PACELAB WEAVR

WEAVR - USER MANUAL

Version: 2.0.0

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

Register and Revision History

Rev.	Date	Sections	Description
0	30.05.2018	All sections	First Issue
1	16.01.2019	All sections	Update compliance with editor v1.0.7
2	28.01.2019	All sections	Update compliance with editor v1.0.9
3	09.04.2019	All sections	Update compliance with editor v1.1
4	28.05.2019	All sections	Update compliance with editor v1.2.0
5	26.08.2019	All sections	Update compliance with editor v1.2.3
6	19.04.2020	All sections	Update compliance with editor v2.0.0
7	26.09.2020	All sections	Update compliance with editor v2.0.0
8	02.09.2020	All Sections	Document organization

Acronyms and Definitions

Acronym	Definition
GUI	Graphic User Interface
OpS	Operation Support
Pacelab WEAVR	Product of TXT Group
VR	Virtual Reality
SW	Software
VT	Virtual Training
TTS	Text to Speech

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

Table of Contents

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

Table of Figures

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com |website: www.txtgroup.com | Tel: +49 30 293620

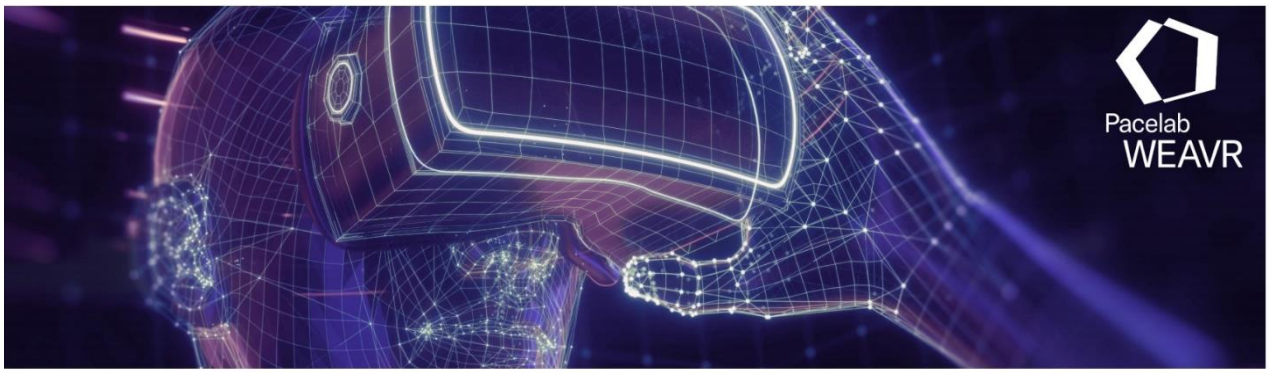
PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

1. INTRODUCTION



Pacelab WEAVR is a powerful software toolbox to streamline the design and development of virtual training systems. It provides an integrated Editor platform, which supports the entire production process from instructional design to final delivery.

Pacelab WEAVR enables you to create a wide array of training systems from the same set of training data, creating a seamless and consistent learning experience from a desktop procedural training to a full mission practice. Its template-based visual approach requires little to no programming or scripting skills allowing subject matter experts to create tasks efficiently and without involving 3D Editors or software engineers.

This innovative toolbox is designed to promote a high level of reuse seamlessly integrating existing components and simulation modules, as well as deploying the same training content across a variety of systems including desktop PC, mobile, and VR devices.

1.1 WEAVR Environment

Pacelab WEAVR offers a unique set of features and modules enabling companies to create, manage, and implement a wide array of training systems from the same set of training data, as well as to support field operations from a desktop procedural training to a full mission practice.

The solution is designed for subject matter experts to easily create content; for students and field operators to learn and get support while carrying out operations collaboratively; for instructors and managers to monitor and support these activities.

Pacelab WEAVR comprises three main modules:

- **Creator**
- **Manager**
- **Player**

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

WEAVR **Creator**

Simplified application development

WEAVR **Manager**

Comprehensive management portal

WEAVR **Player**

Centralized application execution

1.1.1 WEAVR Creator

Built on top of Unity 3D, WEAVR Creator is a WYSIWYG authoring tool providing wizards and libraries for the creation of procedural-based content enhanced by virtual and mixed reality. The Creator is designed to require little to no programming or scripting skills. It enables the definition of behaviors and animations of 3D elements empowered by customizable assets, implementing:

- Basic tools with standard behavior and animations, e.g. buttons and levers.
- Cameras and relative movement scripts.
- A flow-chart editor that enables users to model procedures by defining steps, groups of steps, navigation flows, animations, conditional navigations, and any other useful actions.

The optional Developer Simulation Hub module is an SDK that allows integrating with existing real-time simulations and related visual streams.

1.1.2 WEAVR Manager

The Manager module collects and distributes content and basic functionalities to instructors and managers. It also enables the handling of users, groups, and job assignments, as well as controls real-time monitoring.
The optional modules provide:

- Periodic reports summarizing activities that also use xAPI logs.
- Remote collaboration allowing more parties to share the same view (immersive or augmented) and exchange information.

1.1.3 WEAVR Player

The Player enables to access, select, and execute the available procedures in different modalities. WEAVR Player is available across various systems including desktop PC, mobile and VR devices, such as HTC VIVE, Oculus, and Microsoft HoloLens. The Player needs to connect to WEAVR Manager through a login process to update the state of its existing procedures and/or download new procedures. The Player also allows sharing a procedure between multiple students.

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

1.2 WEAVR Procedures

In WEAVR nomenclature, a procedure can be a Virtual Training (VT) lesson, an Operations Support (OpS) helper, a Computer-Based Training (CBT), or even an automatic system to execute logic. Procedures are created by the WEAVR Creator and executed by the WEAVR Player. Procedures can have a sequential execution, as well as parallel one, and its execution can be shared by multiple users.
It can also alter its execution based on user input and/or any other external input (e.g. Simulation Hub). A procedure is designed to be self-contained, which means it can store all its assets with it, but it also can get external data during the runtime.
A standard procedure has the following structure:

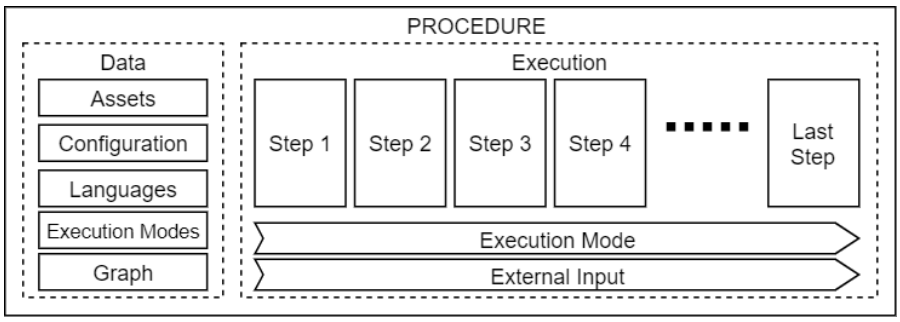


Figure 1.Procedure Structure

- **Assets**: contains assets that the procedure needs for its execution (e.g. audio files, images, etc.).
- **Configuration**: configuration that describes how and where this procedure can be executed along with some additional data used by the WEAVR Player.
- **Languages**: contains languages that this procedure supports (these languages will be converted to speech). You can define languages when creating a procedure (in Procedure Wizard) or while configuring procedure steps (in Procedure Inspector)
- **Execution Modes**: define the type of mode (Automatic, Guided, or Feedback) and how it should be executed by the procedure.
- **Graph**: contains the steps and navigation among steps.

1.2.1 Procedure Configurations

Since the procedures can be anything from VT lessons to automatic execution systems, the WEAVR Player needs a way to know how to execute, and if it can execute the procedure. The configurations are used to determine the type of the procedure. These configurations can be customized, and new ones can be created. The standard set of configurations are:

- **Virtual Training** configuration: used for VT lessons. Usually allows user navigation and interaction with virtual objects.
- **Operations Support** configuration: used as supportive media to accomplish various field operations (e.g. technical assistance for maintenance workers). Usually does not allow user navigation in the virtual worlds and allows the user to interact only with User Interface (UI) elements.

WEAVR Creator has a quite modular system for the procedure configurations allowing an expert user to extend existing configurations or create new ones without a line of code.

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

1.2.2 Procedure Execution Modes

The Execution Modes are at the core of a procedure execution and, together with a procedure configuration, define the procedure type. An execution mode directs he WEAVR Player which steps to execute, what to execute in every step, and how to execute it. Like configurations, execution modes are extendable, or even new ones can be created by an expert user without any coding knowledge.
The standard set of execution modes is the following: Automatic, Guided, Feedback.

1.2.2.1 *Automatic Execution Mode*

This mode is designed to be handled and executed entirely by the WEAVR Player with the minimal user interaction (move back and forth between steps). It is usually used to show the user how the entire procedure should be performed (e.g. for a student to learn a lesson). It involves predetermined camera movements, an automatic object interaction, automatically triggered animations, informative audios, videos, images and texts, as well as object highlighting. The only interactions allowed by the user are to navigate between procedure steps, either to skip or replay them.
This mode is not always available for execution; for example, in VR environments, it is rarely used because there are features not suited for VR (e.g. automatic camera movement can cause nausea)

1.2.2.2 *Guided Execution Mode*

This mode is designed to help the user perform and complete a procedure. It allows the user to freely navigate the virtual world and to interact with virtual objects. It has various helpers for the user ranging from guiding informative audios (instructor voice), billboards, images, videos, to object highlighting, animations, world navigation guidance, etc. While the user can navigate the world, the interaction with objects is limited by the step requirements; for example, if the step requires the user to press the button X, only button X is interactive during that step.
By default, unlike Automatic mode, this mode does not allow navigating steps (mainly because of physics states), however, this logic can be activated if it is needed.

1.2.2.3 *Feedback Execution Mode*

This mode is designed to test the user ability to perform the procedure (as an exam for a maintenance student). This mode does not have any helpers to guide the user, thus the user must figure out himself what to do. The only feedback for a user is whether a step has been completed or not. The user can navigate freely throughout the entire virtual environment and interact with objects. It is up to the procedure designer to put constraints on which objects can be interacted with during each step.
By default, this mode allows a user to request help, which invokes some helpers defined for other execution modes (e.g. Automatic or Guided). This is useful when the user is blocked and cannot proceed. However, it also can be used to penalize the user and lower its final score. This functionality can be easily disabled if not needed.
Like Guided mode, the Feedback mode, by default, does not allow navigating steps , but it can be enabled if needed.

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com |website: www.txtgroup.com | Tel: +49 30 293620

2. WEAVR CREATOR

The Creator is built on-top of Unity editor and is designed to require little to no programming or scripting skills.
It enables the definition of behaviors and animations of GameObjctects empowered by customizable assets, implementing:

1. Basic tools with standard behavior and animations, e.g. buttons and levers.
2. Cameras and relative movement scripts.
3. A flow-chart editor enables users to model procedures by defining steps, groups of steps, navigation flows, animations, conditional navigations, and any other useful actions.

2.1 Import WEAVR

You can import WEAVR in Unity by doing one of the following:

- Option 1. In the *Project window, right-click Assets, point to Import Package, and then click Custom Package. In the Import Package dialog box, navigate to the WEAVR package* (file .*unitypackage*), and then click Open.
- Option 2. In the Project window, move the WEAVR package (file .*unitypackage*) from the PC to the *Assets* folder.

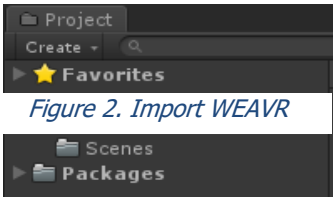


Figure 2. Import WEAVR

Wait until the installation is completed successfully.

> WARNING
> To work properly, the WEAVR folder should be included in the Assets folder of the Unity project.
> This operation should be done for every project.

2.1.1 Welcome Screen

When you open a new project with the WEAVR package for the first time, the Welcome WEAVR window appears. It allows you to easily identify and activate the needed extensions (Figure 3) and to set up a project suitable for WEAVR features.

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

Figure 3. Welcome Screen

The same extensions (Virtual Reality, Augmented Reality, and Multiplayer) can be activated later in the Manage Extensions window, explained in Load WEAVR License.

> NOTE
> The available extensions depend on the current project platform. If a non-compliant extension is selected in the Welcome window, the extension is activated when changing the project platform with the compliant one (e.i. if Augmented reality is enabled while in PC platform it is not immediately imported, it is activated when changing to Android or iOS platform).

2.1.2 WEAVR default assets

The WEAVR folder contains all the scripts and components necessary for the tool to work properly (e.g. objects interactions, behaviors, actions, etc.).

> WARNING
> It is recommended not to modify the content of the WEAVR folder.

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

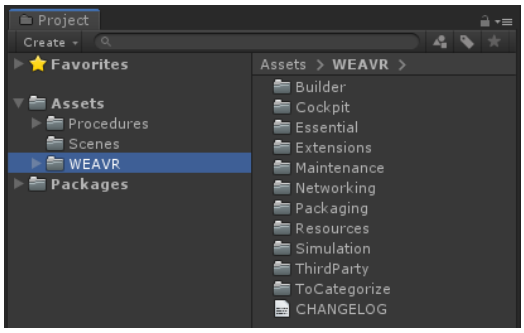


Figure 4. Project window, WEAVR folder

2.2 WEAVR Menu

After installing the WEAVR package successfully, the new WEAVR tab appears on the toolbar. (Figure 5. WEAVR menu).

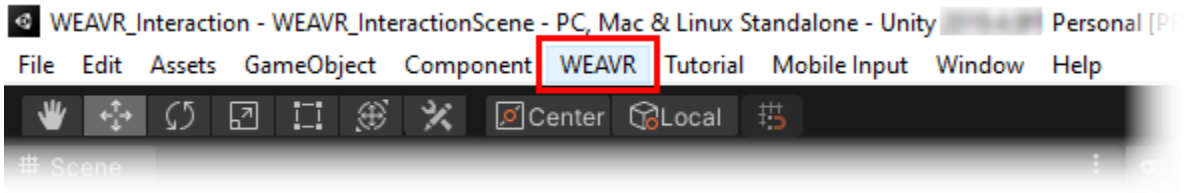


Figure 5. WEAVR menu

Using the WEAVR tab, you can manage the following:

Procedures

- Create Procedure: allows creating a procedure using either a VR or OpS templates.
- Catalogues: allows setting up defaults for each procedure element.
- Inspector: opens Inspector window.
- Procedure Editor: opens Procedure Editor window.

Utilities

- Camera Manager: allows creating Camera Poses for OpS procedures.
- Copy Components: allows automatic copying of components between two objects.

Diagnostics

- Variables: shows the variables status while in play.

Builder: *internal purposes features*

Licensing

- Information: information about the WEAVR license.
- Load Editor License: upload Editor license.
- Remove License: removes the Editor license.

Setup

- Manage Extension: allows managing extensions.
- Settings: allows configuring Editor and Player.
- Setup Scene: allows setting up the Unity scene for WEAVR procedures.
- Welcome Screen: helps setting up the project as required.

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

2.3 Load WEAVR License

To load the WEAVR License:

1. *Go to WEAVR > Licensing > Load Editor License*.
2. In the Load WEAVR Editor License File dialog box, navigate to the license file (plweavr.lic), and then click Open.

The license is automatically located in the correct folder:
C:\ProgramData\Pacelab\WEAVR

2.4 Manage Extensions

> ATTENTION
> Before enabling the required extensions, make sure the project is in the correct platform in *File > Build Settings.*

To use extensions in the project, turn them on in the WEAVR Extensions Manager dialog box.
To open the window, go to WEAVR > Setup > Manage Extension.

- Virtual Reality: toolkit for projects involving a Virtual Reality tool (using SteamVR).
- Augmented Reality: toolkit for augmented reality applications
- MRTK: toolkit for projects involving a Microsoft Hololens tool.
- Network: toolkit for multiplayer

Switch ON or OFF the extensions to enable or disable them.

> NOTE
> Only the extensions compliant with the current platform of the project are visualized on the list (e.i. Android platform does not visualize Virtual Reality extension). The enabled extensions are proper of the current platform: changing the project platform changes also the enabled extensions.

If something went wrong while managing the extensions, click on "Repair Extension" button repair the packages status.

> ATTENTION
> If errors in console persist, restart Unity to let the project compile properly.

Figure 6 shows an example available extension in WEAVR Extension Manager for Windows standalone platform.

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

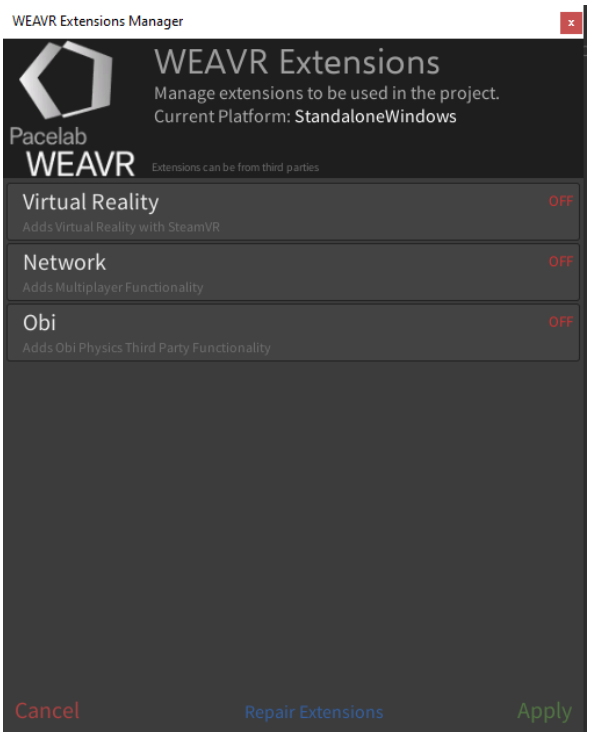


Figure 6. WEAVR Extension Manager for Windows Standalone Platform

2.4.1 Virtual Training

To be able to create a virtual training, configure the SteamVR settings in the Assets of the project. In the Project window, go to *Assets* > *WEAVR* > *ThirdParty* > *VIVE* > *SteamVR* > *Resources* > *SteamVR_Settings*, and then select the "Auto Enable VR" check box (Figure 7).

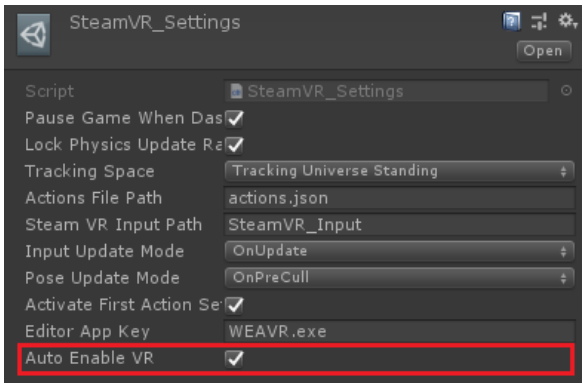


Figure 7. Steam VR Settings

To set up the WEAVR project to support VR:

- Go to WEAVR > Setup > Manage Extension, and then turn Virtual Reality on.

Wait for the project to load the new configuration.

- When the SteamVR window appears:click the recommended configurations for each segment, and then click *Accept All.*

Note: the SteamVR window may not pop up as soon as the project is configured for the VR extension.

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

Figure 8. Steam VR window

2.4.2 Augmented Reality

When the augmented reality extension is enabled, the following unity packages are automatically installed in the project:

- AR Foundation (version 4.1.x)
- AR Core (version 4.1.x)
- AR Kit (version 4.1.x)

Note: If the AR packages are not installed automatically, you can install them manually in Package Manager.

2.4.3 Mixed Reality

To start using Mixed Reality components:

1. In the WEAVR Extensions Manager dialog box, turn the MRTK extension on. ATTENTION: disable Virtual Reality and Network extensions.
2. go to Mixed Reality Toolkit > Add to Scene and Configure, to configure the scene
3. Change the project platform with: File > Build Settings, switch to "Universal Windows Platform".
4. Set the Mixed Reality Toolkit (in *Hierarchy > MixedRealityToolkit)* tag to "EditOnly".
5. Set the Mixed Reality Playspace (in *Hierarchy > MixedRealityPlayespace)* tag to "EditOnly".

2.5 WEAVR Settings Window

You can configure Editor and Player settings using the Settings dialog box. It can be accessed from: *WEAVR > Setup > Settings*

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

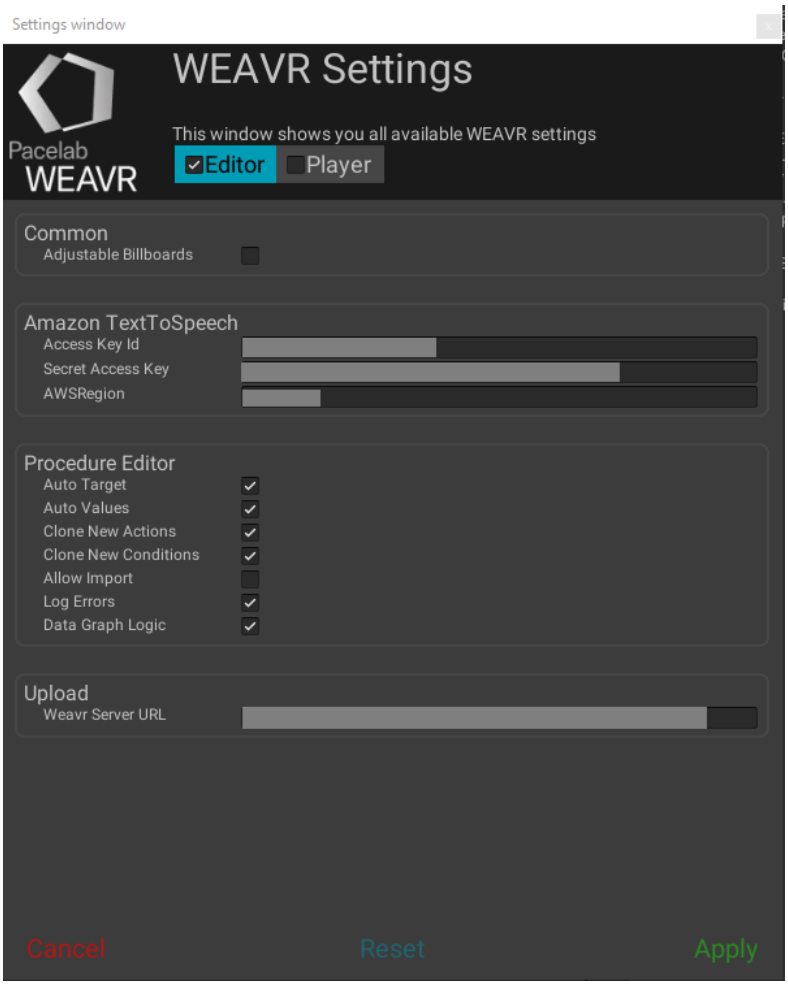


Figure 9. WEAVR Settings – Editor

Amazon TextToSpeech: credentials to use the Amazon TTS during the WEAVR procedure. To use Amazon TTS, instead of Windows TTS, make sure the internet connection is enabled.

- Access Key Id: first part of the user id.
- Secret Access Key: second part of the user id.
- AWSRegion: the hand point region.

Procedure Editor: general settings for the procedure editor.

- Auto Target: when creating new actions or conditions, the target is set from previous actions/conditions.
- Auto Values: when creating new actions or conditions, their values are set automatically from previous actions/conditions.
- Clone New Actions: the newly created action copies all the values from previous action of the same type.
- Clone New Conditions: the newly created action copies all the values from previous condition of the same type.
- Allow Import: whether to allow or not the import of legacy procedures.
- Adjustable Billboards: whether to adjust a billboard start point (used for correcting purposes).

Upload: URL to upload procedures to the server.

- WEAVR Server URL: address for WEAVR server upload API's.

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

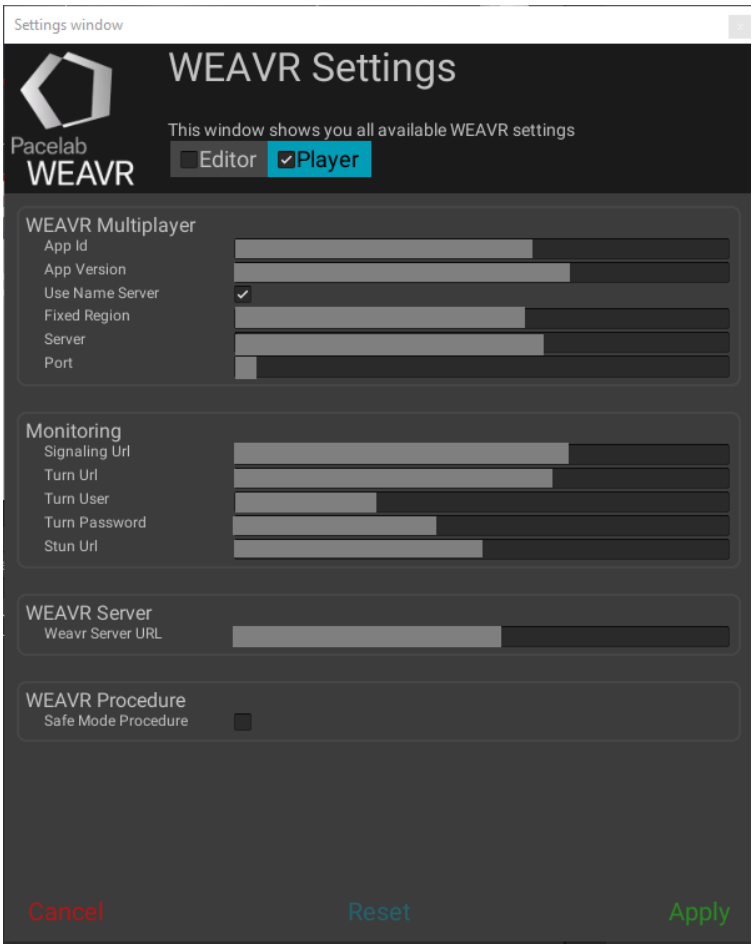


Figure 10. WEAVR Settings – Player

Monitoring: information to allow the instructor to monitor the user improvements.

WEAVR Server: URL address to download procedures.

- WEAVR Server URL: the URL of the server

2.6 Scene Setup

Before setting up a Unity scene with the WEAVR components, you should activate all the extensions necessary for the project. The available extensions can be managed from Manage Extensions window.
To setup the necessary WEAVR environment in a Unity scene, go to WEAVR > Setup > Setup Scene

1. In the WEAVR Setup dialog box, turn on the needed options:

- Essential: to set the major objects of the WEAVR in the hierarchy.
- Maintenance: to import the main components for the object Interactions and Behaviors.
- Simulation: to import components related to simulation.
- Cockpit: to import components related to cockpit.

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

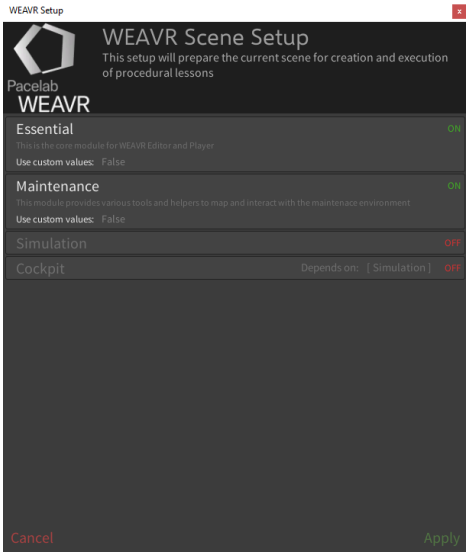


Figure 11. WEAVR Scene Setup

2. Select Apply

NOTE
According to the Operation Mode selected in the WEAVR Setup dialog box, specific GameObjects appear in the Hierarchy window of the scene. These objects contain essential components for the WEAVR project.

