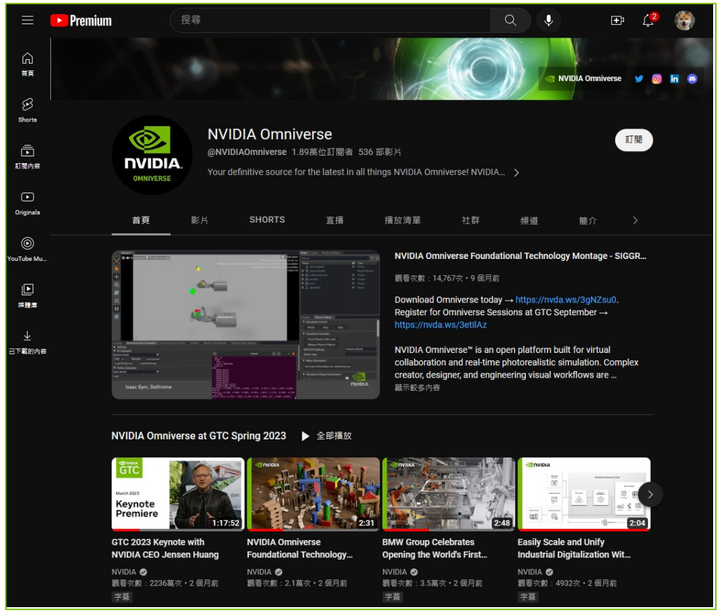
https://docs.omniverse.nvidia.com/

NVIDIA Deep Learning Institute



https://www.nvidia.com/en-us/training/online/

Omniverse YouTube



https://www.youtube.com/@NVIDIAOmniverse

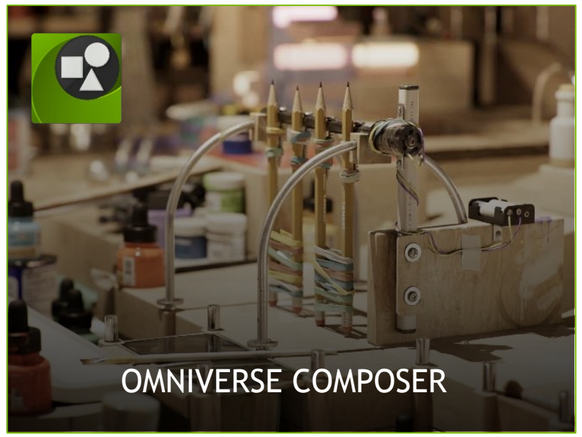

NVIDIA

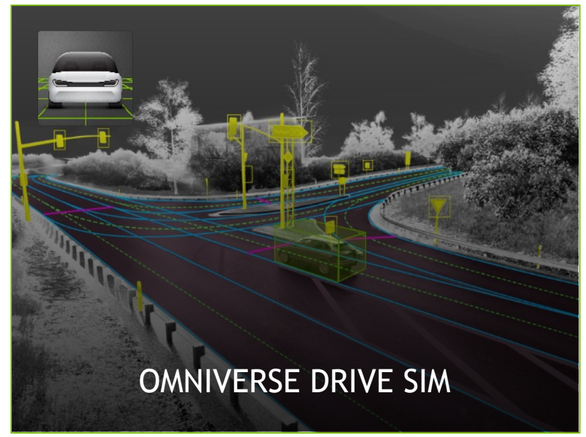
Kit Solutions

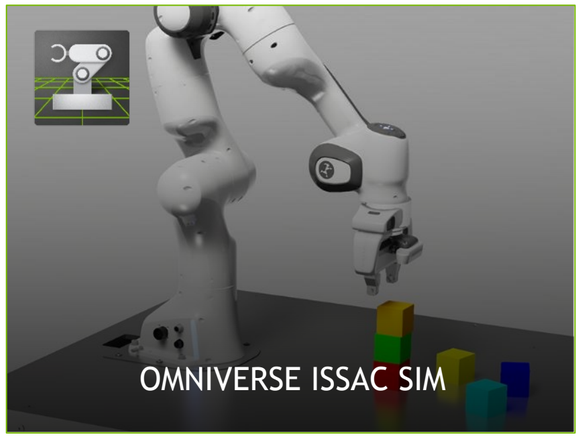
Presenter, Composer, and Code

Start Workflows with Omniverse Foundation Applications

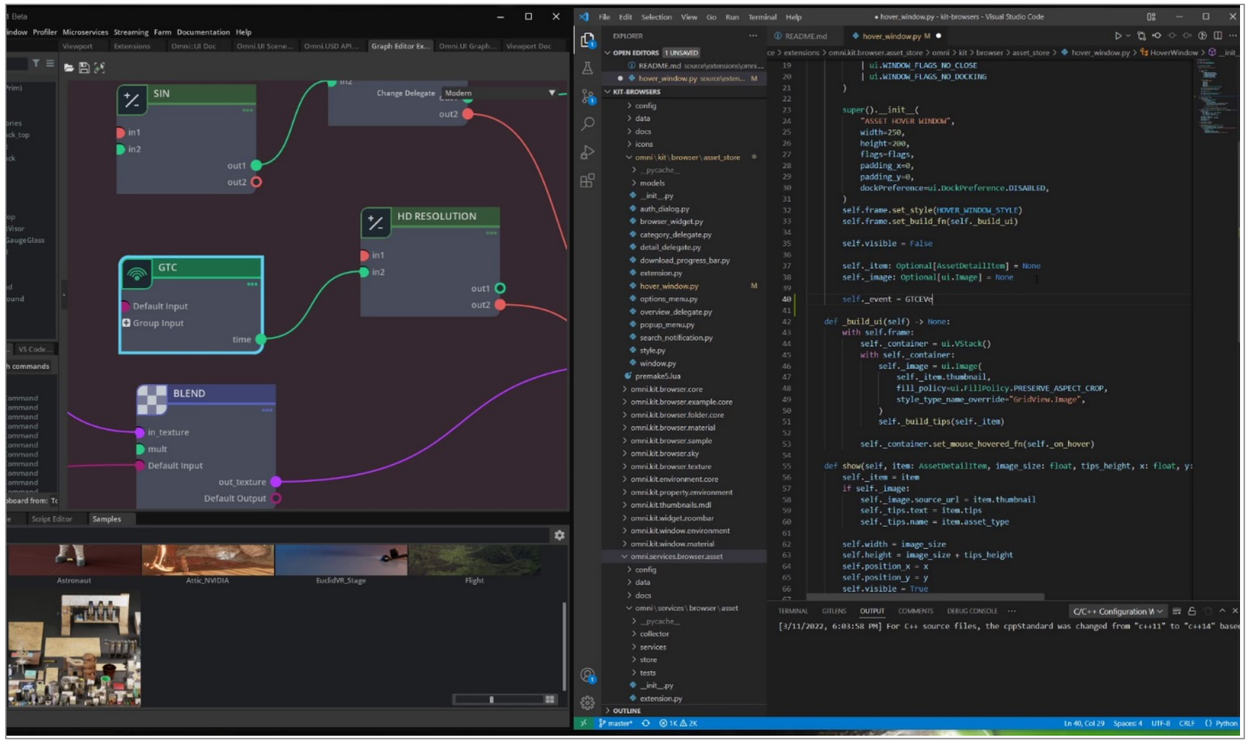
Fully Customizable and Extensible “Template” Applications







More APPs Build on Kit



OMNIVERSE CODE

- Simple to learn and use – easy user interface, interactive documentation, sample templates, and ‘Hello World’ exercises
- Helps developers and power users achieve maximum output with minimal code – free to use any of the 300+ NVIDIA-built Omniverse Extensions in their projects, so no need to start from scratch
- Easily package and publish to a private or public registry
- Includes Omniverse Kit runtime

