Use Case Template

[Use Case ID] - TitleNVIDIA Logo.

Placed at the bottom left of the title page.

Documentation Control

| Item | Description |
| --- | --- |
| Title | <doc title> |
| Author(s) | <author> |
| Revision | <date> |
| State | <Development, Baselined, Released for Approval, Approved> |

Approvers

| Date | Name | Notes |
| --- | --- | --- |
| <month dd, yyyy> | <Full Name> | <Brief description.> |
|  |  |  |

Reviewers

| Date | Name | Notes |
| --- | --- | --- |
| <month dd, yyyy> | <Full Name> | <Brief description.> |
|  |  |  |

Revision History

| Date | Author | Summary of Change |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

## Use Case PIC

* Who is the Use Case PIC for this project? <DEFINE>

## Capability Subject Matter Experts

* Who are the SME’s that can help define the capabilities for the defined Use Case? <DEFINE>

## Definition: What are Use Cases?

A use case leverages (aggregated) assets and runtimes in order to address a particular need or problem by simulation, aggregation, visualization etc. Multiple assets and simulators are often combined to realize a singular use case. **The use case should provide immediate context for the capabilities that 3D assets require, in order to satisfy the goals of the use case.** Creating a well-defined use case also informs the identification of the types of assets (e.g. environments, robots, vehicles, props, etc.) that are needed to build scenarios and how the capabilities apply to each.

The order of operations here builds on each step and helps define the asset needs as a whole.

1. **Use Case** defined
2. **Simulation Runtimes** defined (includes specific features)
3. **3D Assets Types** identified
4. **Capabilities** on those 3D Assets to support the defined runtimes for the defined use case

## Use Case Description

**Use Case Goal:** <Define the Use Case as specifically as possible. What is the goal of this project for the customer and ecosystem?>

**Runtime Simulators Needed:** <Define which simulation runtime components are necessary to achieve the Use Case goal. What has to happen in the simulation to achieve this goal (e.g. does the runtime need to get RTX Sensor returns)?>

**Process:** <Diagram the process and personas involved with each system of the use case. How does the use case start, and which data flows between the systems involved. Indicate the delineation of each phase of multi phase processes. >

## Use Case Asset Capability Matrix

< For each Asset Type, define the capabilities that are required by the assets to fulfill the requirements of the systems / runtimes that have been identified above. Choose the code of the capability from Appendix A, or note ***gap*** whenever the required capability is not defined yet. If specific levels of detail are required (for example, lights need to be in realistic values, write that into the cells.>

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Capability / Asset Type** | **Runtime Simulator Need 1** | **Runtime Simulator Need 2** | **Runtime Simulator Need 3** | **Runtime Simulator Need 4** | **Runtime Simulator Need 5** | **Etc.** | **Etc.** |
| **<Use Case Phase>** | | | | | | | |
| Asset Type 1 |  |  |  |  |  |  |  |
| Asset Type 2 |  |  |  |  |  |  |  |
| Asset Type 3 |  |  |  |  |  |  |  |
| Asset Type 4 |  |  |  |  |  |  |  |
| Asset Type 5 |  |  |  |  |  |  |  |

## Appendix A

Below is a list of identified capabilities along with IDs that can be used in the asset matrix. Please note that simply because a capability is listed here, it does **not** mean that the capability has been created or validated.