2.6.1 WEAVR scene (Hierarchy Window)

The Hierarchy window contains a list of every GameObject in the current Scene. Some of these are direct instances of Asset files (like 3D models), while others are instances of Prefabs. As objects are added or removed in the Scene window, they appear and disappear from the Hierarchy window.

WEAVR adds components that are necessary to set up and manage a procedure to the Hierarchy window, by default. This is due to the operation mode that you select in the WEAVR Setup dialog box.
This section lists the basic WEAVR scene settings and WEAVR prefabs.
You can manage the procedure, the player, and canvas related to the WEAVR with the help of GameObjects in the Hierarchy window. The components of these objects contain advanced settings that are referenced by default. For the accurate execution of procedures, you should not modify the settings, except for the settings explained in WEAVR basic settings. The following sections list the basic settings that can be customized, and the default Prefab objects that can be used in the scene.

Adding assets: Importing the 3D model
To use the 3D model in the scene, import the Unity package file first.

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

In the Project window, right-click the desired folder, point to Import Package, and then click Custom Package.

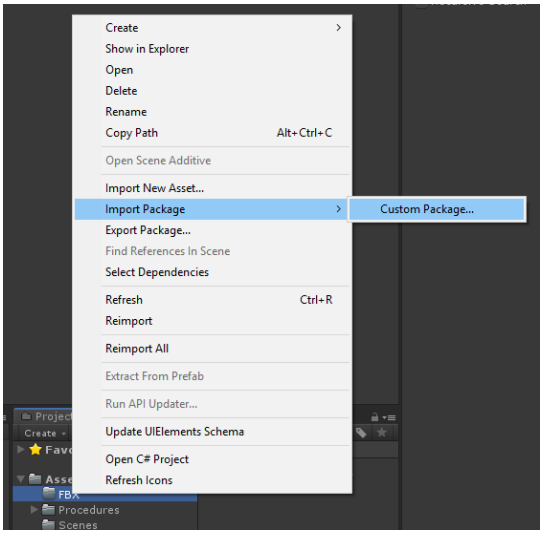


Figure 12. Flow for importing the model

2.7 Develop a Project

After configuring the project to support the WEAVR tools, you can develop the project further. The flow to develop a project is shown in Figure 13.

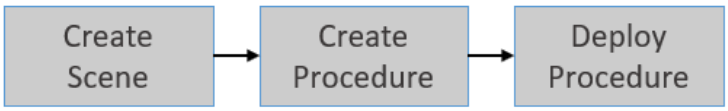


Figure 13. Flow to develop a project

1. Create the scene: First, add assets to the project and add objects to the scene,then add components to the objects (WEAVR interactions or Unity components). An advanced customization of the scene is possible (matter of advanced sections).

2. Create the procedure: First, create and connect steps, , add actions, exit conditions, and exit actions to each step, then test the procedure.

3. Deploy the procedure: build the application.

2.8 WEAVR Basic Settings

This section explains the basic settings that are related to the scene. These settings are controlled by components of GameObjects generated with the Scene Setup.

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

2.8.1 Billboard Manager

Billboard Manager (UI canvas) helps orient players to the needed GameObject and guide them through the steps of the procedure by appearing in the correct position and moment. You can manage billboards in the procedure using the Hint actions To configure billboard settings, in the Hierarchy window, go to:
WEAVR > MANAGERS > Billboard Manager

The *Billboard Samples* is used as a generic object from which the specific billboard is created.
By default, one billboard sample is generated with the Setup:

- Default Billboard Sample: a simple billboard with a text message.

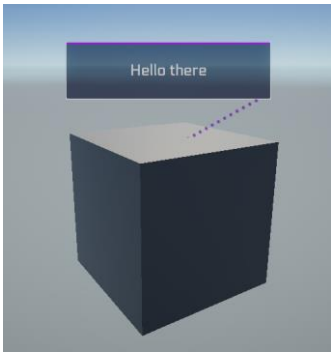


Figure 14. Default Billboard Sample

You can create more Billboard Samples if needed. To work in the procedure, the object needs:

- The Billboard component (Figure 16): allows managing the billboard as an object in the scene.
- Canvas as a child: contains the elements (texts, images, etc.) of the billboard. To set these elements in the procedure, each element needs to have the Billboard Element component (Figure 15).

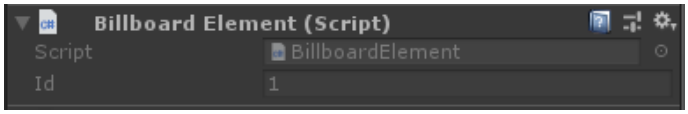


Figure 15. Billboard Element

The Billboard component allows managing the billboard as an object in the scene by customizing the following properties.

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

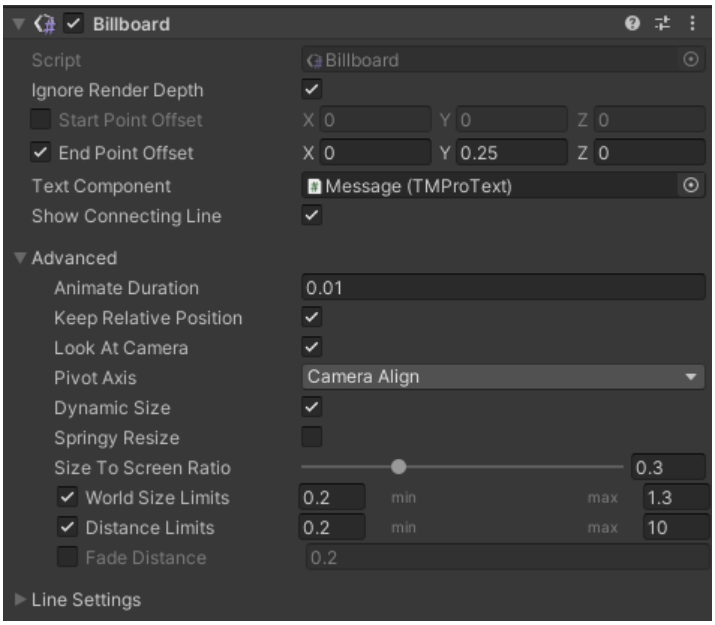


Figure 16. Billboard component

Properties

Property	Function
Ignore Render Depth	If enabled, the billboard is not occluded by other objects in the scene.
Start Point Offset	If enabled, the start position of the billboard line is set in respect to the transform of the focused object (this setting can be set in the Billboard actions).
End Point Offset	If enabled, the end position of the billboard line is set in respect to the transform of the focused object (this setting can be set in the Billboard action)
Text Component/Text Element	The GameObject containing the billboard text. If it is a TextMeshPro – Text (UI) it is referenced in Text Component, if it is a Text it is referenced in Text Element.
Show Connecting Line	If enabled, the line connecting the billboard and the object is shown.
Animate Duration	How long the billboard animates when it appears in the scene.
Keep Relative Position	If enabled, the billboard keeps the same relative position with respect to the focused object and does not rotate.
Look at Camera	If enabled, the billboard always faces the camera.
Pivot Axis:	Rotational axis of the billboard when facing the camera.
Camera Align	Billboard rotates with the camera view.
Up	Billboard rotates only on a vertical axis.
Free	Billboard rotates on three axes.
Dynamic Size	If enabled, the billboard size changes regarding the camera distance.
Springy Resize	If enabled, the billboard animates when resized.
Size to Screen Ratio	Percentage of the camera view that the billboard occupies. It is with respect to the screen width and restricted to the minimum and maximum world sizes.
Word Size Limits	Billboard size in the virtual world will be constrained between the min. and max. limits (in meters).
Distance Limits	Billboard will be visible to the camera if their distance is between the specified range (in meters).
Fade Distance	Distance from the Distance Limits property at which the billboard starts to fade.

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

NOTE
To customize Billboard settings, in the Project Inspector window, go to Add Action > Hints > Advanced > Show Billboard Expert.

The Billboard Manager component (the *Hierarchy window > WEAVR > MANAGERS > BillboardManager)* references one Billboard Sample; this will be the reference of the Show Simple Billboard action.

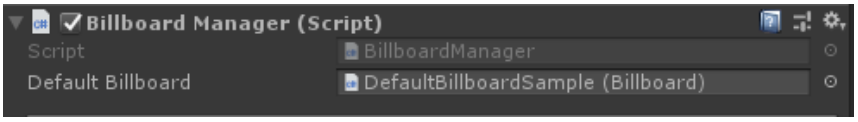


Figure 17. Billboard Manager component

2.8.2 Border Outliner

WARNING
This component does not have any effect on URP projects.

The Border Outliner component creates an outline of an object. It is used by the interaction system (e.g. when you hover over the object with the controller in VR), as well as the actions in procedures (e.g. Outline Object). This component should be attached to the main camera in one of the following locations in the Hierarchy window:

- *WEAVR > PlayerRIG > PlayerVR > SteamVRObjects > VRCamera*
- the *WEAVR > PlayerNonVR > WEAVRCamera*
- *WEAVR > FreeCamera*

The Border Outliner component allows creating various outlines for objects by customizing the following properties.

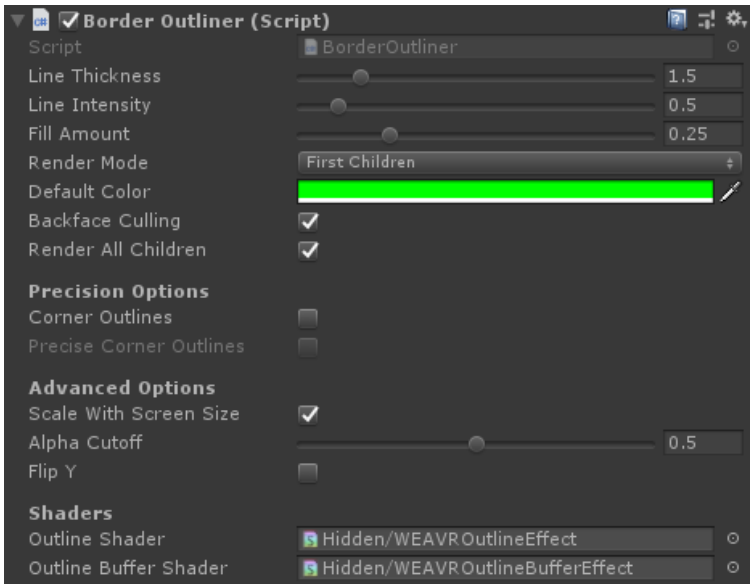


Figure 18. Border Outliner

Properties

Property	Function

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

Line Thickness	Controls the thickness of the outline.
Line Intensity	Controls the intensity of the outline.
Fill Amount	Amount of color to be applied to the object (in percentage).
Render Mode:	Controls which objects to outline. You can override this parameter by adding the Override Outline component.
One Renderer	Outline renders only the first detected mesh (i.e. the first children of the object)
First Children	Outline renders the mesh of all the first children of the object.
All Children	Outline renders the mesh of all the children of the object.
Default Color	Color of the outline.
Backface Culling	If true, the outline is visible also on back faced polygons.
Render All Children	If enabled, the outline is visible on all children in hierarchy.
Corner Outlines	If enabled, the corners of sharp objects are outlined correctly with a small impact on performance.
Precise Corner Outlines	If enabled, the sharp corner outline is improved with a higher impact on performance.

The advanced properties of the Border Outliner component are internal purposes features.

2.8.3 Camera Orbit

The Camera Orbit component is used in OPS applications to control the camera movement around the target object pivot point. This component is attached to the WEAVR Camera and allows configuring camera movement parameters.
To configure properties of Camera Orbit, go to *Hierarchy window > WEAVR > WEAVR Camera.*

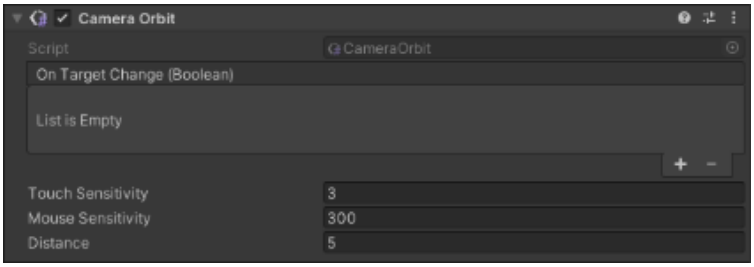


Figure 19. Camera Orbit

Properties

Property	Function
Touch Sensitivity	Define how sensitive the touch should be.
Mouse Sensitivity	Define how sensitive the mouse click should be.
Distance	Initial distance to the object.

NOTE
this component is linked to the Set Orbit Target action.

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

2.8.4 VR Hand

The visualization of hands in VR immersive scenes is configurable from the following components.

2.8.4.1 *Model*

The hands model helps visualize 3D hands in VT scenes. To change this model as desired, in the *Hierarchy window, go to WEAVR > PlayerRIG > PlayerVR > SteamVRObjects > Hand_Left/Hand_Right > Hand component.*

Figure 20. Render Model Prefab, VR_Hand

Properties

Property	Function
Render Model Prefab	3D model of the left/right hand to be visualized in a game.

2.8.4.2 *Terrain Hover Sphere Radius*

To make the interaction with objects on the ground easier, configure the hover sphere radius of motion controllers. The radius can increase proportionally after reaching a height threshold.
This parameter aims at avoiding interaction problems coming from a room setup miscalibration.
To configure the controller's hover sphere radius, in the *Hierarchy window, go to VR_RIG > Player > SteamVRObjects > Hand_Left/Hand_Right > VR_Hand.*

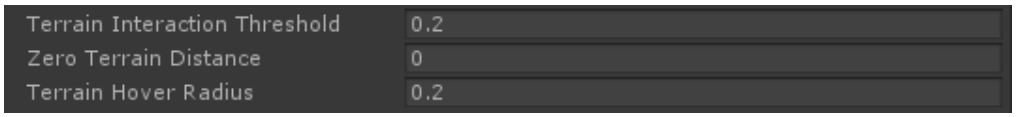


Figure 21. Terrain Hover Sphere Radius, VR Hand

Properties

Property	Function
Terrain Interaction Threshold	Controller's height threshold. The lower the value, the controller's hover sphere radius is increased more proportionally. The value is relative to the VR_RIG transform in the scene.
Zero Terrain Distance	Controller's height to be considered as a floor. This value is relative to the VR_RIG transform in the scene.
Terrain Hover Radius	Controller's hover sphere radius when the controller reaches the terrain height.

2.9 WEAVR Interactions

2.9.1 Overview

Interactions are components that can be associated with game objects to integrate/add a specific behavior.

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

The WEAVR contains predefined scripts (components) that can be added to the object to associate a certain behavior with it (e.g. to be able to grab an object in the scene, the Grabbable component is added).
This section describes the available interactions, their behaviors, and settings.

2.9.2 Object Behaviors

Object behaviors are components that allow an object to behave in a specific way.
In this section you will learn about the following components:

- VR Object
- Interactions Controller
- Grabbable
- Connectable
- Executable
- Operable
- Hinge Door
- Slide Door
- Panel Door
- Door Lock
- Knob
- Push Button
- Multi Button
- Two-Way Switch
- Three-Way Switch
- N-Way Switch
- Placeable
- Immersive VR Specific Manipulators

2.9.2.1 *VR_Object*

The VR Object component allows interacting with an object using controllers.

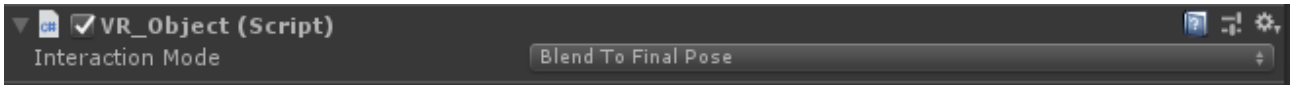


Figure 22. VR_Object

Properties

Property	Function
Interaction Mode:	How the hand interacts with the object. The interaction is different according to the selected mode.
Glue to Object:	Hand changes its position automatically and reaches the interaction point with no blending when a controller is close to the object.
Glue Distance	Minimum distance at which the controller glues to the object.
Glue Time	Time it takes for the hand to glue to the object.
Blend to Final Pose	Hand changes its position and reaches the interaction point of the object with blending.
None	Hand does not change its position when interacting with the object.

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

2.9.2.2 *Interactions Controller*

The Interactions Controller component is created automatically when you add the behavior to the object. Interactions controller keeps and controls the list of all interactions associated with the game object.

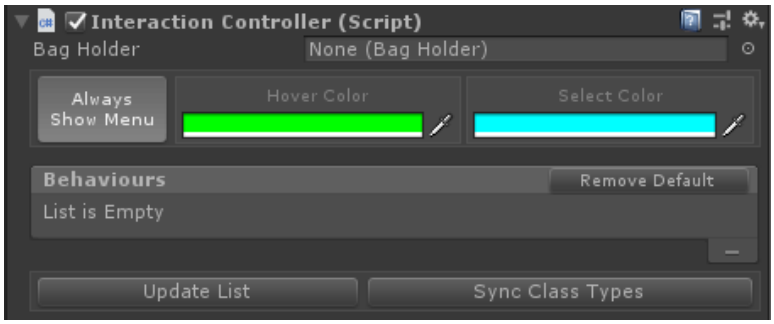


Figure 23. Interaction Controller

Properties

Property	Function
Bag Holder	*Reserved for future developments.*
Always Show Menu	If enabled, the available interaction with the object is shown even if there is only one behavior associated.
Hover Color	Color of an object outline when you hover over it.
Select Color	Color of an object outline when you select it.
Behaviors	List of behaviors/functions. If the behavior is set as default, it is associated with the trigger button of the controller
Update List	Updates the list of behaviors.
Sync Class Types	Every behavior is tagged through an object class. When clicked, the behaviors are synchronized with the same name if they are not the same yet.

NOTE
The behaviors, which are controlled by the Interactions Controller component, are enabled when hovering over the object collider. If the component does not have a collider, a new collider is created automatically together with Interactions Controller and is based on the object mesh.

ATTENTION
If an object does not have a mesh, but its children have meshes, then a collider is added to each child mesh. If, however, the object does not have child meshes, than a $1m^3$ collider is created by default. It is recommended to edit the collider to match the Game Object dimensions.

2.9.2.3 *Grabbable*

The Grabbable component enables you to pick up an object. The object appears as grabbed and follows the controller in the VR environment only. To connect two objects together, at least one of them must have the Grabbable component.

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

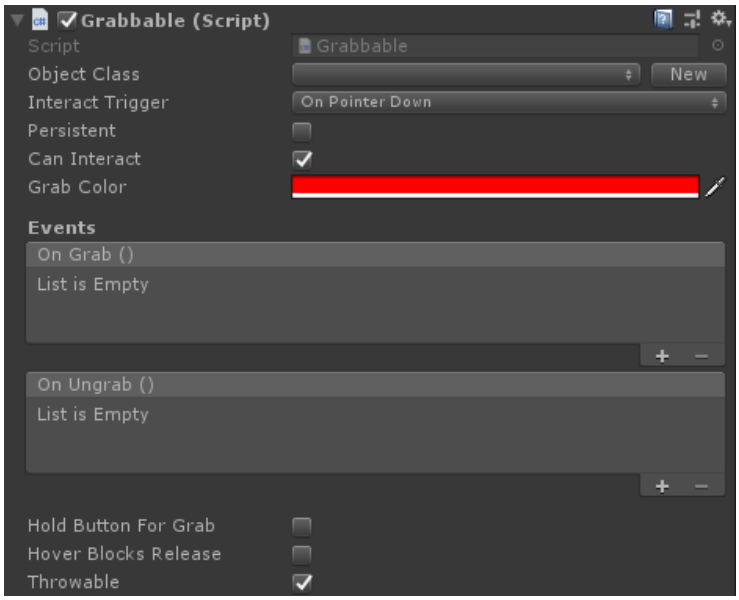


Figure 24. Grabbable

Properties

Property	Function
Object Class	Component tag that is used for interactions between behaviors.
Interact Trigger	Behavior is activated when the trigger is pressed (Pointer Down) or released (Pointer Up).
Persistent	If enabled, the behavior activates, even when another behavior is triggered on the same object.
Can Interact	If enabled, the behavior activates.
Grab Color	Color of the object outline when it is grabbed.
Hold Button for Grab (VR)	If enabled, the object is released when you release the trigger.
Hover Blocks Release	If enabled, the object is released when you press the trigger.
Throwable (VR)	If enabled, the object follows the physics rules when it is released. Otherwise, the object hangs in the air.
Snap on Attach	If not enabled, the grab position is the position of the controller on the object. Otherwise, on grab the object snaps in the defined Attachment Point. It works if the Skeleton Poser is null.
Skeleton Poser	Settings vary depending on the selected reference. For example, if you reference a Hand Pose, the Skeleton Poser component is attached to the GameObject automatically. The controller interacts with the object using the 3D hand. With no reference, the controller grabs the object.

The following cases illustrate the setup to grab the object with a hand model or a controller.
Case 1: the hand model is grabbing the object.

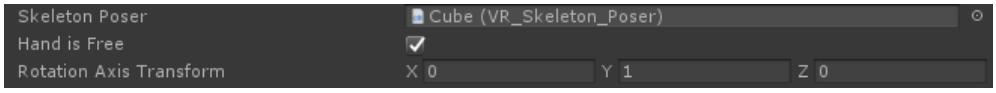


Figure 25. Grabbable if Hand Pose is used

Property	Function
Skeleton Poser	References the GameObject with the Skeleton Poser component. If the same GameObject of the Grabbable has a Skeleton Poser, and this setting is null, this Hand Pose is taken automatically.

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

Hand is Free (VR)	If enabled, the hand repeats movements of the controller defined in the Rotation Axis Transform property.
Rotation Axis Transform	Position of the axis around which the hand can move.

Case 2: the controller is grabbing the object.

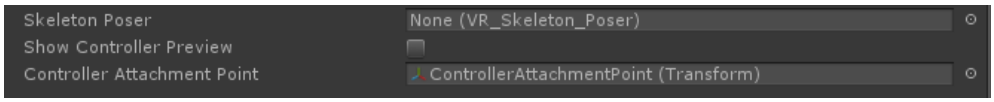


Figure 26. Grabbable if Hand Pose is not used

Property	Function
Skeleton Poser	References None. The GameObject should not have a Skeleton Poser component.
Show Controller Preview	If enabled, the preview of the controller (in respect to the GameObject when grabbing it) is shown in the scene.
Controller Attachment Point	The transform of the controller's attachment point in respect to the object. The transform is created automatically as a child of the GameObject and can be changed according to the required position.

2.9.2.4 *Connectable*

The Connectable component connects two game objects together. To connect successfully, both objects need to have the Connectable component or, at least, one of the objects needs to have the Grabbable component. For example, to connect a cable plug to the socket, you should grab the cable plug.
There are, however, cases when neither of the connectable objects has Grabbable components. In these cases, the connection is triggered by other interactions or actions.

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

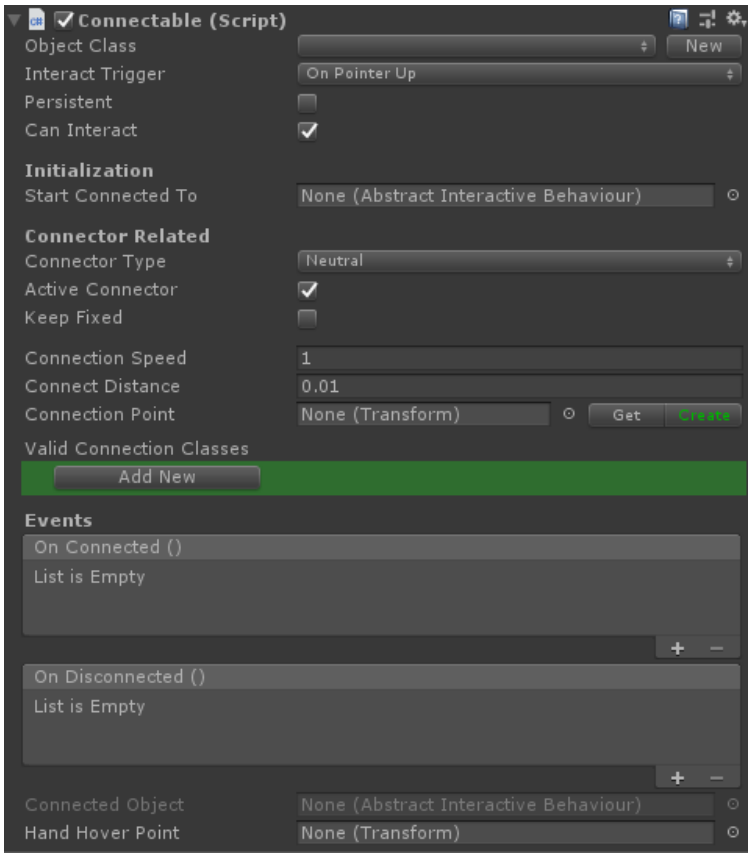


Figure 27. Connectable

Properties:

Property	Function
Object Class	Component tag that connects behaviors.
Interact Trigger	Behavior is activated when the trigger is pressed (Pointer Down) or released (Pointer Up).
Persistent	If not enabled, the behavior can be interrupted by other behaviors.
Can Interact	If enabled, the behavior activates.
Start Connected To	If referenced, when the game starts the objects are connected.
Connector Type	Select the appropriate connector type for two objects: either Neutral-Neutral or Male-Female.
Active Connector	If enabled, this object initiates the connection to the other object.
Keep Fixed	If enabled, the GameObject remains in a connected position even after you release it.
Use Constraints:	If enabled, sets constraints when connected.
Constraint Weight	
Break Force	
Fix Only One Axis	If enabled, allows rotation around the axis.
Local Axis to Fix	Axis around which you can rotate the object.
Connection Speed	Time to connect (in seconds).
Connect Distance	Distance at which this object connects to valid connectors (in meters).
Connection Point	Point where this object connects to other connectors.
Valid Connection Classes	Select the object that can connect to other objects which have a defined Object Class (the Connectable behavior).
Connected Object	It is for debug purpose. It Indicates the connected object on play.
Hand Hover Point	(VR only) Point at which a hand interacts with the object.

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

2.9.2.5 *Executable*

The Executable component triggers generic actions when interacting with the object.

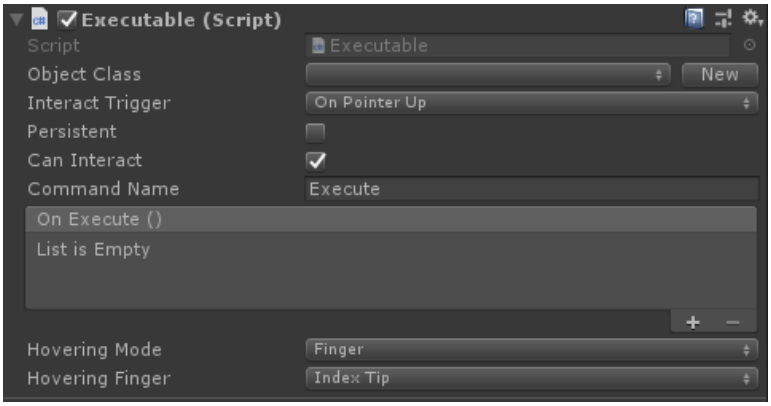


Figure 28. Executable

Properties:

Property	Function
Object Class	*Reserved for further development.*
Interact Trigger	Select whether to activate the behavior when the trigger on a controller is pressed (Pointer Down) or released (Pointer Up).
Persistent	If enabled, the behavior can be executed multiple times.
Can Interact	If enabled, the behavior is activated.
Command Name	Name of the command as it is displayed in the list of available behaviors in case of VT applications.
Hovering Mode:	(VR only)
Any	Hand uses any collider to interact with the object.
Finger	Hand uses a collider that is placed on the indicated part of a finger.
Skeleton Poser	Position of the hand when interacting with the object.
Hovering Finger	Part of a finger that interacts with the object.