Omniverse Code

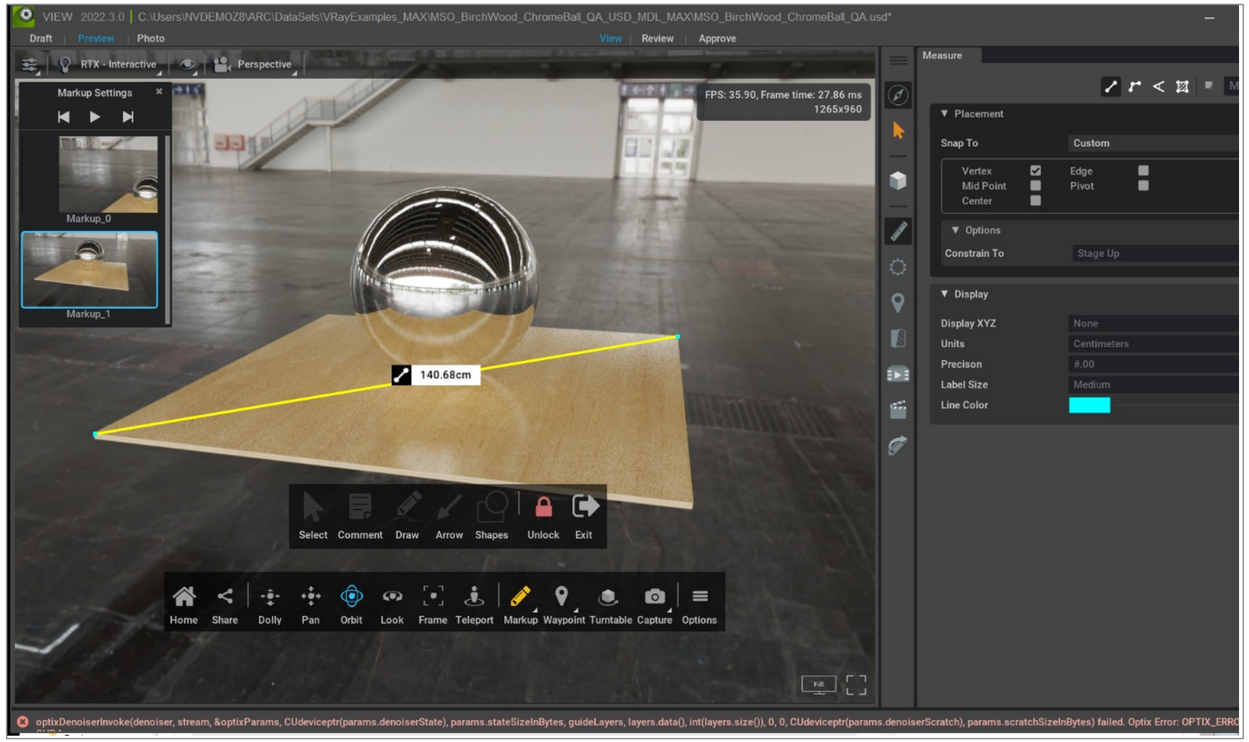
Integrated Development Environment (IDE) to build Omniverse extensions, apps, microservices

OMNIVERSE COMPOSER

- Simplify world building with intuitive layout tools and physics
- Breathtaking photorealism with physically-accurate materials, and real time ray and path traced rendering
- Advanced simulation capabilities with NVIDIA PhysX 5, Flow, and Blast integration

Omniverse USD Composer (formerly Create)

Advanced USD Scene Composition, Lighting, Rendering



OMNIVERSE
PRESENTER

- Simple to use tools for project reviews including camera waypoints, annotations, measure, and markup
- Ability to make minor environment or material iterations and edits to present multiple options
- Quick toggling between real-time ray traced and ultra-high-fidelity path-traced mode allows teams to visualize interactively

Omniverse USD Presenter (formerly View)

Immersive, true-to-reality visualization for reviews & approvals