## Capabilities

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Capability / Job to be done** | **Scene Description** | **Runtimes / Simulators** | **Description** |
| VF-1 | **Visual Fidelity** |  |  | Creation of images from virtual scenes with a renderer. |
| VF-1.1 | Visualization, Spatial Planning | Geometry | Visual Renderers | * Polygonal Meshes or Subdivision Surfaces * Or optionally Solids Boundary Representations (B-Reps) |
| VF-1.2 | Visualization, Marketing, ML Training (Perception, Physical) | Appearances (~Materials) | Visual Renderers | * Physically plausible materials and textures.   In the context of SimReady targeted use cases, this means that materials should be supplied in portable material standards which allow pathtracers to perform physically plausible surface appearance calculation when rendering. |
| VF-1.3 | Visualization, Marketing, Factory Planning, ML Training (Perception, Physical) | Lights | Visual Renderers | * Physically plausible Light emitters   In the context of SimReady targeted use cases, this means that lighting within assets should be supplied such that it allows pathtracers to perform physically plausible illumination when rendering (eg. IES and alternatives). |
| VF-1.4 | Visualization, Marketing, Factory Planning, ML Training (Perception, Physical) | Cameras | Visual Renderers | * Physical camera parameters and attributes |
| MH-1 | **Component Model Hierarchy** |  |  |  |
| MH-1.1 | BOM & Product Information, Parts and Assemblies | Product / Part & Assembly Hierarchy |  |  |
| MH-1.2 | Component Assembly | Annotated Geometry or Coordinate Systems / Frames | Assembly / mating tools,   Kinematic Emulation / Simulation | * Attachment frames for other components in an assembly (gripper connections on robots, tools, sensors, fasteners, holes etc) |
| GT-1 | **Ground Truth Training** |  |  | Ground truth data provides baseline identification of what a digital asset represents within a runtime and allows it to be directly discovered and introspective. It plays a role in enhancing simulations by providing accurate and reliable data that serves as a benchmark for evaluating and refining simulation and ML models. |
| GT-1.1 | ML Training, SDG | Semantic Labels | Metadata Introspection,  AOV Renderers | * Embedded metadata to describe the asset, its properties and characteristics * Provides runtimes with information to interpret what it is “seeing” to aid with Synthetic Data Generation |
| GT-1.2 | ML Training, SDG | Dense Captions | Metadata Introspection | * Human readable text metadata to more fully describe an object beyond Semantic Labeling. |
| *GT-1.3* | *PMI - Product Manufacturing Information* | *Product Datum* |  | * Product specifications, tolerances, finishes and information that is necessary for manufacture of a real-world object |
| NS-1 | **Sensors** |  | Non-Visual Renderers / Simulators |  |
| NS-1.1 | Sensors perceiving Non-Visual Scene Description | Non-Visual Sensors | Non-Visual Renderers / Simulators | * Non Visual sensors models such as Lidar or Radar Scanners |
| NS-1.2 | Objects perceived by Non-Visual Sensors | (Non-visual) Materials | Non-Visual Renderers / Simulators | * Non Visual Surface Material descriptions to aid non visual Sensor Simulation (painted vs coated, paint type etc) |
| IOT-1 | **IOT** |  |  |  |
| IOT-1.1 | IoT Metadata connectivity | IoT driven attributes / metadata | Metadata Introspection / Custom Applications |  |
| RB-1 | **Rigid Bodies** |  |  |  |
| RB-1.1 | Collide with other objects in the simulated worlds | Collision Geometry & Physical Materials | Collision Simulation (in physics simulation) | * Geometry representing the physical shape of an object * Material data describing the physical properties of an objects material, such as friction |
| RB-1.2 | Physics Simulation | Mass | Rigid Body Physics Simulation | * The mass and inertia tensor of an Object |
| KS-1 | **Kinematics** |  |  |  |
| KS-1.1 | Playback of Kinematic Motion / Animation / Time & Motion Studies | Transform Hierarchies and Time Series data (animation) | Visualization, Factory Planning / Aggregation, Clash Detection | * Incorporation of pre-generated animation and timesamples, created in other applications as a playback mechanism for machines like robots, vehicles & machinery.. |
| KS-1.2 | Kinematic Poses and Movement ( velocity irrelevant), Clash Detection | Kinematics - Joints and Constraints | Forward/Inverse Interactive Manipulators, Kinematics, Animation and Manipulation Runtimes | * Interactive manipulation of objects kinematically, without simulation requirements, but potentially affecting the simulated world. * Joint definitions (min/max) and constraints of articulated elements |
| KS-1.3 | Actuation - Simulation of Kinematic Motion  (Sim2Real) | Forces and Kinematics -, Drive Definitions  (requires RB-1.2) | Physics Simulators | * Physical attributes required for full simulation that dictate how objects like robots and machinery behave:   + Describes forces like acceleration, friction, drives/actuators, damping and stiffness |
| KS-1.4 | Motion Planning - Kinematics for Motion Generation | Machine Data (MADA),   Spaces, Collision objects, Drive Definitions | Motion Planning and Kinematic Articulation Solvers | * Machine specific data which is required to generate accurate motion for a machine (robot) (+ C-Space, Robot Configuration Files) |
| RB-1 | **Robotics** |  |  |  |
| ROB-1.1 |  |  |  | Data which is specific to robotics - ie new Schemas |
| SB-1 | **Soft Bodies** |  |  |  |
| SB-1.1 | Deformable Object Simulation | Soft Bodies | Soft Body Simulator | * Description of an deformable object, such as elasticity, stiffness / damping, mass distribution |
| SB-1.2 | Suction Cup Simulation |  |  |  |
| SB-1.3 | Wire and Cable Poses and Movement | Parametric Description of Wires and cables | Forward/Inverse Interactive Manipulators, Kinematics, Animation and Manipulation Runtimes |  |
| DL-1 | **Runtime Behavior Patterns / Domain Logic** |  |  |  |
| DL-1.1 | Environmental Lighting Control | Environmental Lighting |  | * Identification and definition of environmental controls needed to provide for different lighting conditions |
| DL-1.2 | Walkable Surfaces ? |  |  |  |
| DL-1.3 | Time of Day Emulation | Time of Day State Annotations | World Simulator | * Description of logic that dictates how time passes in the simulation and its impact on other runtime actors |
| DL-1.4 | Traffic Signal State Visualization | State Machine | World Simulator | * Definition of logic that drives the behavior of an illuminated signal to control traffic patterns |
| DL-1.5 | Human and Animal behaviors | Behaviors, Motion Clip Library | World Simulator | * Definition of logic that drives the behavior of animals and humans and their interactions within other elements in a simulation |
| DL-1.6 | Conveyor System Control | Conveying Behaviors | Conveyor Belt Controller | * Definition of controls needed to animate a converyor belt and emit associated forces into the simulated world |
| VW-1 | **Virtual World Actors** |  |  |  |
| VW-1.1 | Humans and Animals - Posing and movement | Skeletons, Blendshapes | Animation Runtime | * Definition of skeletons and blendshapes * Motion can be applied via animation of the skeleton and the blendshapes |
| VW-1.2 | Vehicle actors | Vehicle Component Annotations, Transforms + Pivots | Vehicle Physics Rig / Vehicle Simulator | * Definition of attributes within multi-wheel vehicles to allow for creation of rigs, to support behaviours   + Suspension, articulation of doors, lighting |
| VW-1.3 | Robot actors | Hierarchy and construction | Robotic Physics Rig / simulation of grippers | * Definition of attributes within a robotic mechanism (robotic arms, AMRs, humanoids) to allow for creation of rigs to support industrial simulation behaviors   + Grippers |

### 