2.9.2.6 *Operable*

The Operable component allows you to change the measure value that is stored in memory. For example, you can see how pressure value changes from 2 Pa to 10 Pa.

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

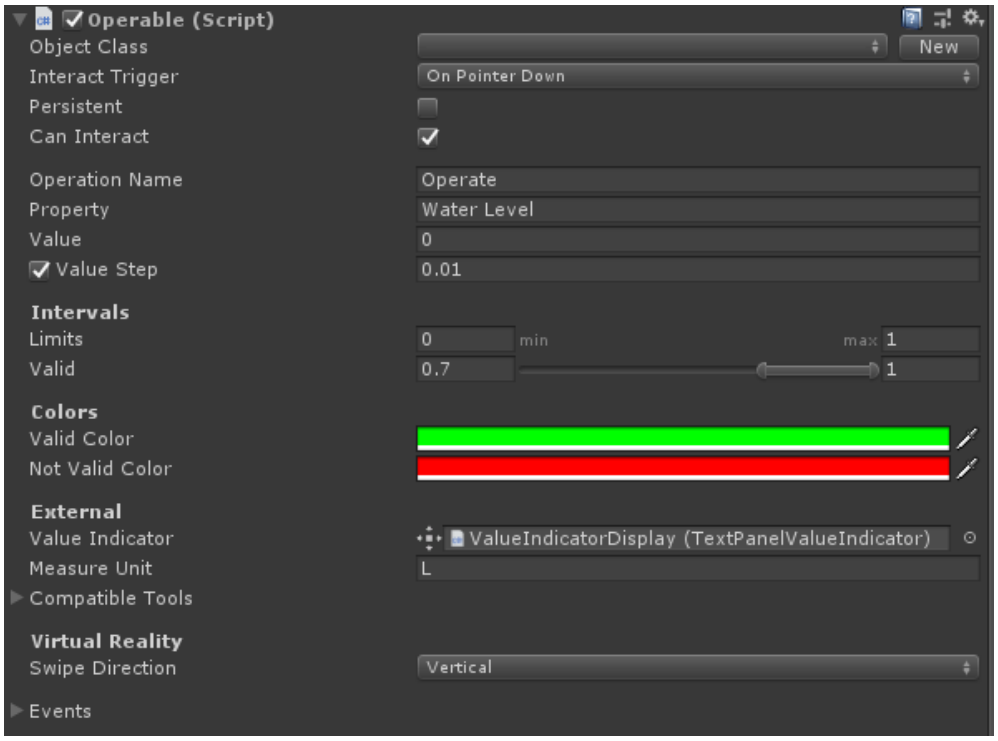


Figure 29. Operable

Properties:

Property	Function
Object Class	Component tag that other objects use to understand if this object can be operated or not.
Interact Trigger	Select whether to activate the behavior when the trigger on a controller is pressed (Pointer Down) or released (Pointer Up).
Persistent	If enabled, the behavior can be executed multiple times.
Can Interact	If enabled, the behavior is activated.
Operation Name	Name of the operation as it is displayed in the list of available behaviors in case of VT applications.
Property	Name of the property controlled by the Operable component (e.i. water level, torque)
Value	Starting value.
Value Step	If enabled, enter an increment or decrement for the value.
Limits	Min and max values available.
Valid	Range of valid values.
Valid Color	Color of the number if it is in the valid range.
Not Valid Color	Color of the number if it is not in the valid range.
Value Indicator	GameObject with the Text Panel Value Indicator component (Section 2.9.2.6.1) that displays the operable current value.
Compatible Tools:	Tools that can be used to interact with hands.
Size	Number of compatible tools.
Element	GameObject that can interact with the operable GameObject (e.i. screwdriver).
Swipe Direction	(VR only) Direction in which you swipe the controller touchpad to change the value.

2.9.2.6.1 Text Panel Value Indicator

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

The Text Panel Value Indicator component is associated with the Operable behaviour and is used to control the UI that displays the current Operable value, its unit of measure, and its name.

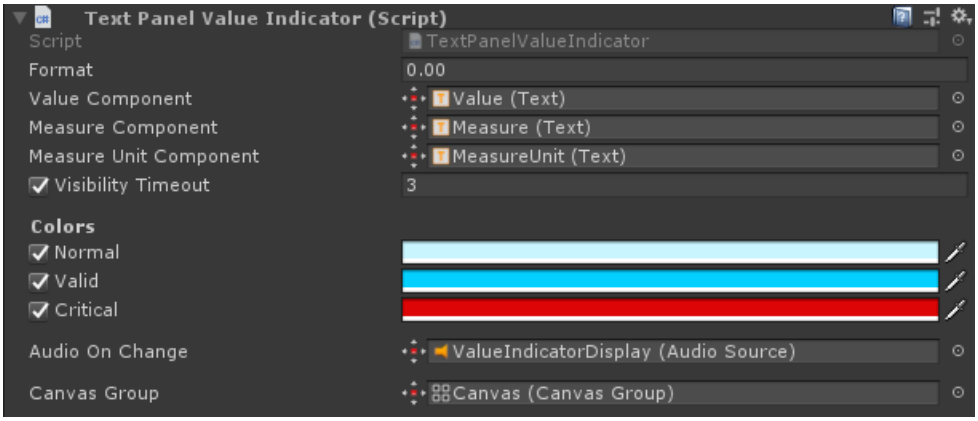


Figure 30. Text Panel Value Indicator

Properties:

Property	Function
Format	Text format.
Value Component	Text component that displays the operable value.
Measure Component	Text component that displays the measure name.
Measure Unit Component	Text component that displays the unit of measure.
Visibility Timeout	If enabled, the time (in seconds) after which the Canvas Group displaying the operable values hides if there was no interaction with the Operable GameObject.
Colors:	
Normal	If enabled, text color of the Value component if its value is lower than the minimum value of the operable range (indicated in the Operable component).
Valid	If enabled, text color of the Value component if its value is in the valid range (indicated in the Operable component).
Critical	If enabled, text color of the Value component if its value is higher than the maximum value of the operable range (indicated in the Operable component).
Audio on Change	Audio source that is used when the Operable value changes.
Canvas Group	Canvas owner of the Value, Measure, and Measure Unit components.

2.9.2.7 *Hinge Door*

When assigned to a Game Object, the Hinge Door component allows you to open the object like a door with the hinge system. The user can move the object on its pivot point from the "Opened" to the "Closed" position and vice versa.

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

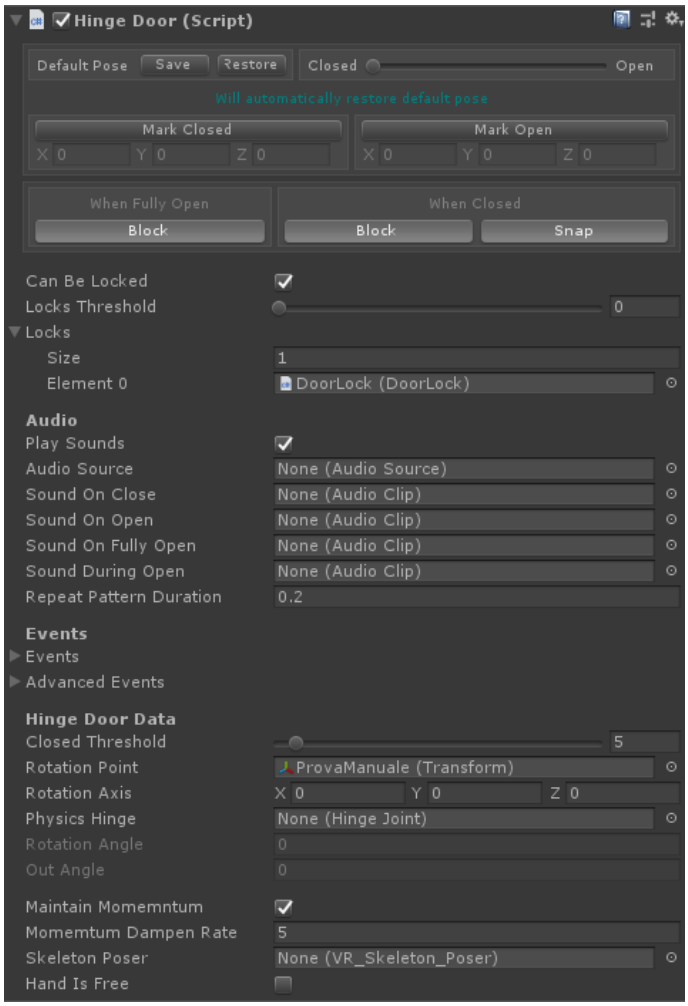


Figure 31. Hinge Door

Properties:

Property	Function
Default Pose:	
Save	Saves the position indicated in the sidebar as default. It will be the object position when entering the play mode.
Restore	Restores the position of an object to the one saved as default or the "Closed" position if it was not changed.
Mark Closed	Saves the "Closed" position based on the current transform of the Game Object.
Mark Open	Saves the "Open" position based on the current transform of the Game Object.
When Fully Open	If Block is enabled, the interaction (e.g. door grabbing) ends when the door is fully open.
When Closed	If Block is enabled, the interaction (e.g. door grabbing) ends when the door is fully closed. If Snap is enabled, the door snaps to its closed position when you close it.
Can be Locked	If enabled, the Hinge Door movement is not allowed if the indicated Locks are in a locked mode.
Locks Threshold	It is the number of Locks that should be unlocked to open the Slide Door.
Locks:	
Size	Number of locks associated with the Game Object.

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

Element	The Lock object (requires the Door Lock component).
Play Sounds:	If enabled, you can configure a sound system.
Audio Source	Object that is a source of audio (requires the Audio Source component).
Sound on Close/Open/Fully Open/During Open	Audio clip that is played when the object is closed, open, fully open, or while opening.
Repeat Pattern Duration	Duration of the audio to be repeated. The Sound During Open property must be configured.
Closed Threshold	Degree at which the object is considered to be closed, and the movement finishes automatically.
Rotation Point	Pivot point of the GameObject. It is found automatically when the Open and Close positions are defined.
Rotation Axis	Axis around which the object moves. It is found automatically when two positions are defined.
Physics Hinge	*Reserved for further development.*
Rotation Angle	(Used for debugging) Angle between the closed and open position. It is found automatically when two positions are defined).
Out Angle	(Used for debugging) Current angle position.
Maintain Momentum	If enabled, keeps the door moving after you release it.
Momentum Dampen Rate	Speed at which the door keeps moving after you release it.
Skeleton Poser	(VR only) Position of the hand when interacting with the object.
Hand is Free	If enabled, the hand repeats movements of the controller defined in the Rotation Axis Transform property.

ATTENTION
To move the GameObject according to the Hinge Door component, make sure that the Animator component is disabled for this object.

2.9.2.8 *Slide Door*

When associated with a Game Object, the Slide Door component allows you to slide the object from the Open to Closed position and vice versa.

NOTE
The Rotation property of the Transform component is not allowed for the motion of the object as Slide Door. The object can slide using more than one axis. This will be a linear motion.

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

Figure 32. Slide Door

Properties:

Property	Function
Default Pose:	
Save	Saves the position indicated in the sidebar as default. It will be the object position when entering the play mode.
Restore	Restores the position of an object to the one saved as default or the "Closed" position if it was not changed.
Mark Closed	Saves the "Closed" position based on the current transform of the Game Object.
Mark Open	Saves the "Open" position based on the current transform of the Game Object.
When Fully Open	If Block is enabled, the interaction (e.g. door grabbing) ends when the door is fully open.
When Closed	If Block is enabled, the interaction (e.g. door grabbing) ends when the door is fully closed. If Snap is enabled, the door snaps to its closed position when you close it.
Can be Locked	If enabled, the Slide Door movement is not allowed if the indicated Locks are in a locked mode.
Locks Threshold	Number of door locks that should be unlocked to open the Slide Door.
Locks:	
Size	Number of locks associated with the Game Object.
Element	The Lock object (requires the Door Lock component).
Play Sounds:	If enabled, you can configure a sound system.
Audio Source	Object that is a source of audio (requires the Audio Source component).

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

Sound on Close/Open/Fully Open/During Open	Audio clip that is played when the object is closed, open, fully open, or while opening.
Repeat Pattern Duration	Duration of the audio to be repeated. The Sound During Open property must be configured.
Closed Threshold	Degree at which the object is considered to be closed, and the movement finishes automatically.
Closing Point	Transform of the Closed position. It is generated automatically when defining the Closed position.
Fully Opened Point	Transform of the Fully Opened position. It is generated automatically when defining the Open position.
Maintain Momentum	If enabled, keeps the door moving after you release it.
Momentum Dampen Rate	Speed at which the door keeps moving after you release it.
Skeleton Poser	(VR only) Position of the hand when interacting with the object.

ATTENTION
To move the GameObject according to the Slide Door component, make sure that the Animator component is disabled for this object.

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com |website: www.txtgroup.com | Tel: +49 30 293620

2.9.2.9 *Panel Door*

When associated with a Game Object, the Panel Door component allows you to move the object like a panel and, in case of VR applications, grab and place it elsewhere. The "Open" and "Closed" positions are the final points of the associated animation when the door is opened or closed.

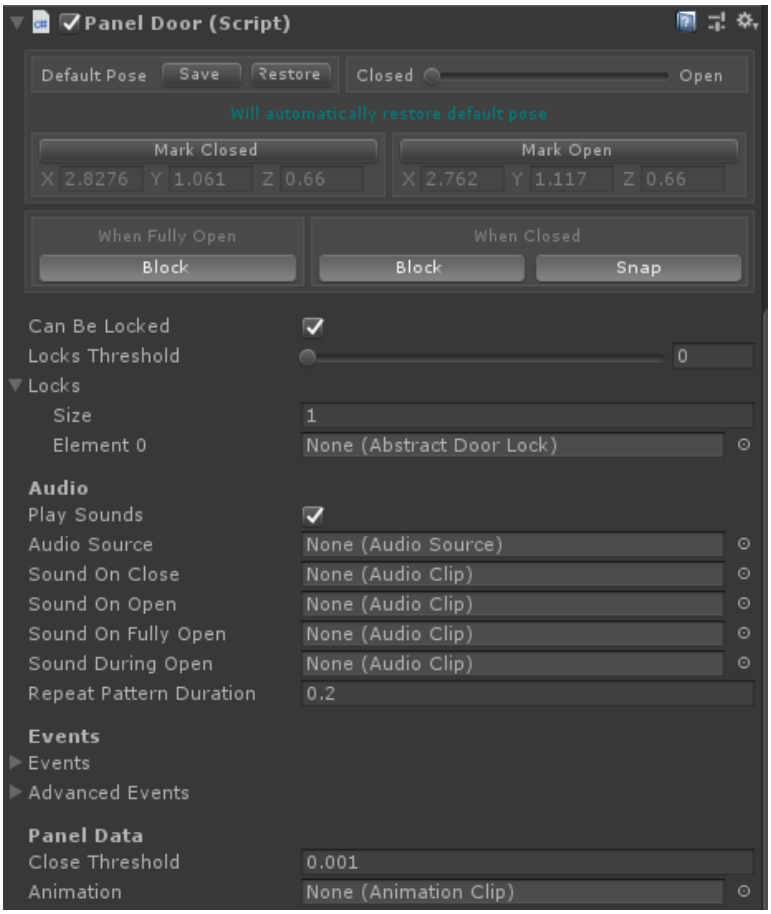


Figure 33. Door Panel

Properties:

Property	Function
Default Pose:	
Save	Saves the position indicated in the sidebar as default. It will be the object position when entering the play mode.
Restore	Restores the position of an object to the one saved as default or the "Closed" position if it was not changed.
Mark Closed	Saves the "Closed" position based on the current transform of the Game Object.
Mark Open	Saves the "Open" position based on the current transform of the Game Object.
When Fully Open	If Block is enabled, the interaction (e.g. door grabbing) ends when the door is fully open.
When Closed	If Block is enabled, the interaction (e.g. door grabbing) ends when the door is fully closed. If Snap is enabled, the door snaps to its closed position when you close it.
Can be Locked	If enabled, the Panel Door movement is not allowed if the indicated Locks are in a locked mode.

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

Locks Threshold	Number of door locks that should be unlocked to open the Slide Door.
Locks:	
Size	Number of locks associated with the Game Object.
Element	The Lock object (requires the Door Lock component (2.9.2.9).
Play Sounds:	If enabled, you can configure a sound system.
Audio Source	Object that is a source of audio (requires the Audio Source component).
Sound on Close/Open/Fully Open/During Open	Audio clip that is played when the object is closed, open, fully open, or while opening.
Repeat Pattern Duration	Duration of the audio to be repeated. The Sound During Open property must be configured.
Closed Threshold	Degree (in meters) at which the object is considered to be closed, and the movement finishes automatically.
Animation	Animation clip to be reproduced when opening or closing the door.

2.9.2.10 *Door Lock*

The Door Lock component defines the Game Object as a locking element of a door object.

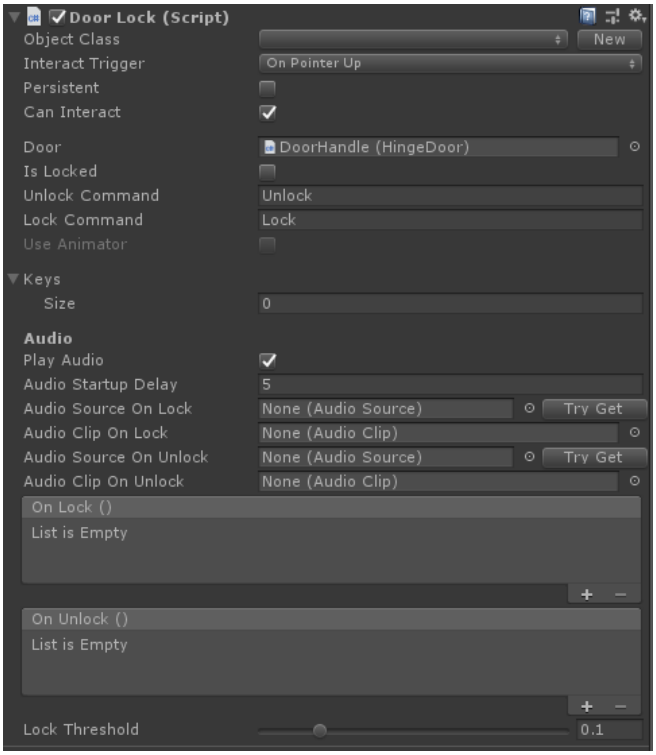


Figure 34. Door Lock

Properties:

Property	Function
Object Class	*Reserved for future development*.
Interact Trigger	Select whether to activate the behavior when the trigger on a controller is pressed (Pointer Down) or released (Pointer Up).
Persistent	If enabled, the behavior can be executed multiple times.
Can Interact	If enabled, the behavior is activated.
Door	Object with any type of a door component.

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

Is Locked	If enabled, the Lock is locked. You can change the state by clicking or triggering the object (in immersive VR environment).
Unlock Command	Action name that appears when hovering over the locked object.
Lock Command	Action name that appears when hovering over the unlocked object.
Use Animator	If enabled, when the lock status changes the animator associated to the same GameObject parameters are triggered (Lock/Unlock)
Keys	Generic objects that interact with the door lock.
Size	Number of elements.
Element	Generic GameObject. If None is selected, the door lock is interactable with anything.
Play Audio:	If enabled, you can configure the sound system.
Audio Startup Delay	Delay before playing the audio file.
Audio Source on Lock	Audio source that is played when the object is locked (requires the Audio Source component). To set an audio source automatically, click Try Get.
Audio Clip on Lock	Audio clip that is played when the object is locked.
Audio Source on Unlock	Audio source that is played when the object is unlocked (requires the Audio Source component). To set an audio source automatically, click Try Get.
Audio Clip on Unlock	Audio clip that is played when the object is unlocked.
Lock Threshold	Degree at which the object is considered to be locked.

2.9.2.11 *Knob*

The Knob component allows the associated GameObject to behave like a knob and rotate according to the defined axis.

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com |website: www.txtgroup.com | Tel: +49 30 293620

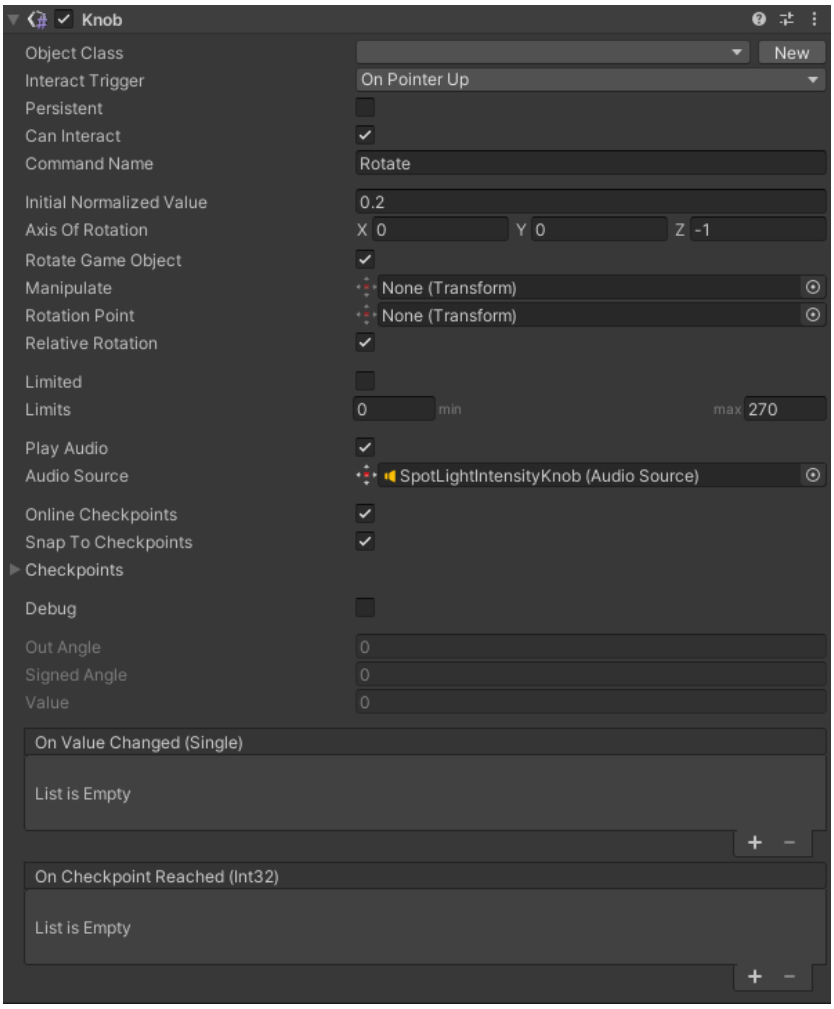


Figure 35. Knob

Properties:

Property	Function
Object Class	*Reserved for future development*.
Interact Trigger	(VR only) Select whether to activate the behavior when the trigger on a controller is pressed (Pointer Down) or released (Pointer Up).
Persistent	If enabled, the behavior can be executed multiple times.
Can Interact	If enabled, the behavior is activated.
Command Name	Name of an action that is displayed when hovering over an object.
Initial Normalized Value	Initial value when the game starts.
Axis of Rotation	Axis around which the circular drive rotates in local space.
Rotate Game Object	If enabled, the transform of the Manipulate property is rotated.
Manipulate	GameObject to rotate. If None, the current GameObject is set.
Rotation Point	Pivot point of rotation.
Relative Rotation	If enabled, the movement continues from the last rotation.
Limited	If enabled, the rotation is limited from [minAngle] to [maxAngle].
Limits	Minimum and maximum limited angles.
Force Start	If enabled, the starting angle is StartAngle that is limited to [minAngle and maxAngle].
Play Audio	If enabled, an audio file is played.
Audio Source	Audio file that is played.
Online Checkpoints	If enabled, all checkpoints are considered when changing values.

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

Snap to Checkpoints:	If enabled, snaps to a Checkpoint position when close to it.
Size	Number of elements.
Element	Checkpoint element
Value	Normalized value [0, 1].
Epsilon	Threshold to reach the Checkpoint.
Span	If enabled, the Checkpoint value becomes a range value instead of a single value.
Out Angle	(Used for debugging) shows the current angle position.

2.9.2.12 *Push Button*

The Push Button component allows you to press the GameObject like a pushbutton.

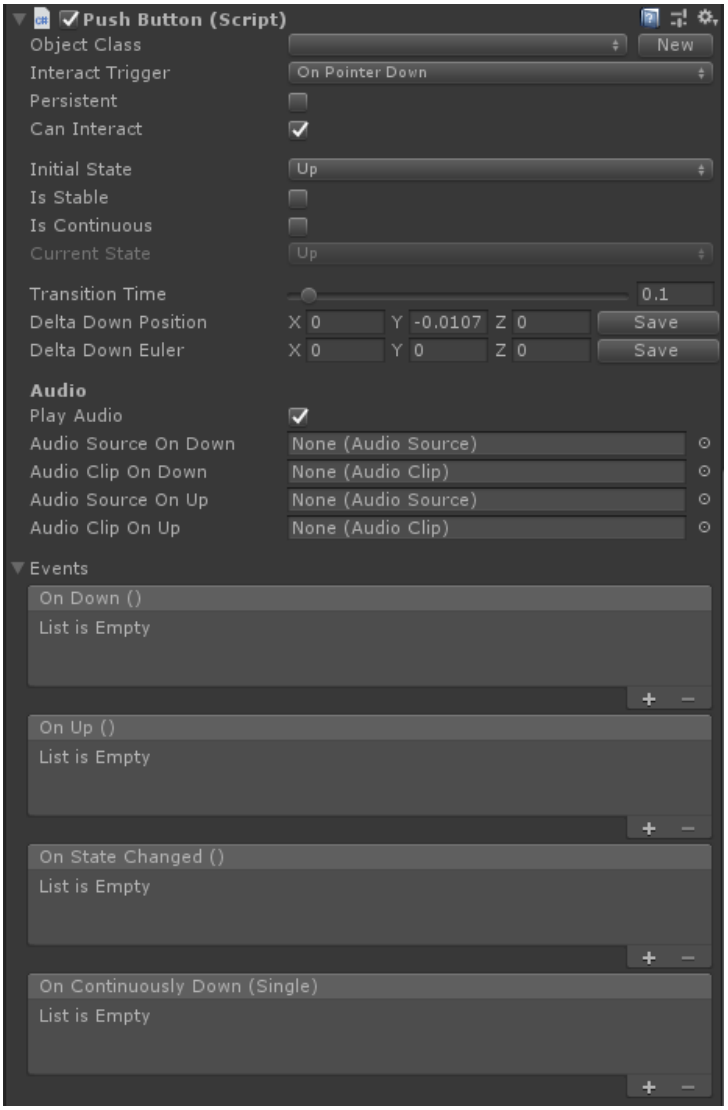


Figure 36. Push Button

Properties:

Property	Function
Object Class	Component tag that connects behaviors (can be null).
Interact Trigger	(VR only) Select whether to activate the behavior when the trigger on a controller is pressed (Pointer Down) or released (Pointer Up).

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

Persistent	If enabled, the behavior can be executed multiple times.
Can Interact	If enabled, the behavior is activated.
Initial State	Position of the GameObject when the game starts.
Is Stable	If enabled, the button is pressed (keeps the Down position). Otherwise, the button reaches the new position and immediately goes back to the previous one.
Is Continuous	If enabled, the button is pressed (keeps the Down position) while you press the trigger on a controller.
Hover Lock	If enabled, the button is triggered until the trigger button is released. Otherwise, the button is triggered until the controller hovers the object.
Current State	Information for debugging.
Transition Time	Time to reach a new position.
Delta Down Position	Translation delta to the Down position. To set this property according to the current position of the object in the scene and its previous position, click Save.
Delta Down Euler	Rotation delta to the Down position. To set this property according to the current position of the object in the scene and its previous position, click Save.
Play Audio:	If enabled, you can configure a sound system.
Audio Source on Down	Audio source that is played when the object reaches the Down position (requires the Audio Source component).
Audio Clip on Down	Audio clip that is played when the object reaches the Down position.
Audio Source on Up	Audio source that is played when the object reaches the Up position (requires the Audio Source component).
Audio Clip on Up	Audio clip that is played when the object reaches the Up position.

2.9.2.13 *Multi Button*

The Multi Button component allows you to interact with the GameObject like with a pushbutton. When you press the button, I event is triggered according to the current Event Group set. The switch among Event Groups can be controlled from different places: the same Event Group (example in Figure 37), another component event, or the procedure itself.

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com |website: www.txtgroup.com | Tel: +49 30 293620

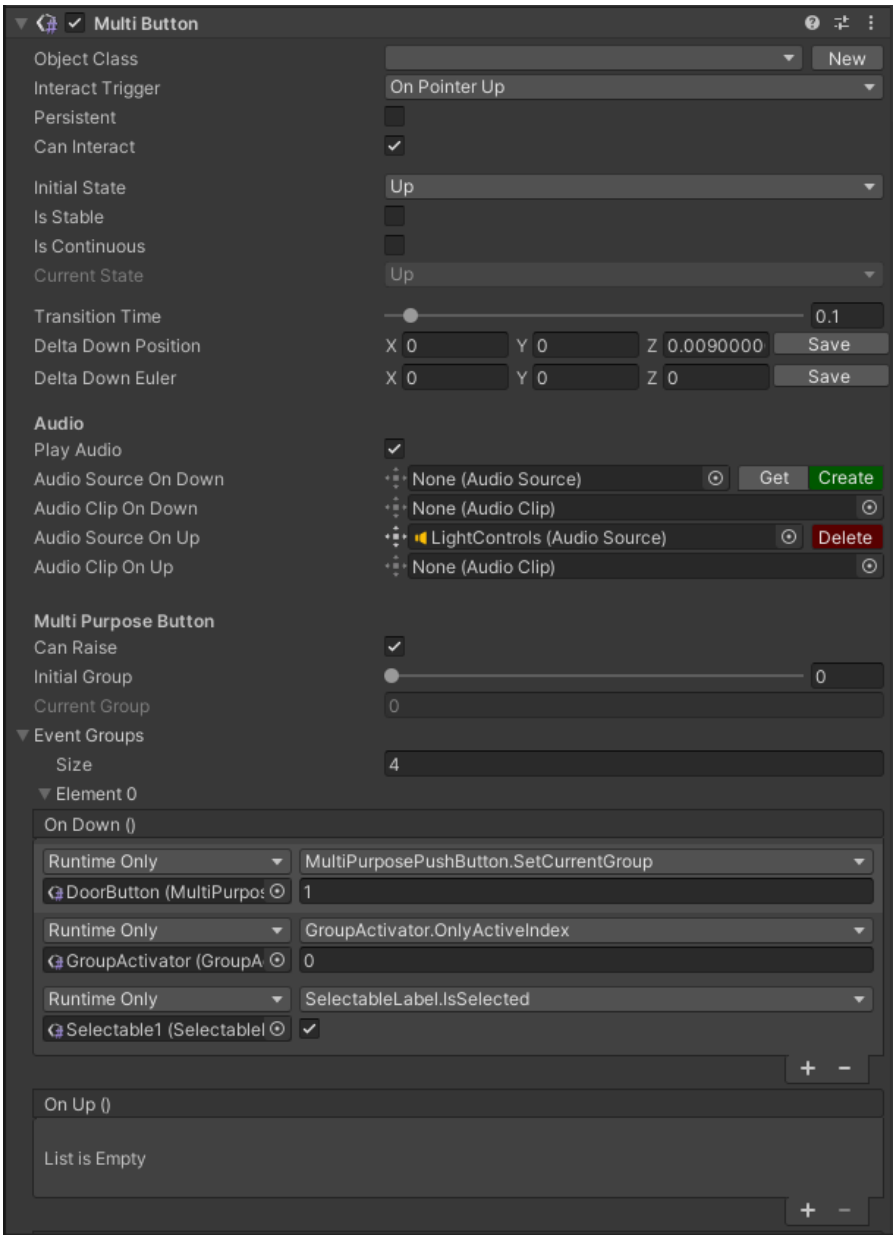


Figure 37. Multi Button

Properties:

Property	Function
Object Class	Component tag that connects behaviors (can be null).
Interact Trigger	(VR only) Select whether to activate the behavior when the trigger on a controller is pressed (Pointer Down) or released (Pointer Up).
Persistent	If enabled, the behavior can be executed multiple times.
Can Interact	If enabled, the behavior is activated.
Initial State	Position of the GameObject when the game starts.
Is Stable	If enabled, the button is pressed (keeps the Down position). Otherwise, the button reaches the new position and immediately goes back to the previous one.
Is Continuous	If enabled, the button is pressed (keeps the Down position) while you press the trigger on a controller.
Current State	Information for debugging.
Transition Time	Time to reach a new position.
Delta Down Position	Translation delta to the Down position.

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

	To set this property according to the current position of the object in the scene and its previous position, click Save.
Delta Down Euler	Rotation delta to the Down position. To set this property according to the current position of the object in the scene and its previous position, click Save.
Play Audio:	If enabled, you can configure a sound system.
Audio Source on Down	Audio source that is played when the object reaches the Down position (requires the Audio Source component).
Audio Clip on Down	Audio clip that is played when the object reaches the Down position.
Audio Source on Up	Audio source that is played when the object reaches the Up position (requires the Audio Source component).
Audio Clip on Up	Audio clip that is played when the object reaches the Up position.
Can Raise	If enabled, the event of the Event Groups set is called.
Initial Group	Event Group set when starting the game mode.
Current Group	For debug purposes, it indicates the current group when in play
Event Groups:	
Size	Number of Event Groups.
Element	Event system of the specific Event Group.

2.9.2.14 *Two-Way Switch*

The Two-Way Switch component allows you to interact with the GameObject like with a two-way switch.

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

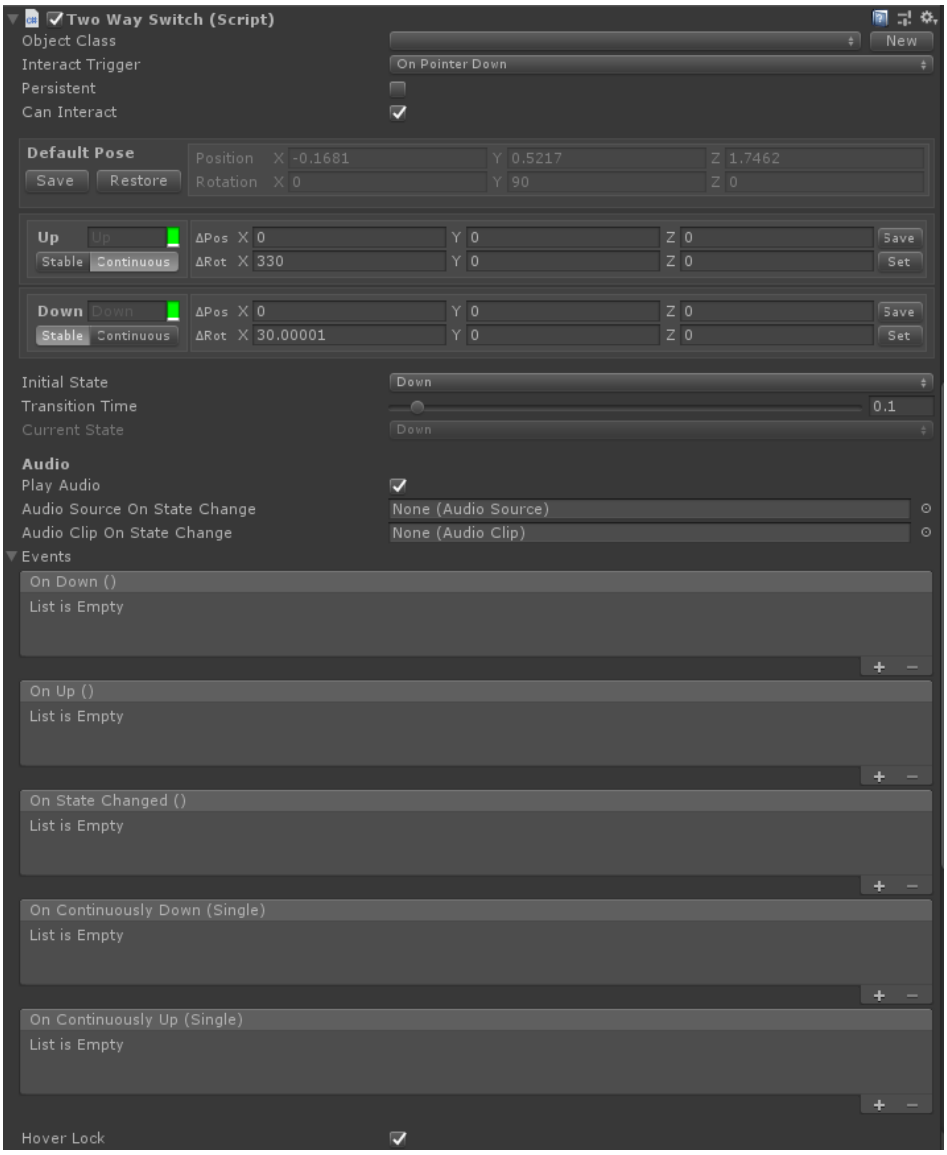


Figure 38. Two Way Switch

Properties:

Property	Function
Object Class	Component tag that connects behaviors (can be null).
Interact Trigger	(VR only) Select whether to activate the behavior when the trigger on a controller is pressed (Pointer Down) or released (Pointer Up).
Persistent	If enabled, the behavior can be executed multiple times.
Can Interact	If enabled, the behavior is activated.
Default Pose	Position of the GameObject from which the delta movement for the Poses is calculated. To configure the Default Pose as the current transform of the GameObject, click Save. To restore the GameObject to the default position, click Restore.
Up/Down:	Positions of a GameObject. To configure the Up/Down position as the current transform of the GameObject, click Save. To configure the Up/Down position for the GameObject, click Set.
Stable	If highlighted, the switch keeps this position.
Continuous	If highlighted, the switch keeps this position until you release the trigger.

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

Initial State	State of the switch when entering the game mode.
Transition Time	Time to reach a new position.
Current State	Information for debugging.
Play Audio:	If enabled, you can configure a sound system.
Audio Source on State Change	Audio source that is played when the object reaches a new position (requires the Audio Source component).
Audio Clip State Change	Audio clip that is played when the object reaches a new position.
Hover Lock	If enabled, the switch is triggered until you release the trigger button. Otherwise, the switch is triggered until you hover over the object with the controller.

2.9.2.15 *Three-Way Switch*

The Tree-Way Switch component allows you to interact with the GameObject like with a three-way switch.

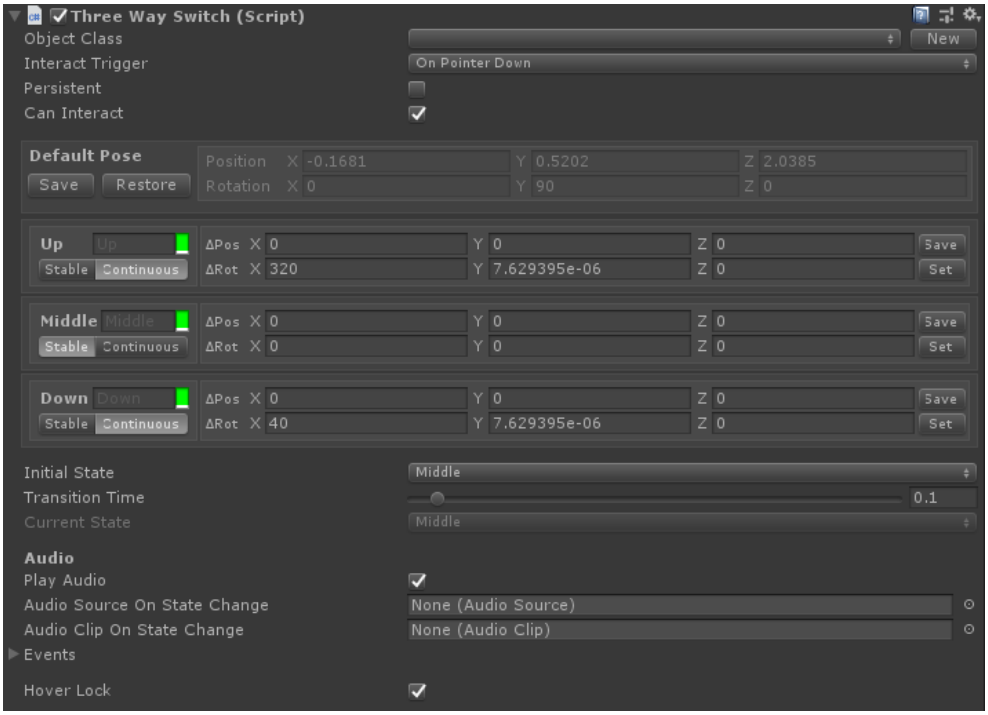


Figure 39. Three Way Switch

Properties:

Property	Function
Object Class	Component tag that connects behaviors (can be null).
Interact Trigger	(VR only) Select whether to activate the behavior when the trigger on a controller is pressed (Pointer Down) or released (Pointer Up).
Persistent	If enabled, the behavior can be executed multiple times.
Can Interact	If enabled, the behavior is activated.
Default Pose	Position of the GameObject from which the delta movement for the Poses is calculated. To configure the Default Pose as the current transform of the GameObject, click Save. To restore the GameObject to the default position, click Restore.

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

Up/Middle/Down:	Positions of a GameObject. To configure the Up/Down position as the current transform of the GameObject, click Save. To configure the Up/Down position for the GameObject, click Set.
Stable	If highlighted, the switch keeps this position.
Continuous	If highlighted, the switch keeps this position until you release the trigger.
Initial State	State of the switch when entering the game mode.
Transition Time	Time to reach a new position.
Current State	Information for debugging.
Play Audio:	If enabled, you can configure a sound system.
Audio Source on State Change	Audio source that is played when the object reaches a new position (requires the Audio Source component).
Audio Clip State Change	Audio clip that is played when the object reaches a new position.
Hover Lock	If enabled, the switch is triggered until you release the trigger button. Otherwise, the switch is triggered until you hover over the object with the controller.

NOTE
You can configure how the switch changes positions with the help of the VR_Slider manipulator (Section 2.9.2.18.2). The starting position of the manipulator corresponds to the Down position of the switch.

2.9.2.16 *N-Way Switch*

The N-Way Switch component allows you to interact with the GameObject like with a switch that has as many ways as required.

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

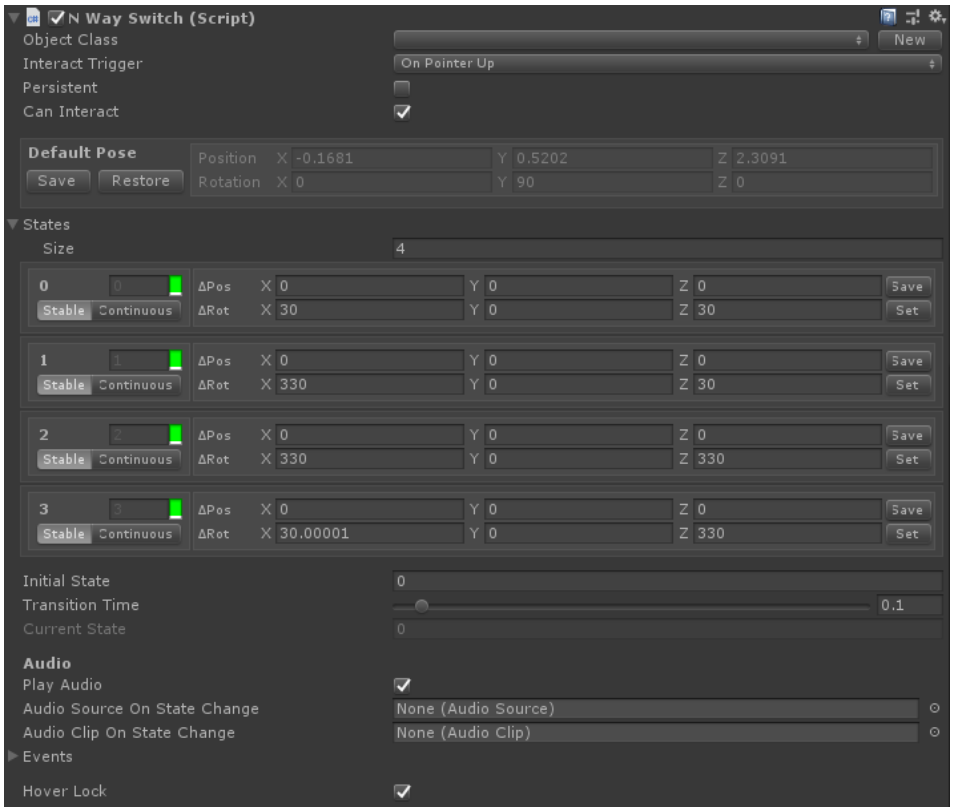


Figure 40. No Way Switch

Properties:

Property	Function
Object Class	Component tag that connects behaviors (can be null).
Interact Trigger	(VR only) Select whether to activate the behavior when the trigger on a controller is pressed (Pointer Down) or released (Pointer Up).
Persistent	If enabled, the behavior can be executed multiple times.
Can Interact	If enabled, the behavior is activated.
Default Pose	Position of the GameObject from which the delta movement for the Poses is calculated. To configure the Default Pose as the current transform of the GameObject, click Save. To restore the GameObject to the default position, click Restore.
States:	Positions of a GameObject. To configure the State position as the current transform of the GameObject, click Save. To configure the State position for the GameObject, click Set.
Size	Number of ways a switch can be positioned.
Stable	If highlighted, the switch keeps this position.
Continuous	If highlighted, the switch keeps this position until you release the trigger.
Initial State	State of the switch when entering the game mode.
Transition Time	Time to reach a new position.
Current State	Information for debugging.
Play Audio:	If enabled, you can configure a sound system.
Audio Source on State Change	Audio source that is played when the object reaches a new position (requires the Audio Source component).
Audio Clip State Change	Audio clip that is played when the object reaches a new position.

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

Hover Lock	If enabled, the switch is triggered until you release the trigger button. Otherwise, the switch is triggered until you hover over the object with the controller.

2.9.2.17 *Placeable*

The Placeable component associates a GameObject with transform locations where it can be placed in the scene.

To define the available transform locations, use Place Points that are created using the Place Manager component. For example, you can place a screwdriver on a specific location, like the left side of the table, automatically after releasing the trigger from the grabbable behavior.

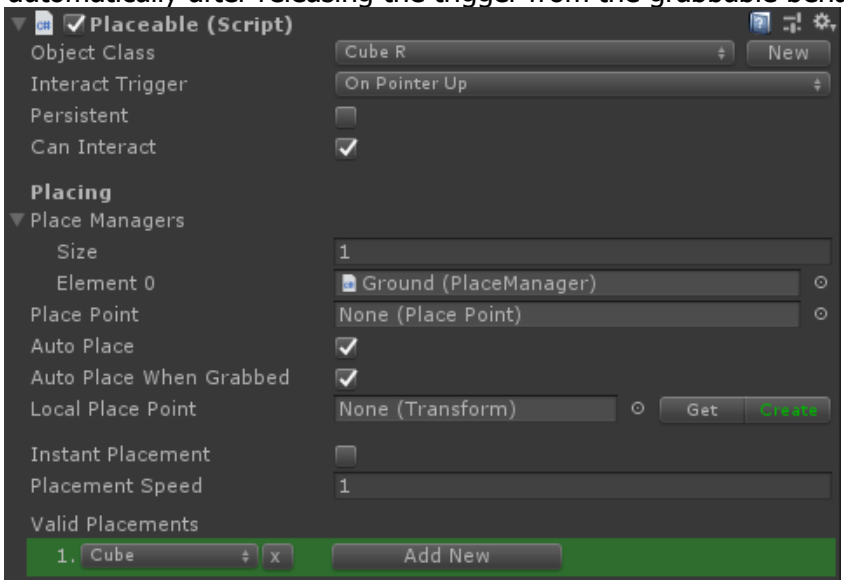


Figure 41. Placeable

Properties:

Property	Function
Object Class	Component tag that determines places where this object can be placed.
Interact Trigger	(VR only) Select whether to activate the behavior when the trigger on a controller is pressed (Pointer Down) or released (Pointer Up).
Persistent	If enabled, the behavior can be executed multiple times.
Can Interact	If enabled, the behavior is activated.
Place Managers:	Additional settings for Place Managers.
Size	Number of associated Place Managers.
Element	Associated Place Manager.
Place Point	Current location of the object.
Auto Place	(VR only) If enabled, the object is placed to a free place point when the object is not connected or grabbed.
Auto Place When Grabbed	If enabled, the object is relocated back to its location automatically when you release it from the grab.
Local Place Point	Select the object-related transform according to which the object is placed in the Place Point.
Instant Placement	If enabled, the object is placed immediately without an animation).
Placement Speed	Speed of an object during the animation (in m/s).
Valid Placements	Object Class of Place Managers that can be associated with this object.

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

NOTE
To synchronize the Placeable object with the Place Manager component, click the Sync with Placeable button in the Place Manager component.

2.9.2.17.1 Place Manager

The Place Manager component creates available Place Points (current locations) for the associated object with the help of the Placeable component.

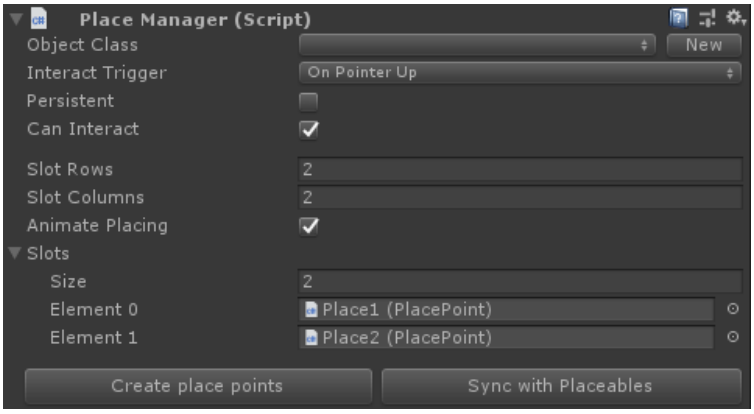


Figure 42. Place Manager

Properties:

Property	Function
Object Class	Component tag that is used by placeable components to validate the placement.
Interact Trigger	(VR only) Select whether to activate the behavior when the trigger on a controller is pressed (Pointer Down) or released (Pointer Up).
Persistent	If enabled, the behavior can be executed multiple times.
Can Interact	If enabled, the behavior is activated.
Slot Rows	Number of rows of the array of Place Points that are created automatically.
Slot Columns	Number of columns of the array of Place Points that are created automatically.
Animate Placing	If enabled, the placement of an object is animated.
Slots:	Additional settings for slots. To create Place Points according to the defined rows and columns, click Create Place Points. To synchronize Place Points with placeable objects that have Object Classes in the Valid Placement list, click Sync with Placeables. You must enable the object first.
Size	Number of elements.
Element	Created Place Points.

2.9.2.18 *Immersive VR Specific Manipulators*

Manipulator components are used to integrate specific functionalities with certain Behaviour components that are part of the same GameObject. The Behaviours that can work with manipulators are the following:

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

- Operable
- Door Lock
- Two-Way Switch
- Three-Way Switch
- N-Way Switch

2.9.2.18.1 Rotator

The Rotator component uses the rotate gesture of the controller to change the Behaviour value of an object. For example, you can integrate the rotation gesture with a Door Lock (Section 2.9.2.10); in this case, the Rotator maximum and minimum angles correspond to the lock and unlock positions of the Door Lock component.

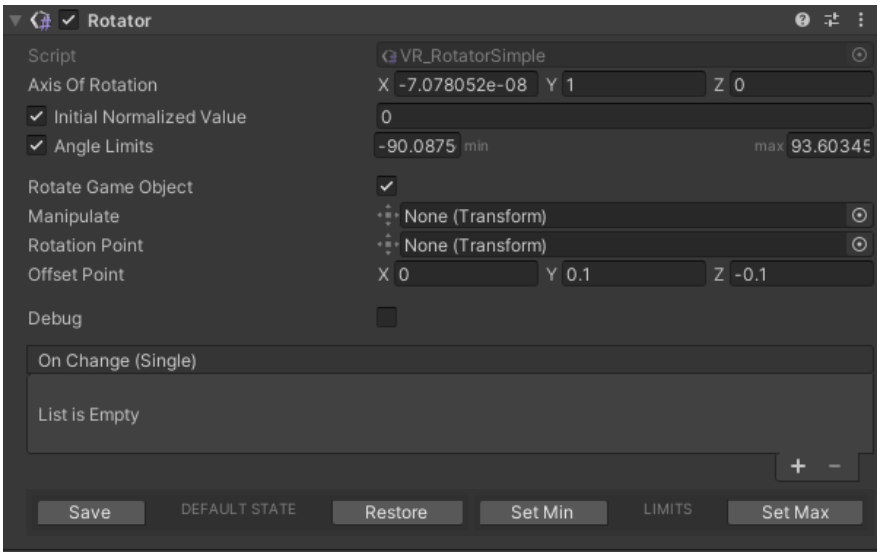


Figure 43. Rotator

Properties:

Property	Function
Axis of Rotation	Axis around which the circular drive rotates in local space.
Initial Normalized Value	If enabled, the initial value when the game starts.
Angle Limits	If enabled, define the limited rotation angles.
Rotate Game Object:	If enabled, the transform of the Manipulate property is rotated.
Manipulate	GameObject to rotate. If None, the current GameObject is set.
Rotation Point	Pivot point of rotation. If you select None, the current GameObject is set.
Offset Point	Offset according to the pivot point of the object that increases its responsiveness.
Debug	If enabled, shows the Offset position in the scene.
Save	To save the property value from the scene status, click Save.
Restore	To restore the property value from the scene status, click Restore.
Set Min/Max	To configure the property value from the scene status, click Set Min or Set Max.

NOTE
To work with the Rotator component, configure the Interact Trigger property of the associated component (for example, a Door Lock) to be activated when the trigger is pressed (Pointer Down).

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

2.9.2.18.2 Slider

The Slider component uses the slide gesture of the controller to change the Behaviour value of an object. For example, you can integrate the slide gesture with a Door Lock (Section 2.9.2.10); in this case, Slider Start and End Positions correspond to the Lock and Unlock positions of the Door Lock component.

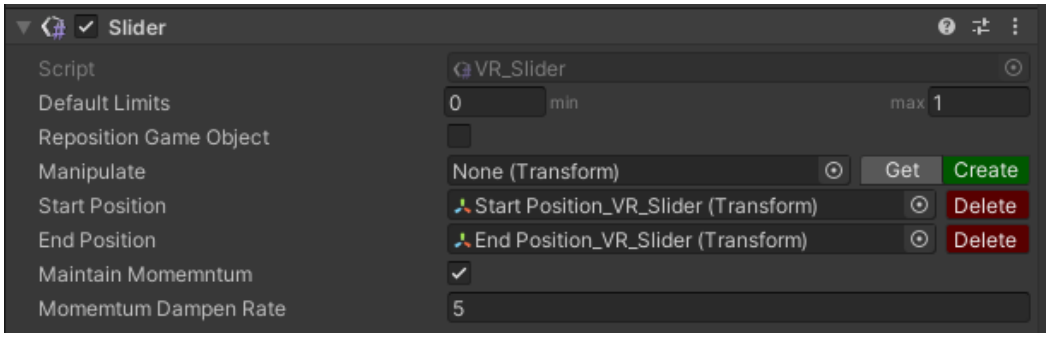


Figure 44. Slider

Properties:

Property	Function
Default Limits	*Matter of further development.*
Reposition Game Object	If enabled, the GameObject slides.
Manipulate	Select the GameObject to manipulate. If null, it is the GameObject owner.
Start Position	Start point transform.
End Position	End point transform.
Maintain Momentum	If enabled, keeps the door moving after you release it.
Momentum Dampen Rate	Speed at which the door keeps moving after you release it.

2.9.3 Enabling System

These components control the enabling/disabling system of other components.
In this section you will learn about the following components:

- Interaction Controllers Group
- Interactive Behaviors Group
- Door Group
- Group Activator
- Random Activator
- On Enable Event
- Shortcut Key

2.9.3.1 *Interaction Controllers Group*

The Interaction Controllers Group component creates a group of GameObjects that have the Interaction Controller component, as well as controls them together.

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

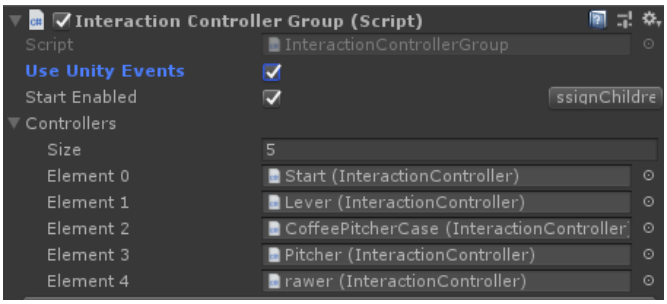


Figure 45. Interaction Controllers Group

Properties:

Property	Function
Use Unity Events	If enabled, all Interaction Components in the list are enabled. If disabled, the group can be controlled only from the procedure.
Start Enabled	If enabled, Interaction Controllers of the list are activated when you play back the scene.
Assign Children	To search for the children with the Interaction Controller component and add them to the group, click Assign Children.
Controllers:	
Size	Number of GameObjects in the group.
Element	GameObject in the group.

2.9.3.2 *Interactive Behaviors Group*

The Interactive Behaviors Group component creates a group of GameObjects that have the Behavior component, as well as enables and disables them at the same time.

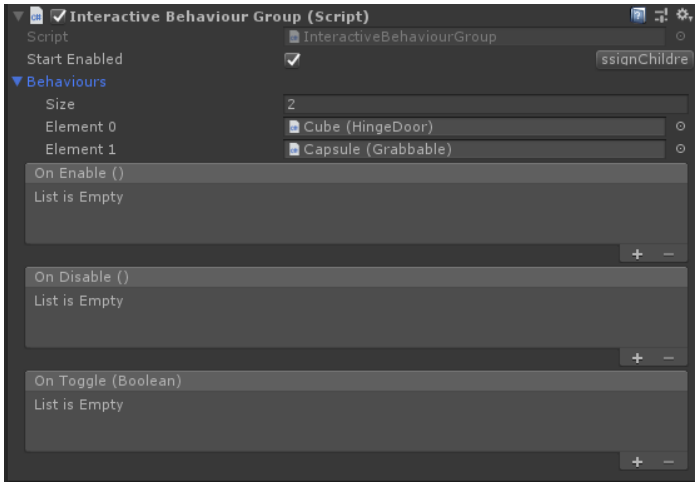


Figure 46. Interactive Behaviors Group

Properties:

Property	Function
Start Enabled	If enabled, the attached components are activated when you play back the scene.
Assign Children	To search for the children with the Behavior component and add them to the group, click Assign Children.
Behaviors:	
Size	Number of GameObjects in the group.
Element	GameObject in the group.

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

2.9.3.3 *Door Group*

The Door Group component creates a group of GameObjects that have the Door component (e.i. Slide Door), as well as enables and disables them at the same time.

Figure 47. Door Group

Properties:

Property	Function
Door Count	Number of doors.
Reload	To search for the children with the Door component and add them to the group, click Reload.
Doors:	
Size	Number of GameObjects in the group.
Element	GameObject of the group.

2.9.3.4 *Group Activator*

The Group Activator component controls a list of GameObjects that will be enabled in a scene one at a time (by event or procedure). The active GameObject can be called directly, or by following the list order in a cycle.

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

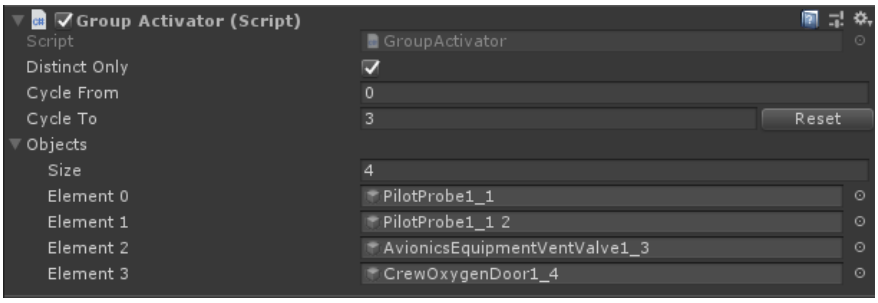


Figure 48. Group Activator

Properties:

Property	Function
Distinct Only	If enabled, GameObjects cannot be listed more than once.
Cycle From	The first Element number of the list that is considered when using the GameObject cycling activation.
Cycle To	The last Element number of the list that is considered when using the GameObject cycling activation.
Objects:	
Size	Number of listed GameObjects.
Element	GameObject to be enabled when disabling other listed objects.

2.9.3.5 *Random Activator*

The Random Activator component randomly activates one listed GameObject, as well as enables one GameObject at a time by following the list order in a cycle.

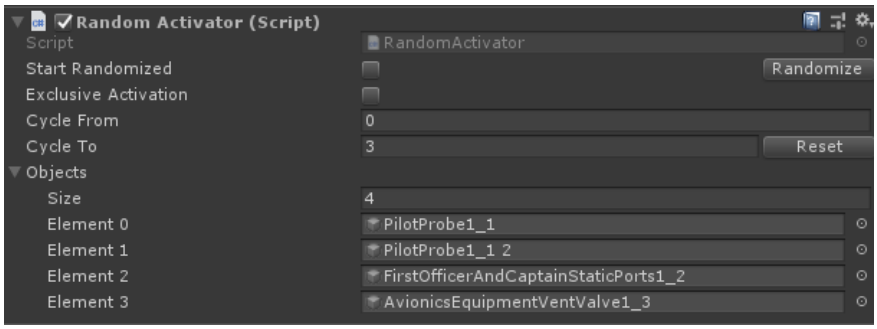


Figure 49. Random Activator

Properties:

Property	Function
Start Randomized	If enabled, one GameObject in the list is activated randomly when the Game is On Play, or the component is activated.
Exclusive Activation	If enabled, all GameObjects in the list that were not activated randomly are disabled.
Cycle From	The first Element number in the list that is considered when using the GameObject cycling activation.
Cycle To	The last Element number in the list that is considered when using the GameObject cycling activation.
Objects:	
Size	Number of listed GameObjects.
Element	GameObject to be randomly enabled.

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

2.9.3.6 *On Enable Event*

The On Enable Event component triggers required Events whenever the GameObject is enabled or disabled.

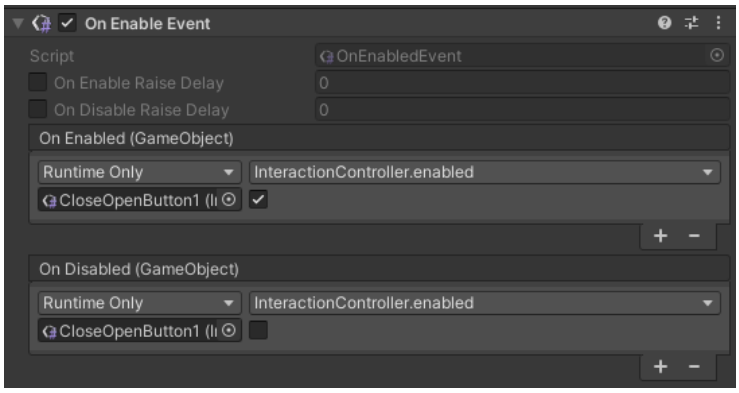


Figure 50. On Enable Event

Properties:

Property	Function
On Enable Raise Delay	If checked, the defined delay [sec] is erased before triggering the event.
On Disable Raise Delay	If checked, the defined delay [sec] is erased before triggering the event.

2.9.3.7 *Shortcut Key*

The Shortcut Key component attaches an event to the target key of the keyboard. Moreover, it allows controlling the text of a canvas whenever the description or the target key changes. This component is useful if the legend of the event command is attached to a canvas.

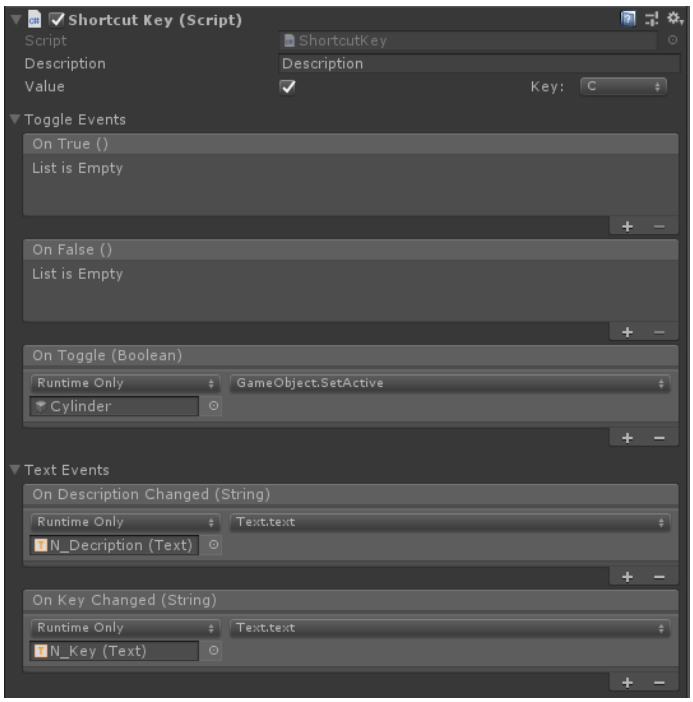


Figure 51. Shortcut Key

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

Properties:

Property	Function
Description	Description of an event that is shown when the Text Events property involves a the control of a canvas text.
Value	Boolean value that is read by the Toggle Events property.
Key	Select the keyboard key to control the event.

The example on Figure 51 displays how the Cylinder GameObject is enabled and disabled whenever you press the key N on the keyboard. Moreover, the N_Description and N_Key texts change automatically with the Description and Key parameters.

2.9.4 Canvas Editing

These components are related to the Canvas objects in a scene.
In this section you will learn about the following components:

- Propagate Texts
- Value to String
- Procedures Dispatcher

2.9.4.1 *Propagate Texts*

The Propagate Texts component overwrites the text of the Text Components property.

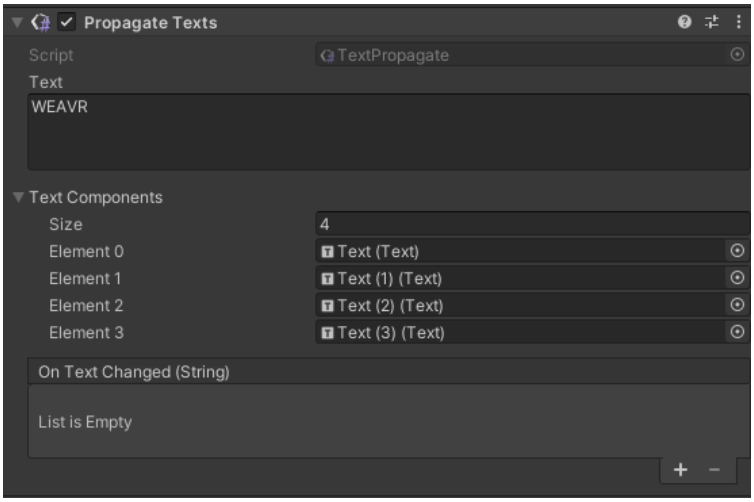


Figure 52. Propagate Texts

Properties:

Property	Function
Text	Text that is shared with the listed Text components.
Text Components:	
Size	Number of Text Components.
Element	Text Component to overwrite.

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

2.9.4.2 *Value to String*

The Value to String component allows printing the indicated string on a GameObject text component.

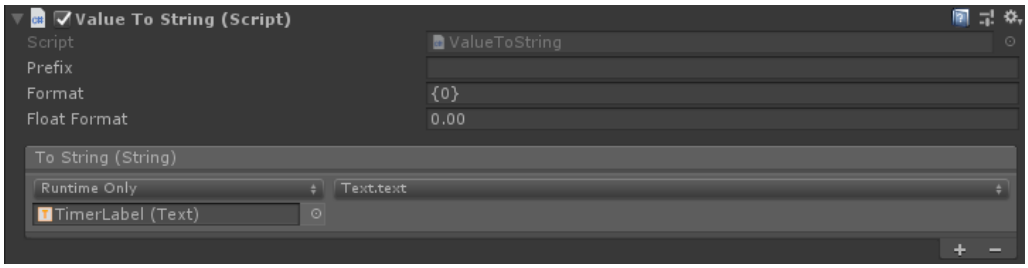


Figure 53. Value to String

Properties:

Property	Function
Prefix	Define a string that is used as a prefix of the floating string text.
Format	Define how to format the incoming GameObject texts. For example, "{0}" shows the first text parameter of the incoming GameObject in quotes, while "{0}" {1}, shows the first text parameter of the incoming GameObject in quotes followed by the second text parameter).
Float Format	Define how to format float numbers.

Figure 53 shows how to assign the text to change (the Timer Label text in the example) from the To String event.

2.9.4.3 *Procedures Dispatcher*

The Procedures Dispatcher component displays the current step information (for example, number, title, and description) on Canvas. This component is strictly related to the steps of the procedure that is running Further events related to the procedure flow can be used.

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

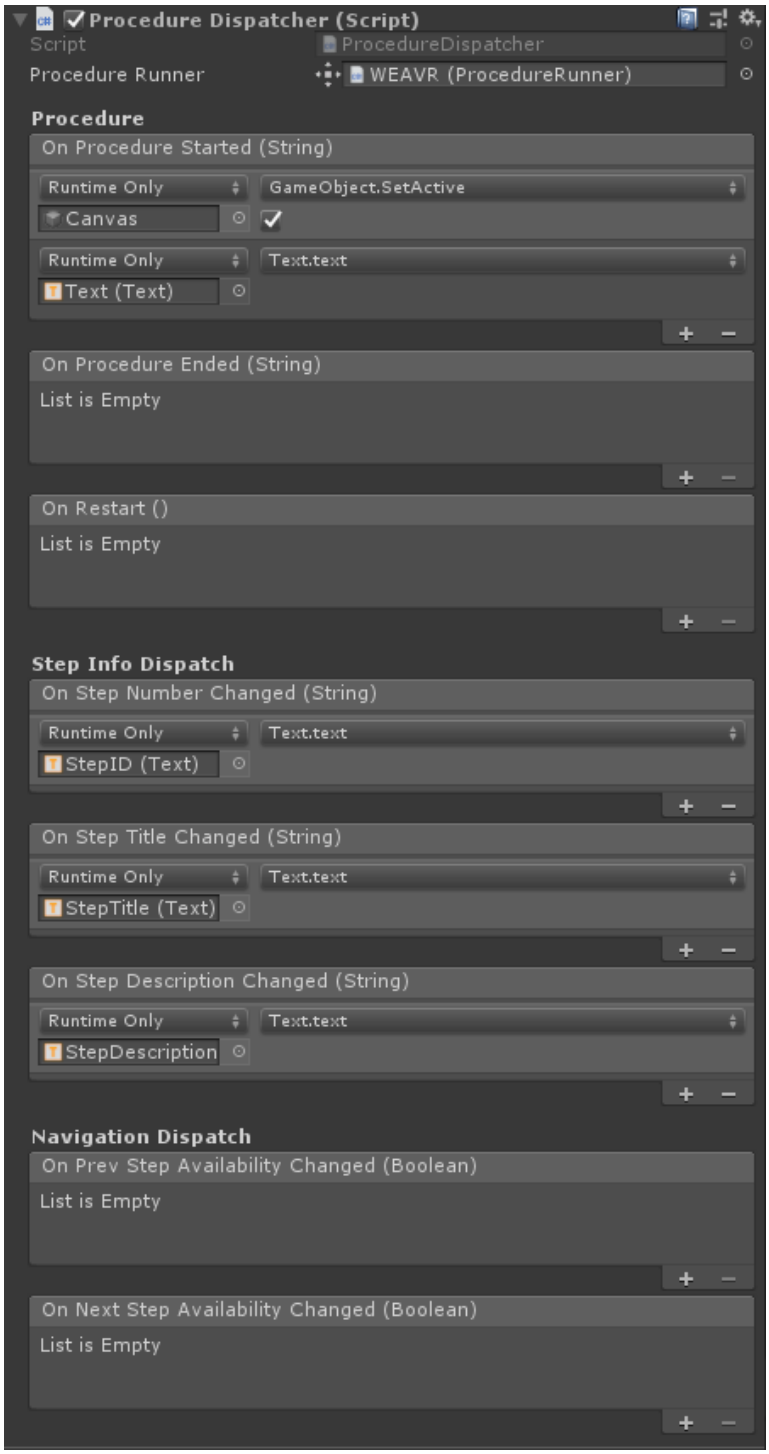


Figure 54. Procedures Dispatcher

Properties:

Property	Function
Procedure Runner	GameObject with the Procedure Runner component. This object is created when you set up the scene.

2.9.5 Utilities

Utilities are components that add features to the scene.

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

In this section you will learn about the following components:

- Timing:
  - Generic Timer
  - Scene Time
  - Timer Formatter
  - Time Text
- Scoring System:
  - Counter
- Advanced Text Editing:
  - Selectable Label
  - Text Editor

2.9.5.1 *Timing*

2.9.5.1.1 Generic Timer

The Generic Timer component tracks the passing time and, if required, triggers events as the time goes by.

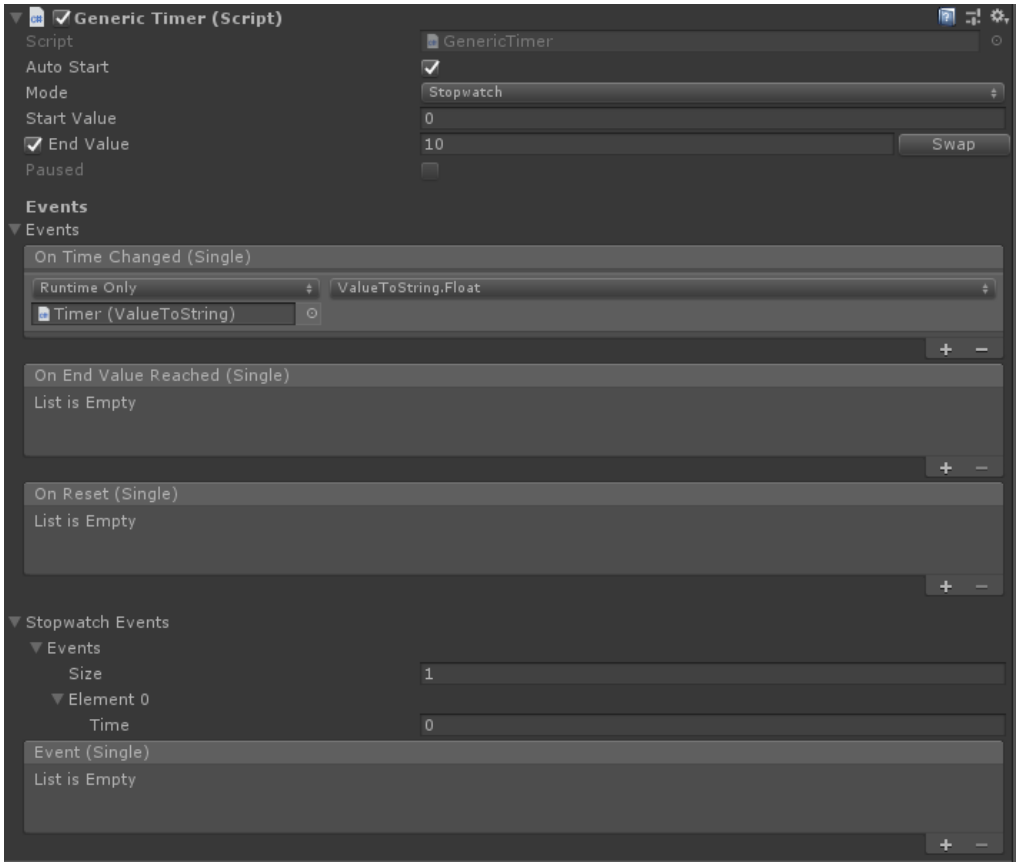


Figure 55. Generic Timer

Properties:

Property	Function
Auto Start	If enabled, the timer enters the Play mode.
Mode:	
Timer	Triggers an event only when the End Value is Reached.
Stopwatch	Triggers events when it reaches indicated time values.
Text Mesh	
Start Value	Timer start value.

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

End Value	If enabled, timer end value. To switch the start and end values, click Swap. Both modalities allow tracking the passing time by incrementing or decrementing it.
Paused	Debug parameter that shows if the timer is paused.
Stopwatch Events:	
Size	Number of time values triggering an event.
Time	Time value.

Figure 55 shows how to print the timer value on text in the scene using the Value To String component.

> NOTE
> To visualize the time on text, a formatting component is mandatory: Value To String, Timer Formatter, or Time Text.

2.9.5.1.2 Scene Time

The Scene Time component traces the passing time according to the selected Time Type.

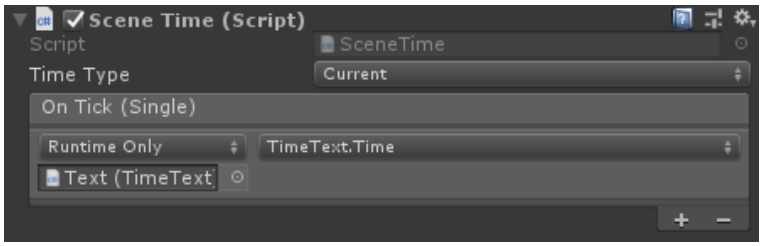


Figure 56. Scene Time

Properties:

Property	Function
SceneTime:	Reference time to be traced.
Component	Time starts when this component is activated.
Scene	Time starts when the scene is loaded.
Application	Time starts when entering the Play mode.
Current	it is the current daytime

2.9.5.1.3 Timer Formatter

The Timer Formatter component is used with the Generic Timer component to display time in the defined format on the listed texts. Furthermore, this component expands the existing Events that check the passing time.

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

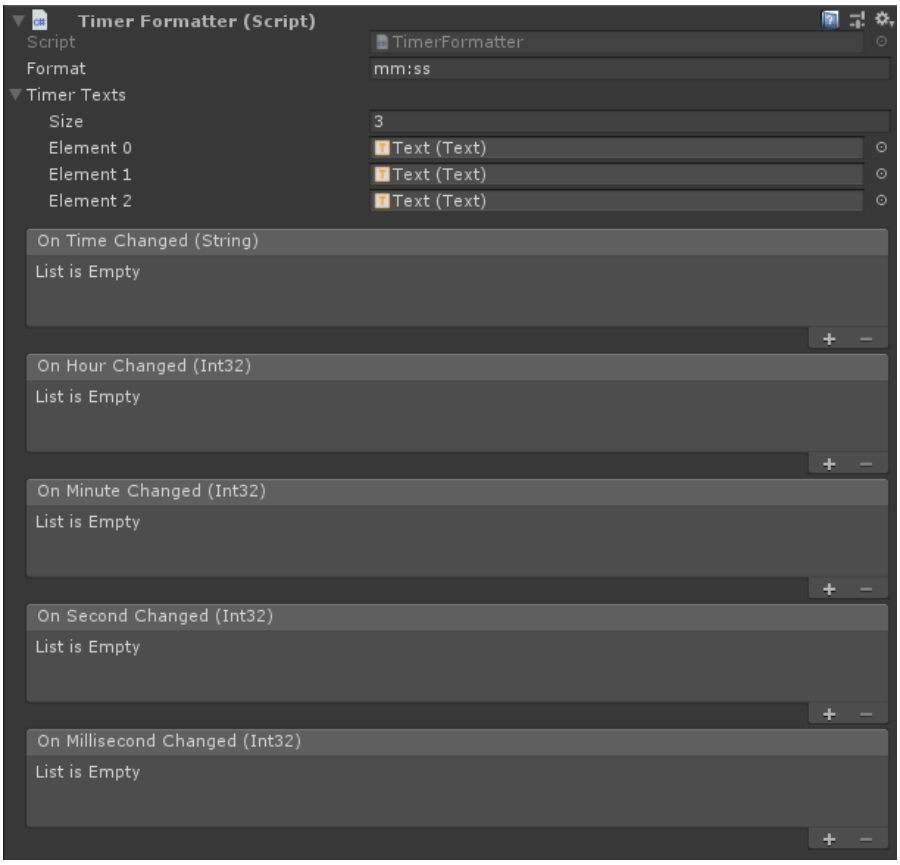


Figure 57. Timer Formatter

Properties:

Property	Function
Format	Enter the needed time format: "HH" for 24h, "hh" for 12h, "mm" for minutes, "ss" for seconds, and "uu" for milliseconds.
Timer Texts:	
Size	Number of Texts.
Element	GameObject owner of the Text used as a timer.

NOTE
If the Generic Timer and Timer Formatter components are in the same GameObject, they synchronize without using an event.

2.9.5.1.4 Time Text

The Time Text component is used together with the Generic Timer and Text components to change the text colour when the Target time property is reached.

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

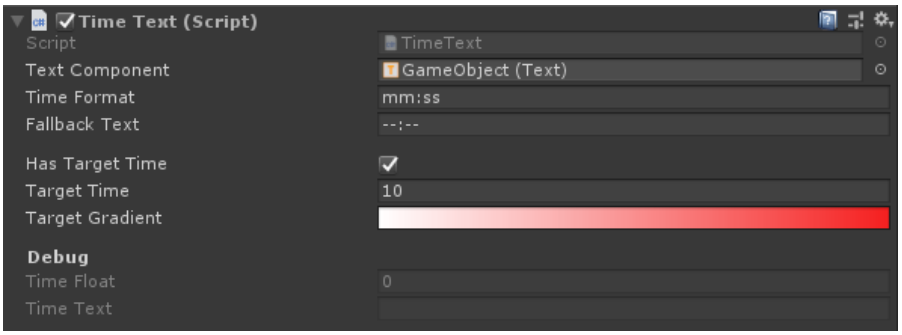


Figure 58. Time Text

Properties:

Property	Function
Text Component	Text displaying the time.
Time Format	Time format: "HH" for 24h, "hh" for 12h, "mm" for minutes, "ss" for seconds, and "uu" for milliseconds.
Fallback Text	Defaoult text if Time Text is disabled.
Has Target Time	If enabled,
Target Time	Target time (in seconds).
Target Gradient	Text colour before and after reaching the target.

2.9.5.2 *Scoring System*

2.9.5.2.1 Counter

The Counter component is used as a counter that increments or decrements by an event or procedure whenever the required condition occurs.

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

Figure 59. Counter

Porperties:

Property	Function
Value	Current value of a counter.
Reset Value	Reset value of a counter.
Limits	If enabled, enter the minimum and maximum counter values.

2.9.5.3 *Advanced Text Editing*

2.9.5.3.1 Selectable Label

The Selectable Label component allows you to apply text background, define text and background colours, as well as change them when the text is selected.

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

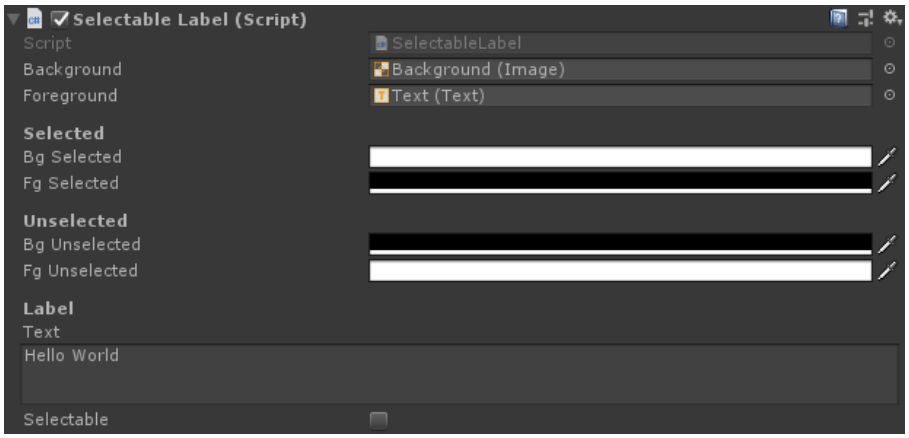


Figure 60. Selectable Label

Properties:

Property	Function
Background	Background image for the text.
Foreground	Text to edit.
Selected:	Colors when the Text Editor is in a Selected mode.
Bg Selected	Background color.
Fg Selected	Foreground color.
Unselected:	Colors when the Text Editor is in an Unselected mode.
Bg Unselected	Background color.
Fg Unselected	Foreground color.
Text	Text to display.
Selectable	If enabled, the text can be selected and so change the background and foreground colors.

2.9.5.3.2 Text Editor

The Text Editor component allows overwriting Text according to the input that is given in play. The Text Editor component can check the text format and trigger an event whenever the input is not allowed. This component includes properties of the Selectable Label component.

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

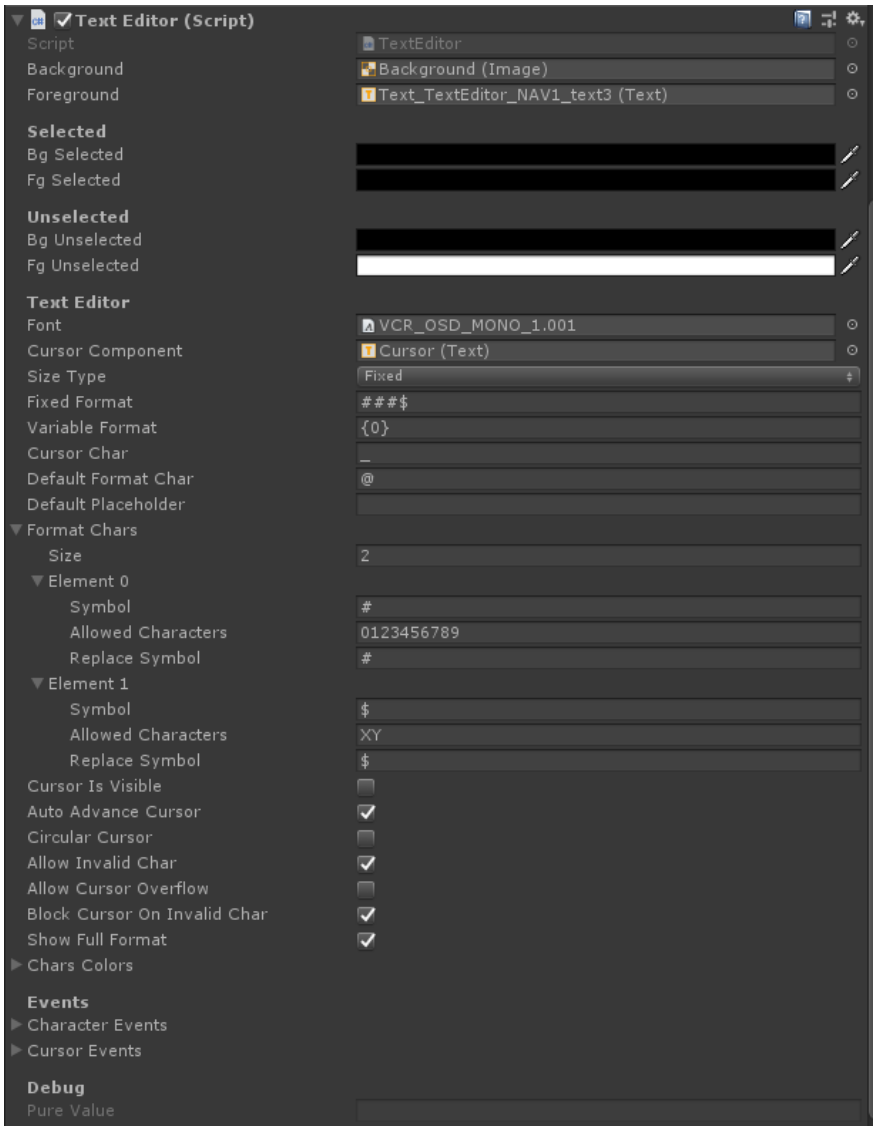


Figure 61.Text Editor

Properties:

Property	Function
Background	Background image for the text.
Foreground	Text to edit.
Selected:	Colors when the Text Editor is in a Selected mode.
Bg Selected	Background color.
Fg Selected	Foreground color.
Unselected:	Colors when the Text Editor is in an Unselected mode.
Bg Unselected	Background color.
Fg Unselected	Foreground color.
Font	Font of the text to edit.
Cursor Component	Text cursor that indicates the letter to be edited.
Size Type	If Fixed, the text size is fixed and follows the Fixed Format property. If Variable, the text size is variable and follows the Variable Format property.
Fixed Format	Text format that uses the Default Format Char and Format Chars properties of the component.
Variable Format	Text format that follows the basic format rules.
Cursor Char	Character that is used for the text cursor.

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

Default Format Char	Symbol that is used in the Fixed Format property to allow any kind of character.
Default Placeholder	Fallback character in case the Text character is not edited.
Format Chars:	Additional settings to customize characters.
Size	Number of customized format characters.
Element:	
Symbol	Symbol that is used for the format.
Allowed Characters	Characters that are allowed for this specific Format Char.
Replace Symbol	Character that is if you insert a character that is not allowed.
Cursor Is Visible	If enabled, the cursor text is visible in a scene.
Auto Advance Cursor	If enabled, the cursor advances to the next character when you insert the current one.
Circular Cursor	If enabled, the cursor goes back to the first character automatically when reaching the last one.
Allow Invalid Char	If enabled, invalid characters are allowed.
Allow Cursor Overflow	If enabled, the cursor continues advancing the last format character.
Block Cursor On Invalid Char	If enabled, the cursor does not advance when an invalid character is inserted.
Show Full Format	If enabled, always shows full format of characters when OnStart.
Chars Colors	Matter of further development.

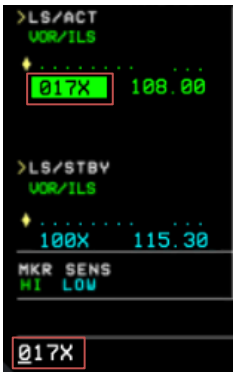


Figure 62. Text Editor example

Figure 62 shows an example of the Text Editor component. The text at the bottom is set by the Text Editor properties of Figure 61. To insert the character underlined by the cursor, you can use the "TextEditor.Insert" method as an event or procedure action. You can then copy this text to the selected text in green using the "TextEditor.CopyFormattedTo" method.

2.9.6 Animator and Animations

These components allow you to add features to Animator and Animations in a scene.
In this section, you will learn about the following components:

- Animator Group
- Animation Float Event
- Animate With Float
- Float Event Forwarder

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

2.9.6.1 *Animator Group*

The Animator Group component allows you to control all animators listed within one action by a component event or a procedure action.

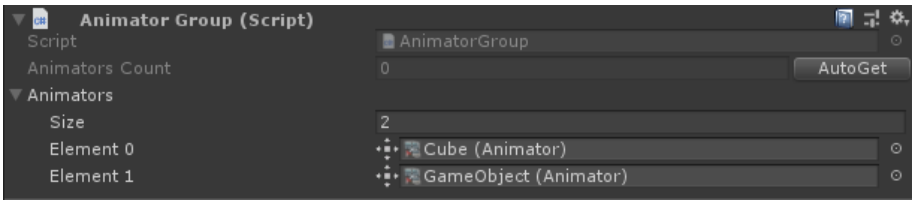


Figure 63. Animator Group

Properties:

Property	Function
Animators Count	For debug purpose.
AutoGet	If clicked, the children's Animator components are added to the list of Animators.
Animators:	
Size	Number of Animators.
Element	Controlled Animator.

2.9.6.2 *Animation Float Event*

The Animation Float Event component allows playing the target animation clip in unison with the behavior progress when attached to the Hinge Door or Slide Door behaviors .

Figure 64. Animation Float Event

Properties:

Property	Function
Target	GameObject to animate.
Animation Clip	Animation to be executed.
Start Point Perc	Starting point of an animation (in percentage).

To attach the Animation Float Event to the Hinge Door or Slide Door behaviors, its PlayAnimation function should be subject of the "On Door Open Progress" Advanced Event (Figure 65).

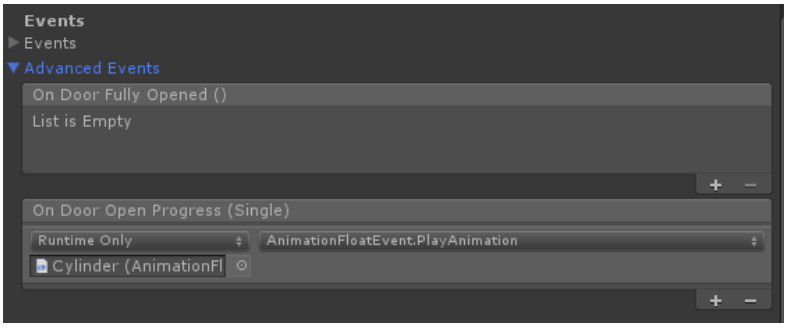


Figure 65. Animation Float Event attached to a Behavior

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

2.9.6.3 *Animate With Float*

The Animate With Float component plays the target animation clip in unison with a Behavior progress movement. This is the advanced version of Animation Float Event (Section 2.9.6.2).

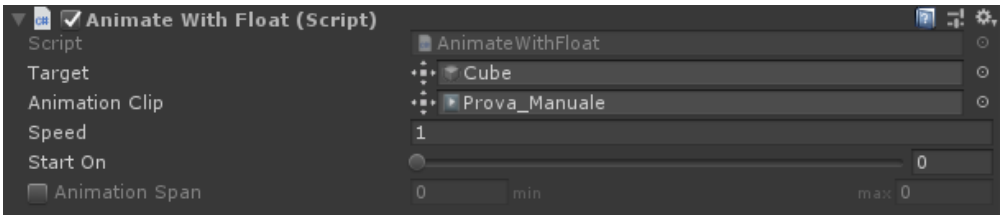


Figure 66. Animate With Float

Properties:

Property	Function
Target	GameObject to animate.
Animation Clip	Animation to be executed.
Speed	Speed of playing the animation.
Start On	Animation starting point in percentage. If the Animation Span property is enabled, the starting point depends on the defined span.
Animation Span	If enabled, the animation range of frames to be executed.

2.9.6.4 *Float Event Forwarder*

The Float Event Forwarder component is an extension of the Animation Float Event or Animate With Float components . When it is attached by Event (example in Figure 67) to the Animation Float Event, this component allows you to control the Animation Clip progress by frame. This is useful for Behaviors such as Push Buttons where the Animation Clip progress depends on how much time the specific position is kept.

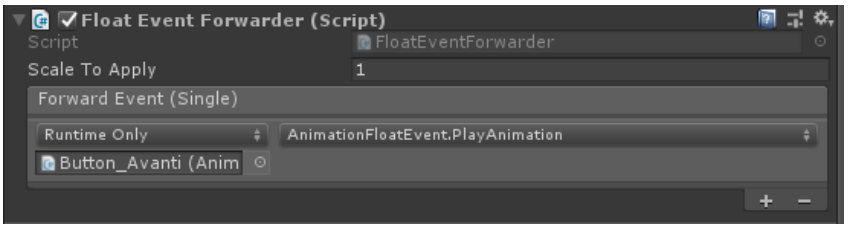


Figure 67. Float Event Forwarder

Properties:

Property	Function
Scale to Apply	Animation according to the frame increment/decrement that is defined by the event (Figure 68).

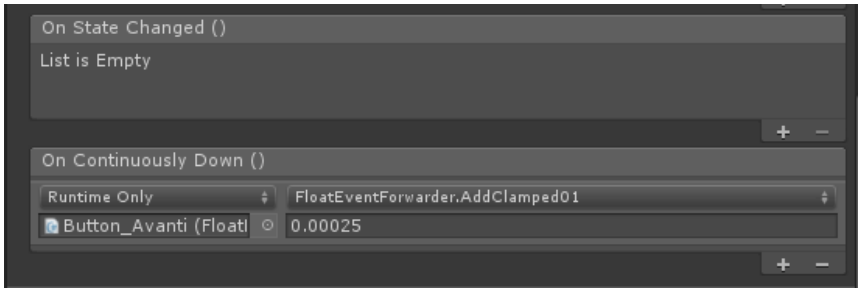


Figure 68. Float Event Forwarder attached to a Behaviour

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

2.9.7 Scene

These components are related tovisualizing GameObjects in a scene.
In this section, you will learn about the following components:

- LED Feed
- Override Outline
- Popup Point

2.9.7.1 *LED Feed*

The LED Feed component changes the texture color and emissivity, as a LED, of the indicated GameObject. It can be controlled using Events of other components or through the WEAVR procedure.

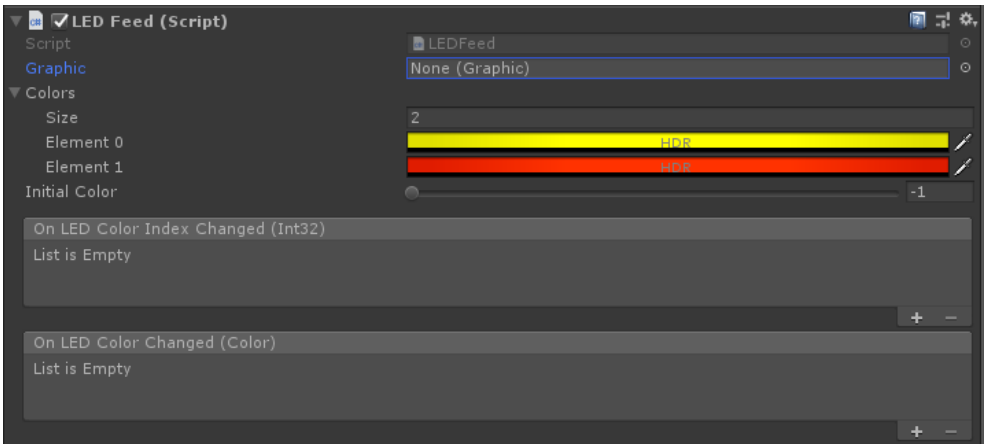


Figure 69. LED Feed

Properties:

Property	Function
Graphic	If null, the texture of the GameObject owner of the LED Feed script changes. Otherwise, graphic such as texts or canvas components can change.
Colors:	Emissivity color.
Size	Number of available colors.
Element	Available color.
Initial Color	Color of the component when you switch it on. The default texture is –1.

2.9.7.2 *Override Outline*

The Override Outline component overrides the Render Mode parameter of the Border Outliner component of the VR_RIG. The component controls which objects are outlined when you hover over the owner GameObject, or when the object is outlined using the Outline actions.

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

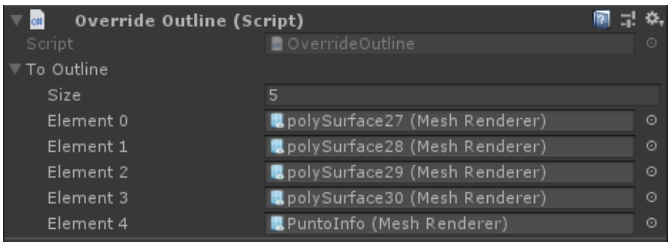


Figure 70. Override Outline

Properties:

Property	Function
To Outline:	
Size	Number of meshes to outline.
Element	Mesh to outline.

2.9.7.3 *Popup Point*

The Popup Point component controls the location point of the billboard listing available interactions with the object in case of non-immersive procedures or more than one available interaction (e.i. A GameObject has both the Grabbable and Executable components. When you click the object, the list of available interactions appears). If this component is not used, the billboard appears in a location that is set automatically.
In case of VR procedures, the Popup Point component overrides the location that was set by the Show Billboard action. Additionally, it is useful when the target object is set by a Variable.

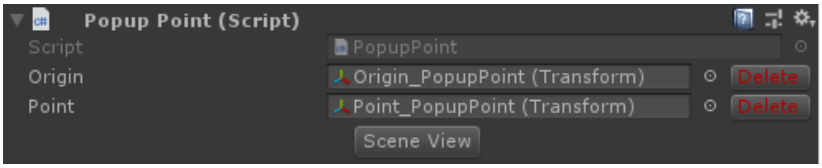


Figure 71. Popup Point

Properties:

Property	Function
Origin	Origin of the line connecting the object and the billboard. To accept the transform that was already created, click Get. To create a transform in the center of the object, click Create.
Point	Transform points where the billboard should appear. To accept the transform that was already created, click Get. To create a transform in the center of the object, click Create.
Scene View	If enabled, the Origin and Point properties are shown in the scene.

2.9.8 Variables

These components that involve the variable system can be used in a scene without a running procedure. Read about Actions and Exit Conditions that involve the use of the variable system. All the Variables that are created in a scene and procedures are stored in the Variable window.
In this section you will learn about the following components:

- Set Global Value
- Query Global Value

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

- Global Value Component
- Data Container

2.9.8.1 *Set Global Value*

The Set Global Value component creates a variable.

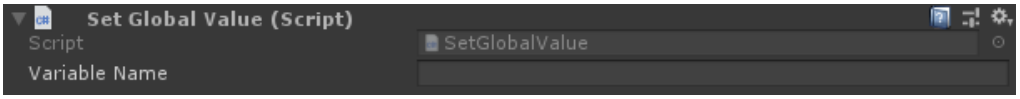


Figure 72. Set Global Value

Properties:

Property	Function
Variable Name	Name of a variable.

2.9.8.2 *Query Global Value*

The Query Global Value component allows querying a variable value. An event is triggered when the variable equals the indicated value. The Figure 73. Query Global Value, shows how an event is raised when the variable reaches the target value 10.

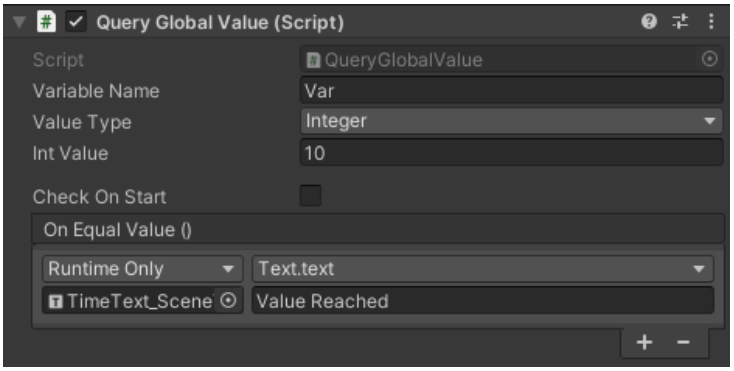


Figure 73. Query Global Value

Properties:

Property	Function
Variable Name	Name of a variable.
Value Type	Type of a variable.
[…] Value	This value depends on the Value Type property. Define the number that the value should be equal to trigger the event.
Check on Start	If enabled, it is checked on start if the variable is set.

2.9.8.3 *Global Value Component*

The Global Value Component creates a variable and interrogates it. An event is raised whenever the variable value changes, and when a target value is reached.

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

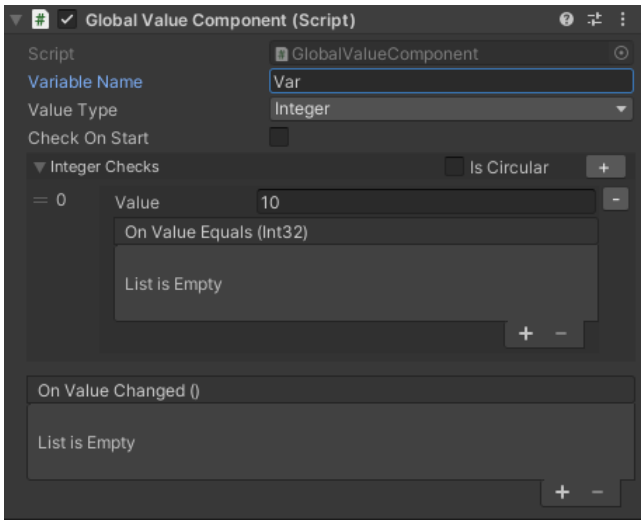


Figure 74.Global Value Component

Properties:

Property	Function
Variable Name	Name of a variable.
Value Type	Type of a variable.
Check on Start	If enabled, it is checked on start if the component is interrogated.
[…] Checks	Target values and its Event are checked when the target is reached.

2.9.8.4 *Data Container*

The Data Container component is a storage of data of different types: Booleans, Integers, Floats, and Strings. An index for each type takes trace of the selected element that can be incremented, decremented, or manually changed by the event or in the procedure.

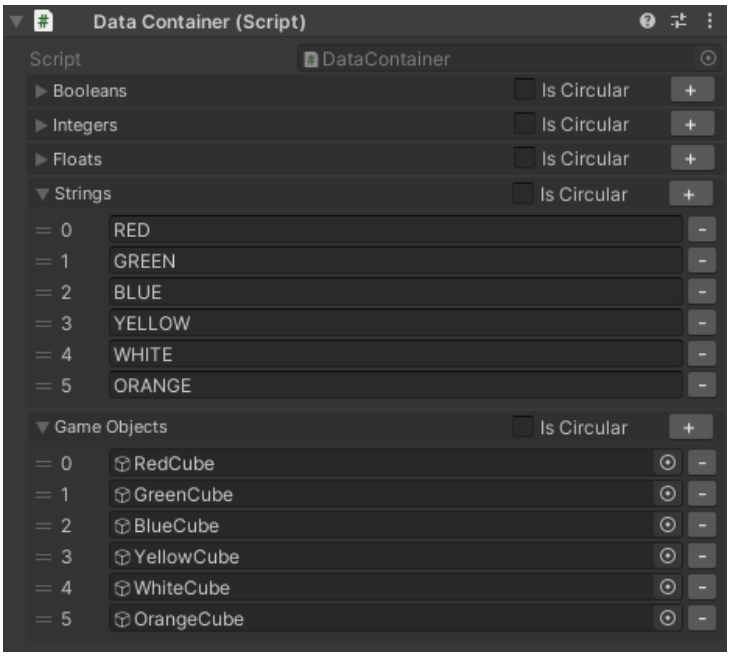


Figure 75. Data Container

Properties:

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

Property	Function
Is Circular	If enabled, the current index restarts from 0 (if incremented) when it reaches the last listed element.

2.9.9 Impact System

These components allow you to add features to Collider impact on each other.
In this section, you will learn about the following components:

- Impact Object
- Object Material

2.9.9.1 *Impact Object*

The Impact Object component manages the Object Material component. It is necessary for the impact system to work.

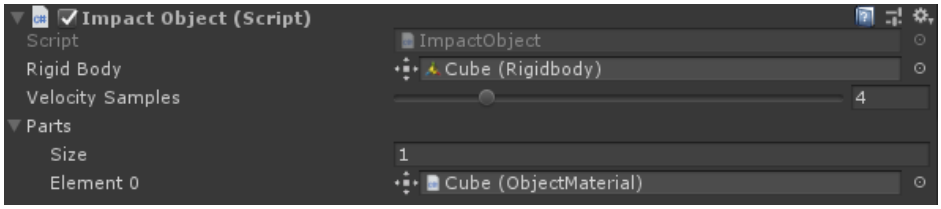


Figure 76. Impact Object

Properties:

Property	Function
Rigid Body	Rigid Body component.
Velocity Samples	When the Rigid Body property is kinematic, this is the fallback value that is used for the impact.
Parts:	
Size	Number of Parts.
Element	Object Material and Terrain Material components.

2.9.9.2 *Object Material*

The Object Material component assigns a behaviour in terms of sound and physics to the list of colliders when they hit other colliders. The component should be referred by the Impact Object component in a scene.

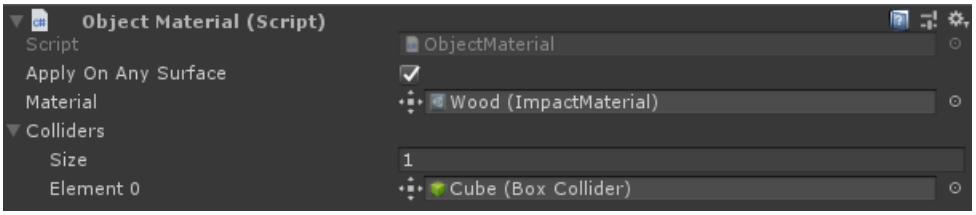


Figure 77. Object Material

Properties:

Property	Function

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

Apply on Every Surface	If enabled, it behaves with any collider. Otherwise, it behaves only with other Object Materials.
Material	Impact Material that is assigned to colliders.
Colliders:	
Size	Number of colliders.
Element	Collider with the Impact Material behaviour.

NOTE
You can find predefined Impact Objects in the folder:
WEAVR/Essential/Assets/ImpactMaterials

2.10 Copy Components Tool

The Copy Components tool manages automatic copying of components between two hierarchies of objects. The tool attempts to find similarities in the From Object and To Object hierarchies and automatically link them. You can edit and delete automatic links as well as create new links. The tool allows you to create new objects in the To Object hierarchy to match objects in the From Object hierarchy.
To open the Copy Components tool, go to WEAVR > Utilities > Copy Components.

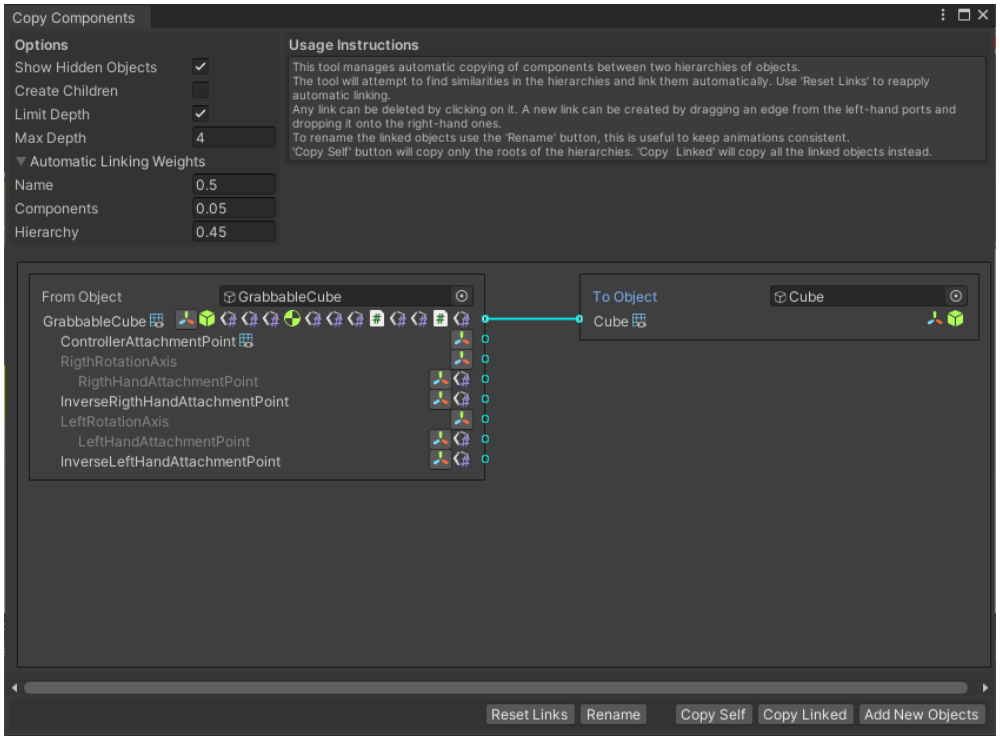


Figure 78. Copy Components

Properties:

Property	Function
Show Hidden Objects	If enabled, hidden objects that are created by the component are shown in the From Object property.

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

Create Children	If enabled, new GameObjects are created in the hierarchy to match children.
Limit Depth	If enabled, the depth of children that are copied to the hierarchy is limited.
Max Depth	Maximum depth of children that are copied.
Automatic Linking Weights:	Algorithm weights that are used to generate automatic links.
Name	Weight for a name match.
Components	Weight for a component match.
Hierarchy	Weight for a hierarchy match.
Reset Links	To reapply automatic linking, click Reset Links.
Rename	To rename the hierarchy in the To Object according to the From Object, click Rename.
Copy Self	To copy only roots of hierarchies, click Copy Self.
Copy Linked	To copy linked objects, click Copy Linked.
Add New Objects	To add new GameObjects to match the hierarchies, click Add New Objects.

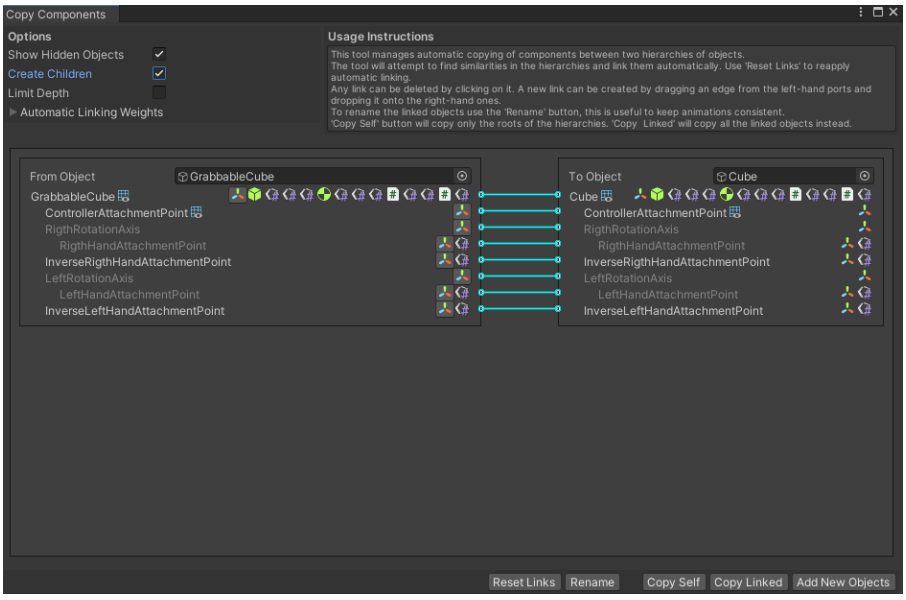


Figure 79. Copy Components, matched hierarchies

Figure 79 shows the hierarchy in the To Object after it was matched with the From Object one using the Copy Components tool.

2.11 VR Standard Components

Steam VR components are used to allow the immersivity in a scene. They are enabled if you configure the VR immersive Setup. This section lists Prefab objects that involve Steam VR components that are useful for the project.
In this section, you will learn about the following components:

- VR_RIG
- Teleport Point

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

- Teleport Area
- Hands Poses

2.11.1.1 *VR_RIG*

The VR_RIG component controls the player movement and interactions in the scene. It is included in the scene Hierarchy by default after the setup. .
VR_RIG involves other components that are related to the player's ability to teleport in the scene and interact with Objects through controllers/hands; furthermore, the component includes the VR camera that allows the full immersivity.

> WARNING
> It is recommended not to modify the VR_RIG settings.

2.11.1.2 *Teleport Point*

To teleport, the scene should include a valid area that allows it. You should place the Teleport Point object above the ground so that it is a valid point in the scene to teleport to.
The WEAVR folder in the Project window contains Teleport Point prefab including the Teleport Point component and the default mesh for this object.
The containing folder is: *WEAVR > Essentials > Assets > Prefabs > VR > TeleportPoint*

The Teleport Point component controls the object behavior.

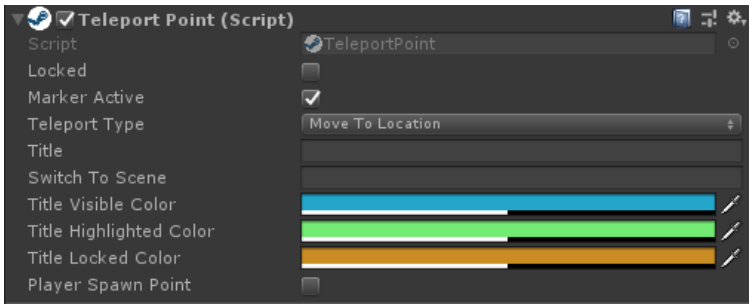


Figure 80. Teleport Point

Properties:

Property	Function
Locked	If enabled, teleporting in this area is not allowed because it is locked.
Marker Active	If enabled, the teleport area is activated.
Teleport Type:	Actions that the teleport allows to execute.
Move to Location	Moves the player location to the location of the teleport point.
Switch to New Scene	Loads a new scene.
Title	Text that is visualized above the teleport point.
Switch to Scene	Name of the scene to be loaded.
Title Visible Color	Color of the Teleport Point when it is visible.
Title Highlighted Color	Color of the Teleport Point when it is highlighted.
Title Locked Color	Color of the Teleport Point when it is locked.
Player Spawn Point	If enabled, the point where the player is located when the scene starts.

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

2.11.1.3 *Teleport Area*

To teleport, the scene should include a valid area that allows it. You should place the Teleport Area object above the ground so that it is a valid area in the scene to teleport to.
The teleport areas dimension is controlled with the transform scale property of the object.
The WEAVR folder in the Project window contains the prefab including the Teleport Area component and the default mesh for this object.
The containing folder is: *WEAVR > Essentials > Assets > Prefabs > VR > TeleportArea*

The Teleport Area component controls the object behavior.

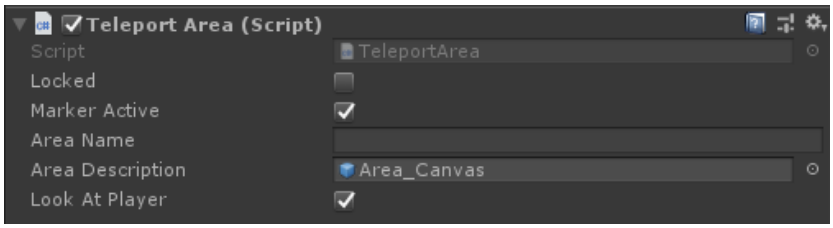


Figure 81. Teleport Area

Properties:

Property	Function
Locked	If enabled, teleporting in this area is not allowed because it is locked.
Marker Active	If enabled, the teleport area is activated.
Area Name	Name of the area.
Area Description	Canvas of the teleport area.
Look at Player	If enabled, the area description faces the player position.

2.11.1.4 *Hands Poses*

The VR immersive scene supports the visualization of hand renderers that repeat movements of hands when interacting with the objects in a scene.
Figure 82 shows the hand renderer in two positions: with the controller and without. The hand renderer is displayed with the controller when it is not hovering over an object with the Behavior components (Section 2.9.2) and without the controller when it is hovering over the object.

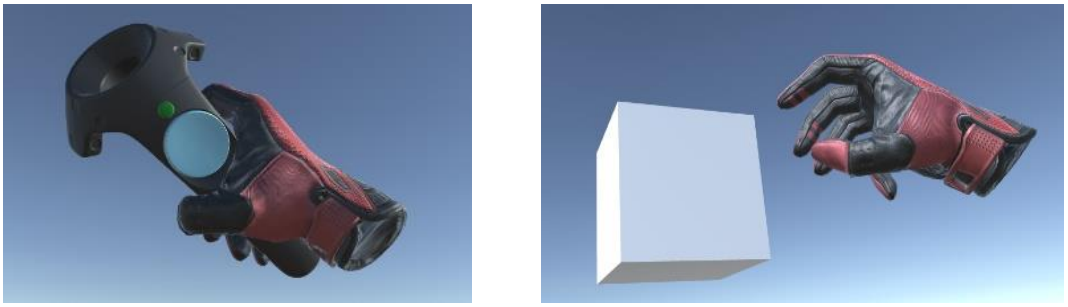


Figure 82. Hand render in VR immersive scene

The hand interaction with objects is not mandatory. You can achieve this visualization when interacting with the object with the help of the Skeleton Poser component.
If the Skeleton Poser component is not attached to the object, hands are hidden when interacting with the object.

Prefabs of Hand Poses are stored in the WEAVR folder:

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

Assets → WEAVR → Resources → VIVE → HandPoses

The default Hand Pose prefabs are:

- ReferencePose_CylindricalOBJ: used to interact with cylindrical objects.
- ReferencePose_SmallOBJ: used to interact with small objects.
- ReferencePose_SmallCylindricalOBJ: used to interact with small cylindrical objects.
- ReferencePose_Horizontal_CylindricalOBJ: used to interact with cylindrical objects.
- ReferencePose_PointSelect: used to point or click a button.
- ReferencePose_HorizontalMoveOBJ: used to move an object horizontally.

2.11.1.4.1 Skeleton Poser

WEAVR Creator allows to:

- Change the default Hand Pose.
- Create a new pose.
- Create a new pose by duplicating the existing prefab, as well as rename and modify the pose as required.
- Change the default hand models.

> NOTE
> It is recommended to save new poses outside of the WEAVR folder in the Project window.

The Skeleton Poser component manages the visualization of Hand Pose when you interact with the object (Section 2.11.1.4) with the help of the controller. The component is referenced by the Behaviour (Section 2.9.2) that is attached to the object (a Behaviour can reference a Hand Pose attached to another GameObject in the scene).
The hand models that are visualized in the scene can be set as desired from the relative VR RIG.

Create or import a Hand Pose

You can create a new Hand Pose, import an existing pose, and much more using the Skeleton Poser component.

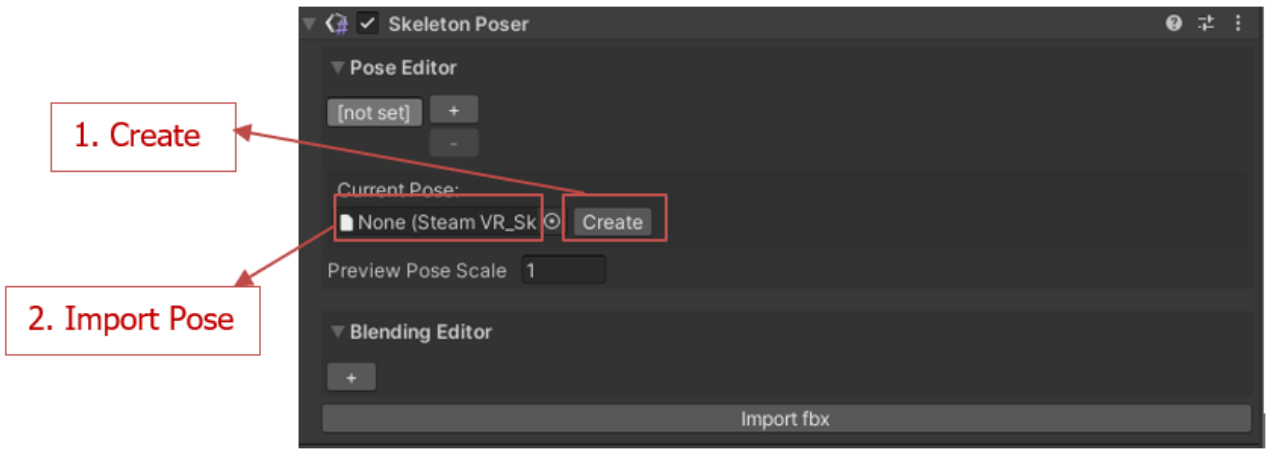


Figure 83. Create a Hand Pose

1. Create a new Hand Pose. Define the path where to create a new pose in the Asset folder of the Project window. It is recommended to save the new Hand Pose outside the WEAVR folder.

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

2. Import Pose. Import an existing pose by dragging the prefab from the Project window to the Skeleton Poser component.

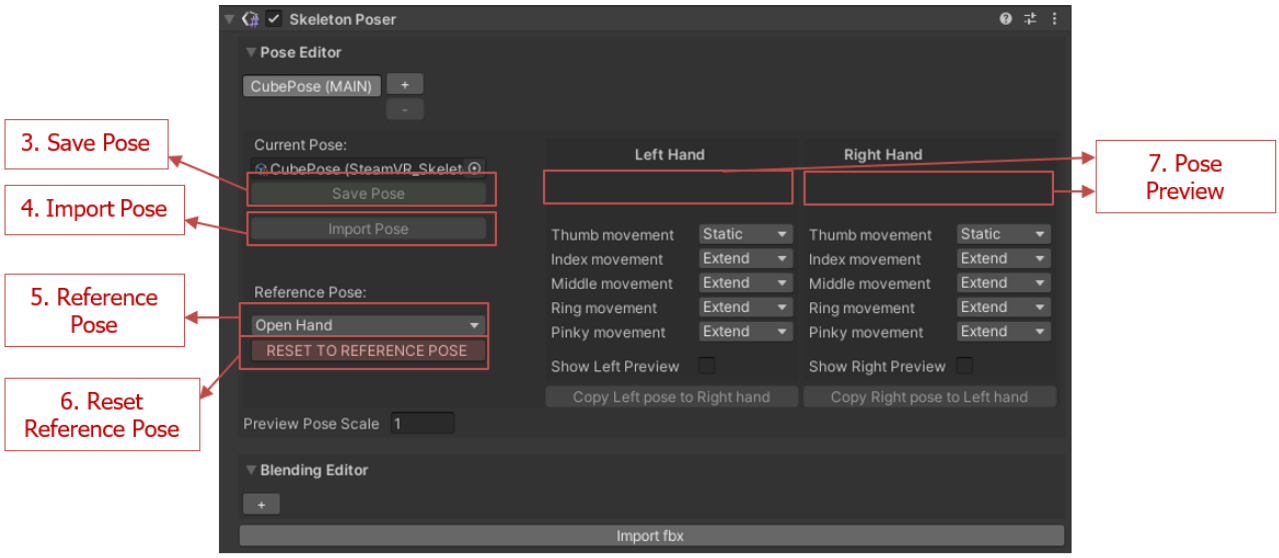


Figure 84. Skeleton Poser

3. Save Pose. Save the modified pose.
4. Import Pose. To visualize a Hand Pose in the scene and associate it with the object, import an existing pose by navigating to the file.
5. Reference Pose. To help the initial modelling of the Hand Pose, reference a pose, click the Reset to Reference Pose button, and then customize the pose.. Available reference poses are:
    - Open Hand
    - Fist
    - Grip Limit

Note: To restart modelling the pose or to overwrite the previous changes, these settings can be used.

Modify a Hand Pose

6. Pose Preview. Pressthe hands icons to highlight them. A preview of the hand pose is visualized in the scene (Figure 85).

Figure 85. Example of Hand Pose preview in the scene

When the Skeleton Poser component is attached to the GameObject, and the pose asset is referenced, children of the object are automatically created (Figure 86). Objects reference the position of the hands (left and right) in the scene and the relative position of each bone of hands that will customize the pose according to their rotation transform.

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

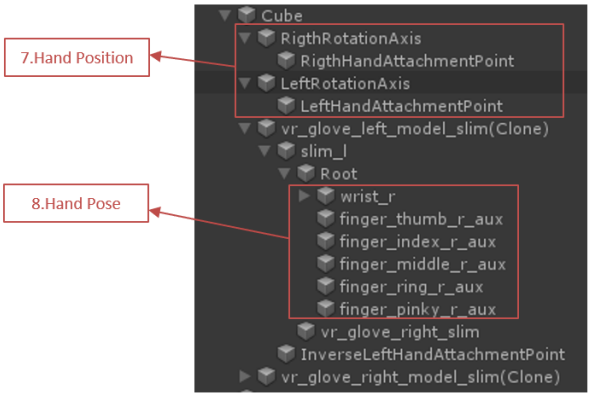


Figure 86. Hand Pose objects in the scene

7. Hand Position. Position of the hand in the scene is controlled by these objects. These are the transform that reference the position of the Hand Pose in respect to the object when interacting with it
    - Rotation Axis (left and right): the hand rotates around this transform when it is set free in the Behaviour that allows it.
    - Attachment Point: the transform of the reference that is used to interact with the object in case of the Grabbable behaviour (Section 2.9.2.1).

8. Hand Pose. These components and their children control the rotation transform and the position of each bone of the hand. Therefore, the hand pose is set according to their transform.

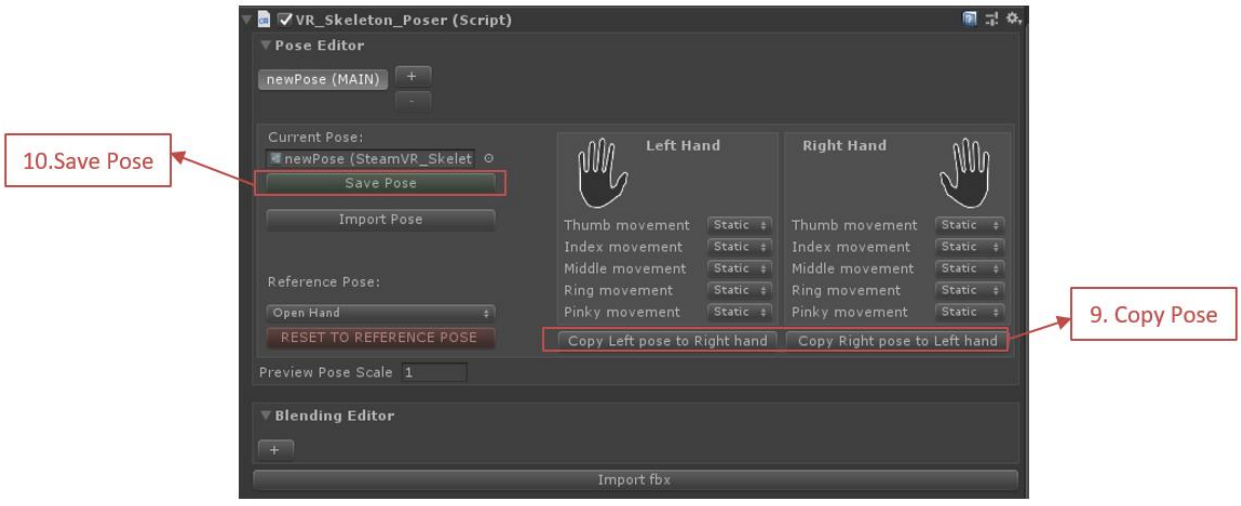


Figure 87. Skeleton Poser

9. Copy Pose. Copy the customized pose of one hand to the other hand by pressing these two buttons.

Save the Hand Pose

10. Save Pose: Save the new pose setting. Note: New settings overwrite the asset settings. If the same pose is referenced in another object, that object pose changes as well.

> NOTE
> The settings that are not explained in this section are not supported by this WEAVR version.

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

2.12 Multiplayer Scene

In case of a multiplayer scene (for scene Setup see Section 2.6), the previous interactions and behaviours should be integrated with specific components that allow scene sharing of multiple players.

2.12.1 Components for Synchronization

2.12.1.1 *Manage Object Ownership*

The Manage Object Ownership component manages sharing of objects in the scene because it controls the ownership of objects among players.

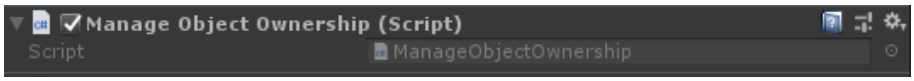


Figure 88. Manage Object Ownership

To manage the ownership for multiplayer scenes, you need to add supplementary components. These components are:

- Photon View
- Photon Transform View
- VR_Object
- Interaction Controller
- Grabbable

NOTE
It is necessary to add one of these components to an object that can be shared in the scene among other players.

2.12.1.2 *Network Outliner*

The Network Outliner component is necessary to synchronize the object outlining among the player scenes.

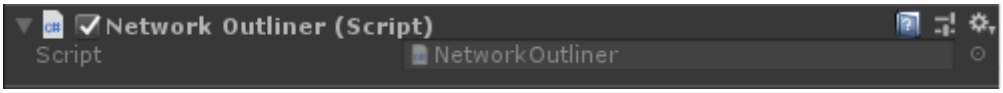


Figure 89. Network Outliner

NOTE
It is necessary to add this component to one GameObject in the scene.

2.12.1.3 *Network Component Toggle*

The Network Component Toggle component is necessary to synchronize components' statuses if components in the scene are toggled by an event, and not by a procedure.

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

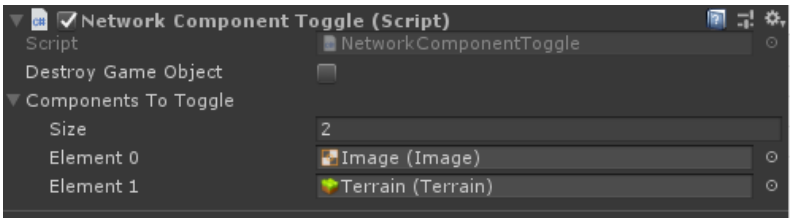


Figure 90. Network Component Toggle

Properties:

Property	Function
Send to New Players	If enabled, sends calls to new players.
Destroy Game Object	If enabled, destroys the GameObject.
Components to Toggle:	
Size	Number of components to synchronize.
Element	Component to synchronize.

2.12.1.4 *Network components*

Every behaviour component should be integrated with its corresponding Network component to share the behaviour-related Events to all players in the scene.
The list of all the behaviours and its Network integrations is the following:

- Grabbable → Network Grabbable
- Connectable → Network Connectable
- Executable → Network Executable
- Operable → Network Operable (matter of further developments)
- Placeable → Network Placeable (matter of further developments)
- Hinge Door → Network Door
- Slide Door → Network Door
- Panel Door → Network Door
- Door Lock → Network Door Lock
- VR_Knob → Network VR_Knob
- Push Button → Network Push Button
- Multi Purpose Push Button → Network Push Button
- Two Way Switch → Network Two Way Switch
- Three Way Switch → Network Three Way Switch
- N Way Switch → Network N Way Switch (matter of further developments)

> NOTE
> If these components are missing, players that do not execute the behaviours will not receive the Event related to the change of the behaviour state.

2.12.2 Setup Multiplayer Application

Before deploying the multiplayer project, you should manage some settings.

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

1. To define the WEAVR Multiplayer settings according to the required Photon settings (see Photon documentation https://doc.photonengine.com), go to *WEAVR > Setup > Settings > Player*

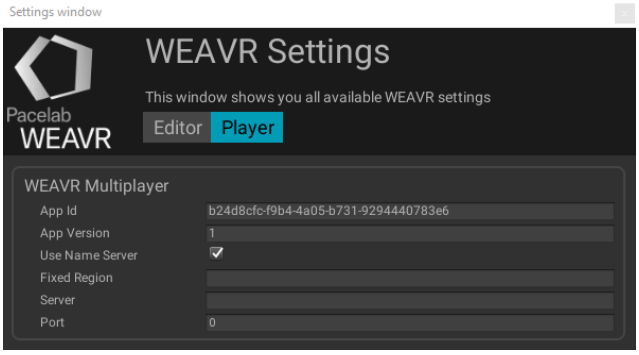


Figure 91. WEVAR settings multiplayer

2. To store the network behaviours that are present in the scene, in the Network RPC Registry component, click Refresh.

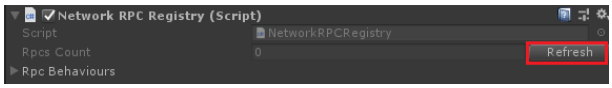


Figure 92. Network RPC Registry

2.13 WEAVR Procedures

This chapter is going to explain how to create a procedure.
Goals:

- Understand basic concepts of a procedure.
- Understand actions.
- Understand conditions.

The **procedure** is composed of steps and the navigation among them. It is graphically created by the user.

Procedure Step is the basic element of a procedure, and it appears as an atomic task to the user when executing the procedure. Every step has a title, number, and description.
Steps can be mandatory or optional. A mandatory step cannot be skipped by the user and is applied only to OpS procedures (see Section 2.13.2.2.1). Each step can be undone, redone, and/or skipped.

The procedure is created once and can be interpreted in different **modalities** by different virtual instructors. Different modalities can vary from procedure to procedure. The default modalities for a Virtual Training procedure are:

- *automatic*: the system automatically executes the procedure while the user remains passive.
- *guided*: the system guides the user throughout the procedure by providing visual and audio hints (e.g.: moves cameras, shows highlights, etc.).
- *feedback*: the user executes a procedure step-by-step, and the system provides a feedback when a step is successfully executed.

The Operations Support (OpS) procedures have only one modality, by default.

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

The system supports the N2M **routing**.
You can define the navigation by adding transitions between steps. The way how you navigate the procedure is based on the evaluation of Boolean expressions, here described as **Exit Conditions** (see 2.13.3.4). These expressions need to be evaluated to **TRUE** to continue to new steps.
The navigation path among steps is defined, and it can be 1:1; n:1, 1:n, n:m according to the number of arrows entering or exiting the step. End steps do not have outgoing arrows. Start steps can have incoming arrows, but they are ignored by the procedure.

2.13.1 Create Procedure

You can configure the creation and the first settings of the procedure in the Create Procedure Wizard dialog box.

To open Create Procedure Wizard, go to *WEAVR > Procedures > Create Procedure*

Figure 93. Create Procedure Wizard, first configuration

This wizard allows you to configure:

- Procedure Name: write the name of the procedure.
- Filepath: choose where to save the procedure (from the Procedures folder, which is automatically created in Assets when configuring the WEAVR project), and the name of the procedure file.
- Select a Template:
  - VT: Virtual Training application is used for both immersive (Virtual Reality) and non-immersive trainings. It involves interactions with the objects in the scene.
  - OPS: Operation Support application does not involve interactions with the objects, only a step-by-step flow of the procedure.

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

2.13.1.1 *Virtual Training Template*

If you select the VT (Virtual Training) template, the following additional settings appear in the Procedure Wizard (Figure 94.).

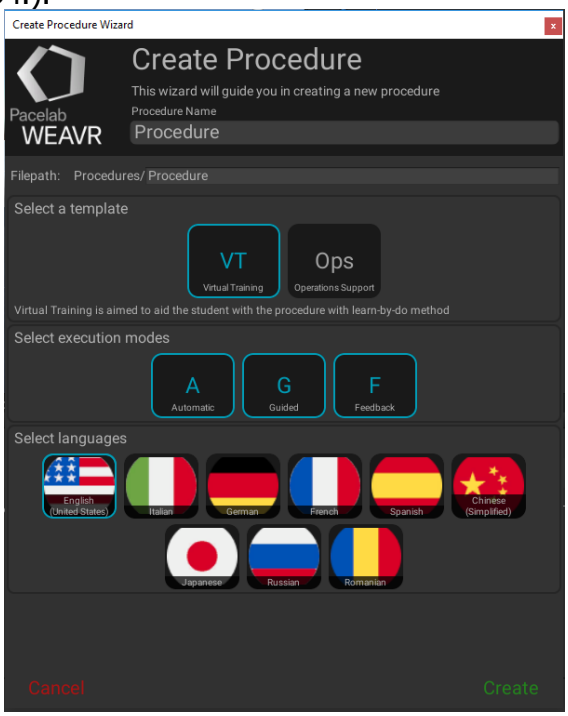


Figure 94. Create Procedure Wizard, VT settings

- Select execution modes: select the available modes of the procedure.
- Select languages: select the languages that will be converted to speech TTS (Text To Speech) and displayed in Billboard actions (Section 2.13.4.6).

Once you choose the procedure settings, press the Create button to create the procedure.

2.13.1.2 *Operation Support Template*

If you select the Ops (Operation Support) template, new settings in the Procedure Wizard appears (Figure 95.).

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

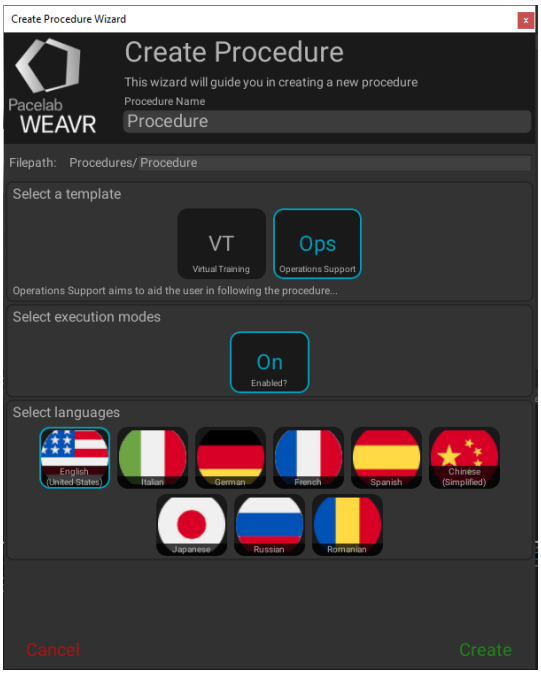


Figure 95. Create Procedure Wizard, Ops settings

- Select execution modes: the only available execution mode for Ops procedures is "On", as the Operation Support involves a step-by-step training. (The execution mode "On" needs to be selected to enable the "Create" button).
- Select languages: select the languages that will be converted to speech TTS (Text To Speech) and displayed in Billboard actions (Section 2.13.4.6).

Once you choose the procedure settings, press the Create button to create a procedure

2.13.2 Procedure Graph

The Procedure Graph section gives an overview of the procedure components, features, and settings.

2.13.2.1 *Procedure Editor*

The Procedure Editor window shows the stream of steps to be defined according to the procedure. To open the Procedure Editor window, go to *WEAVR > Procedures > Procedure Editor*

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

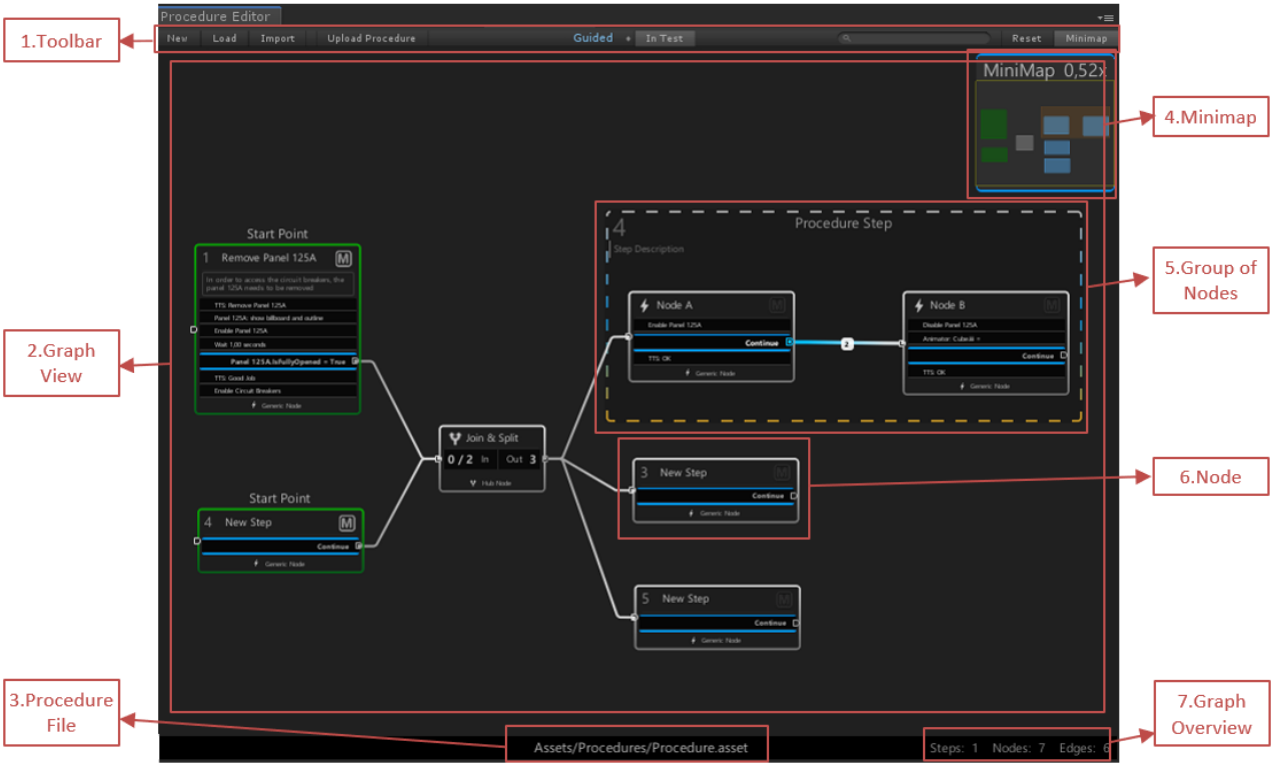


Figure 96. Procedure Editor window

The Procedure Editor window is comprised of the following elements:

1. Toolbar: navigation toolbar that contains commands to create, test, and manipulate a procedure.
2. Graph View: the main view of the procedure, where you can drag, connect, and edit procedure elements (Nodes, Groups, Transitions, etc.), as well as visually debug the procedure.
3. Procedure File: the relative path to the procedure file. If you click a procedure file, the Procedure Inspector shows the procedure file settings.
4. Minimap: shows a small preview of the procedure. It is useful to view complex procedures.
5. Group of Nodes: a group of nodes which is also a procedure step.
6. Node: the smallest unit of execution of a procedure.
7. Graph Overview: small statistics from the graph.

The toolbar of the Procedure Editor window is shown below.

Figure 97. Toolbar of the Procedure Editor window

You should interact visually with the graphic interface of the editor using a mouse and a keyboard. The procedure should be saved in a file .asset. The following commands allow you to manage the procedure:

- New: creates a new procedure.

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

- Load: loads a procedure to the editor. You can filter the list of procedures down to the procedures related to the current scene or all the procedures in the project (Figure 98).
- Import: imports an XML file (legacy procedure file).
- Upload Procedure: prepares the procedure to be loaded and/or executed in the player.
- Automatic/Guided/Feedback: selected execution mode in which the procedure will be tested.
- In Test: tests the procedure. If enabled, the procedure starts playing as soon as you click the Play button  .
- Fast-Forward Debug. 
- Save state. 
- Search: searches by name, GameObjects, or actions in a procedure.
- Reset: resets the procedure to its default state removes all debug colors and resets all nodes and transitions.
- Backup: Backs up the procedure.
- Minimap: toggles the minimap. For the very first time the minimap requires to zoom in or out to be visible.

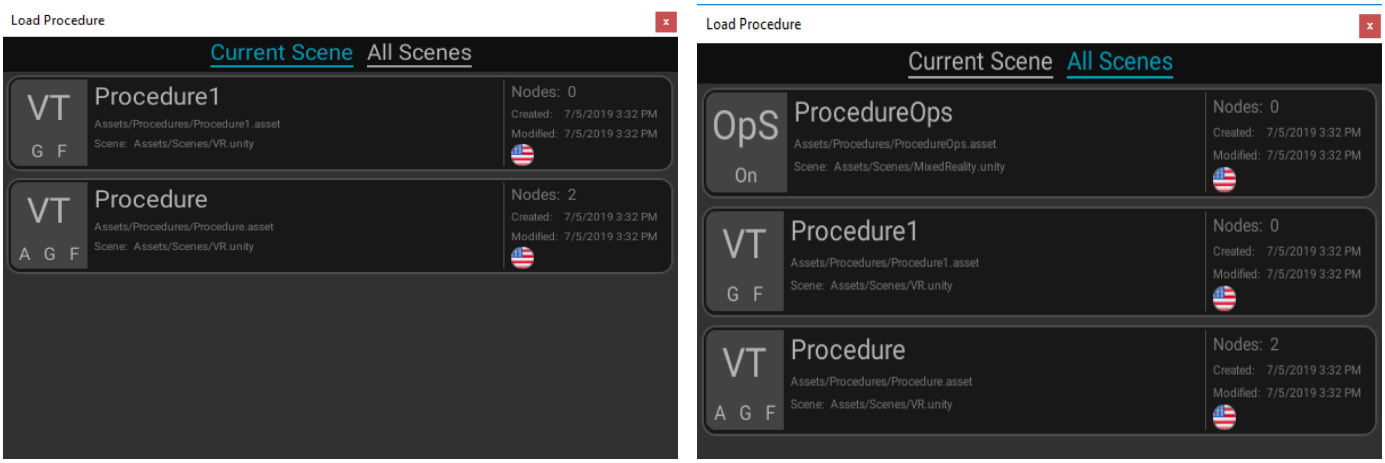


Figure 98. Load Procedure Window for Current Scene (left) and All Scenes (right)

2.13.2.1.1 Procedure File

You can access the procedure file settings by,

- Case 1: click on the procedure file .asset in the Project window.
- Case 2: click on the Procedure File name in the Project Editor window.

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

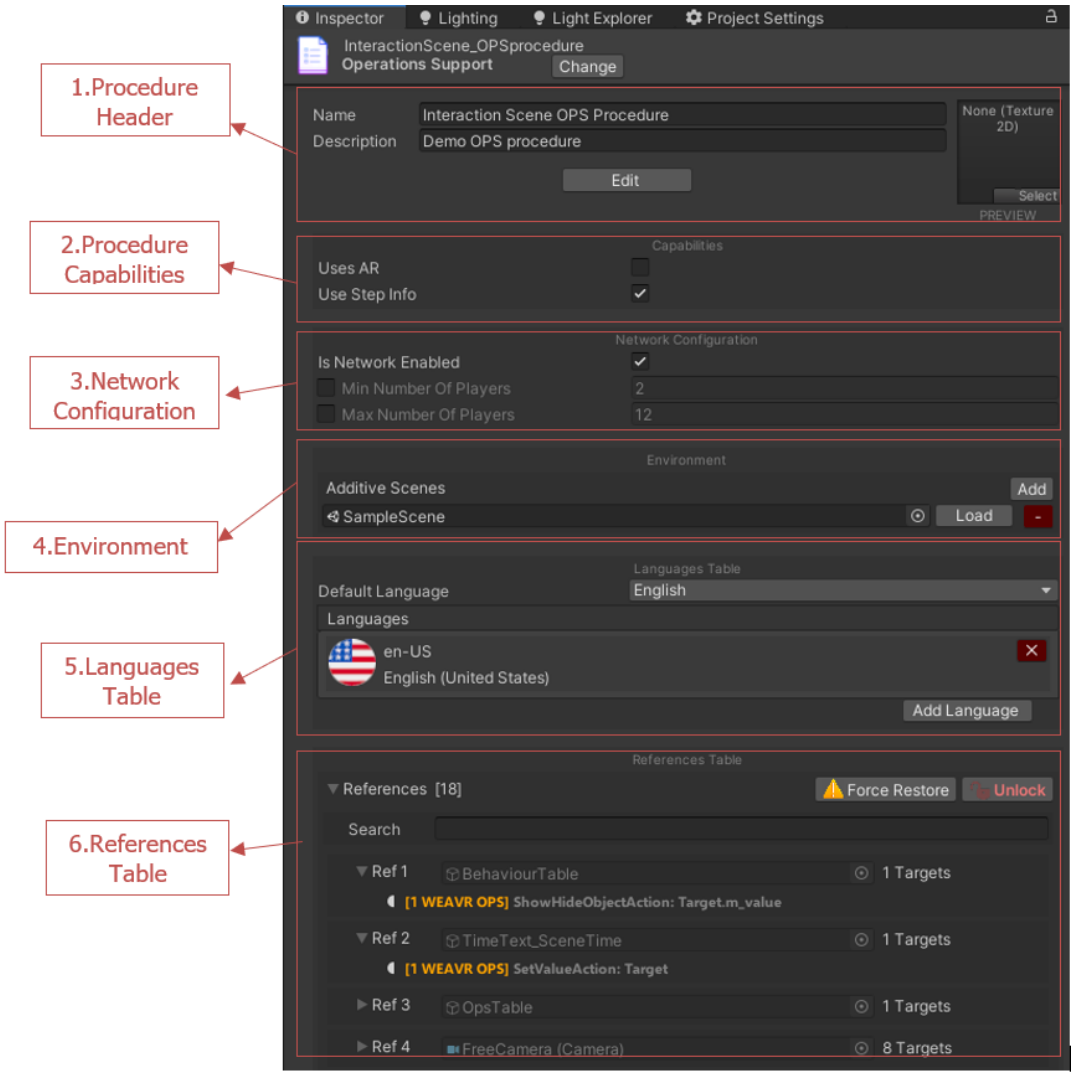


Figure 99. Procedure file from Inspector

1. Procedure Header: allows you to modify a procedure name, and a description, as well as preview an image. You can load the procedure to the Procedure Editor window by clicking the Edit button.
2. Procedure Capabilities: sets procedure features:
   - Uses AR: if enabled, the AR mode is used in the procedure.
   - Use Step Info: in enabled, the step title and a description are visualized in the player.
3. Network Configuration: sets multiplayer features:
   - Is Network Enabled: if enabled, you can use the procedure in multiple players.
   - Min/Max Number of Players: if enabled, you can define the number of players in the procedure scene.
4. Environment: if the procedure is shared among multiple scenes, the list includes the additive scenes.
5. Languages Table: allows you to modify the language settings of the procedure:
   - Default Language: select the default language from the procedure languages.
   - Add Languages: add new languages to the procedure after it has been created.
6. References Table: lists all GameObjects that are referenced in the procedure. The number and targets are specified for each reference.
   - Locked/Unlocked: if Unlocked, the referenced GameObject can be replaced for all its procedure targets. If required, the object is replaced together with references to its children.

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

TROUBLESHOOTING
If you open a procedure, and the actions and exit conditions do not have references, click the "Force Restore" button. The references will be restored in the procedure according to the new unique ID in the scene.

2.13.2.1.2 Graph View Shortcuts:

You can use the following commands from context menus *(by right-clicking objects)* or from shortcuts:

- **CTRL + C**: copy the graph selection (Nodes and Groups).
- **CTRL + X**: cut the graph selection (Nodes and Groups).
- **CTRL + V**: paste the graph selection (Nodes and Groups).
- **CTRL + D**: duplicate the graph selection (Nodes and Groups).
- **CTRL + G**: create a group from the graph selection (Nodes that are not in other Groups).
- **DELETE**: deletes the graph selection (Nodes and Groups).
- **F**: frames the graph selection, if there is any. Otherwise frames the entire procedure.
- **ALT +Left Mouse Button**: move around the graph.

2.13.2.2 *Node*

A graph node (or procedure step) is the main unit of a procedure. A node, by default, is a procedure step. However, when grouped, it loses the status of a procedure step and becomes a part of a larger procedure step. The graph node always contains a set of enter actions (can be empty), a set of conditions, and a set of exit actions (can be empty).

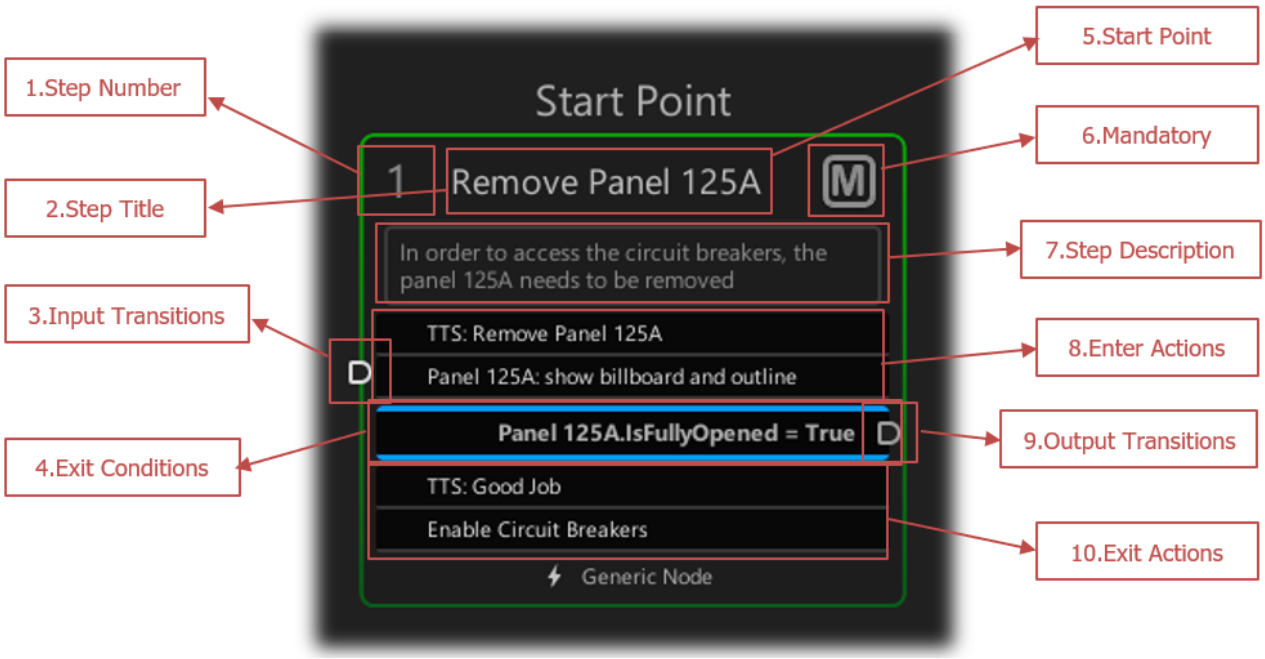


Figure 100. Simple Node (Procedure Step)

A graph node contains the following elements:

1. Step Number: number of the procedure step. It is generated automatically but can be modified further *(Applies only if the node is a procedure step).*
2. Step Title: title of the node (or a procedure step). It supports multiple languages and can be modified both from the Graph Editor and Procedure Inspector.

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

3. Input Transitions: Point where transitions from other nodes are collected.
4. Exit Conditions: List of exit transitions. Each exit condition is listed between the blue lines.
5. Start Point: Label that indicates if the procedure starts with this node. *Note that the procedure can have multiple start nodes.*
6. Mandatory: Label that indicates whether this node is mandatory or not. A mandatory node cannot be skipped by the user. You can define this toggle from Graph Editor or Procedure Inspector.
7. Step Description: Description of the step. It supports multiple languages *(Applies only if the node is a procedure step).*
8. Enter Actions: Set of actions to be executed when the node starts its playback.
9. Output Transitions: Path to follow when the corresponding exit condition is evaluated to true.
10. Exit Actions: Actions to be executed when any of the exit conditions have been evaluated to true.

There are small graphical differences when a node is a procedure step and when it is not. Functionally, however, a procedure step has additional information and can be skipped or undone by the user. The following picture (Figure 101) illustrates two structurally identical nodes, but one node is a part of a procedure step, and the other node is a procedure step.

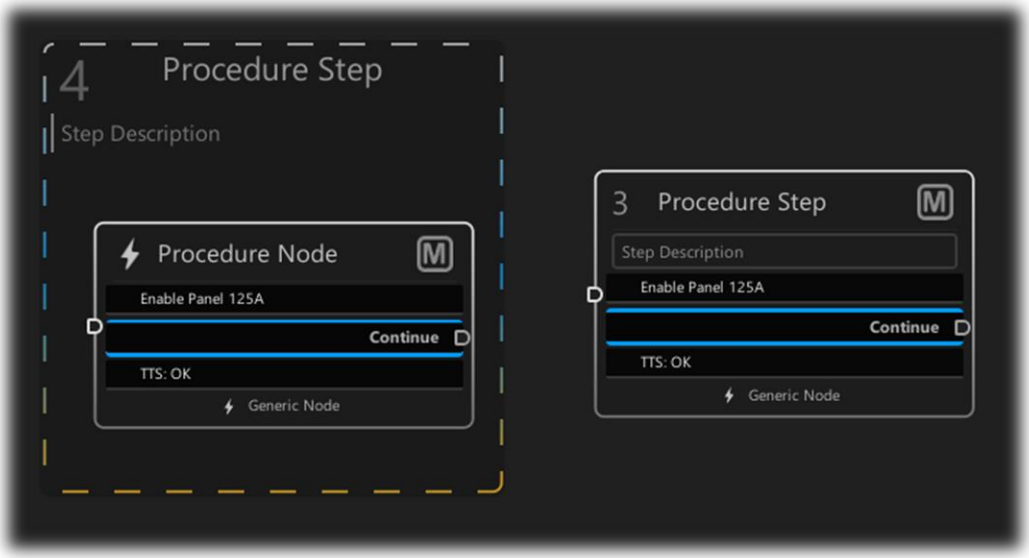


Figure 101. Two identical nodes with different purposes. Left node is part of step, Right node is a step

To create a Node:

- In the Procedure Editor window, right-click any part of the Graph View, and then click Create Node.

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

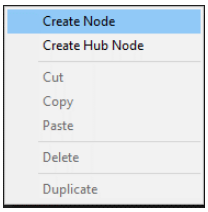


Figure 102. Create Node flow

By default, a newly created node is a not mandatory procedure step with empty lists for the actions, exit conditions, and exit actions. You can add them from the Procedure Inspector window (section 2.13.3).

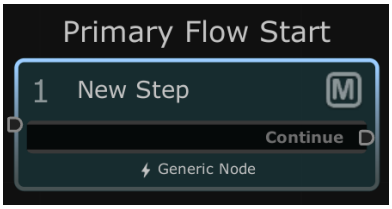


Figure 103. Default procedure step

The first created node has a "Primary Flow Start" label, while the "Start" label indicates that it is a starting point of the procedure. There can be multiple starting points in a procedure, and you can toggle them at any time. The steps, which are before the steps with starting points, are not executed.

Two types of starting points are the following:

- *Primary Flow Start:* references the Title, Description, and Number of the procedure progress. Furthermore, the Primary Flow is affected by the next and previous steps of the procedure.

To toggle a Primary Flow Start:

Right-click any Node > Main Flow

Figure 104. Primary Flow Start

- *Secondary Flow Start:* progresses without affecting the Title, Description, and number of the current procedure step. It is not affected by the manual progress of the procedure (next and previous steps).

Figure 105. Secondary Flow Start

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

2.13.2.2.1 Mandatory Node (OpS procedure)

When a node is set to mandatory in an Operation Support procedure, the exit condition of that node automatically executes when you click the Next button in the Game window. Figure 106 You can see an example of two mandatory nodes that are visualized in the Procedure Editor window (Figure 104). The procedure will not progress to the next step until you click the Next button.

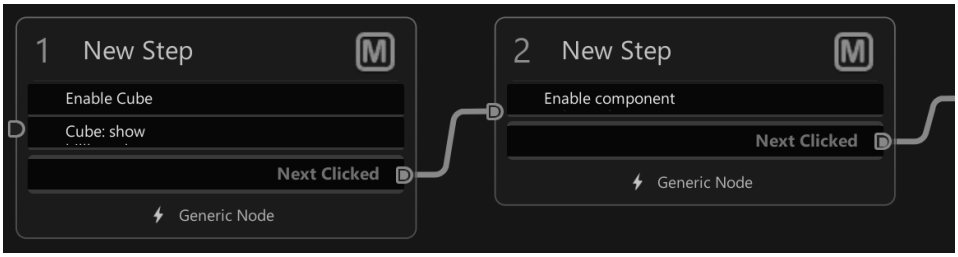


Figure 106. Mandatory node

2.13.2.3 *Nodes Group (Super Step)*

You can group graph nodes into groups. Together groups act as a single procedure step. A group is also referred to as Super Step to indicate its purpose: a step with multiple nodes.

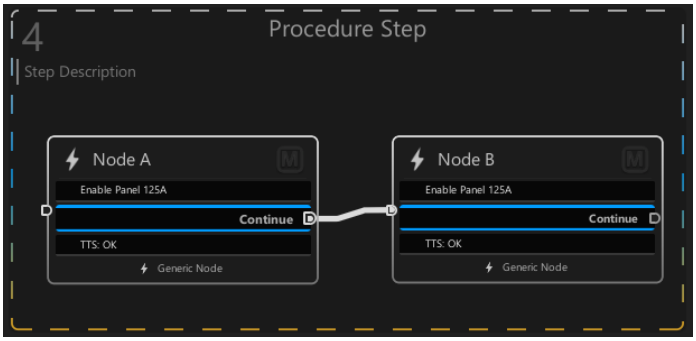


Figure 107. Group of Nodes (Super Step)

The group is composed of a title (supports multiple languages), a description (supports multiple languages), and a number.

To create a Super Step:

- In the Procedure Editor window, right-click selected nodes, and then click Add Group.

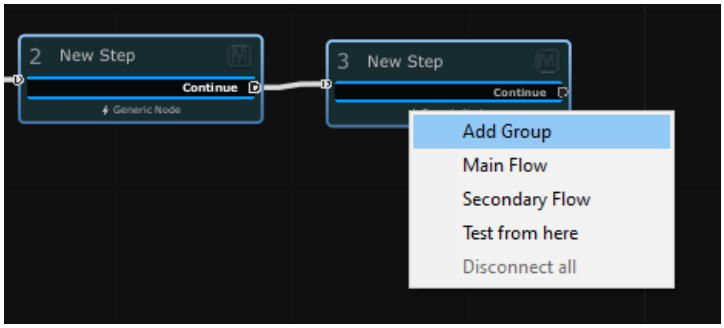


Figure 108. Create Group flow

Grouping is important from the functional point of view, as well as organizational one. A group is always a procedure step, and encapsulated nodes represent the complex logic of the step. By default, a newly created group has its title set to "Super Step", thus enforcing its purpose. The title

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

can be edited directly in the Graph View. However, for a more complete editing (e.g. multilanguage), refer to the Procedure Inspector window where you can also edit the description and step number.

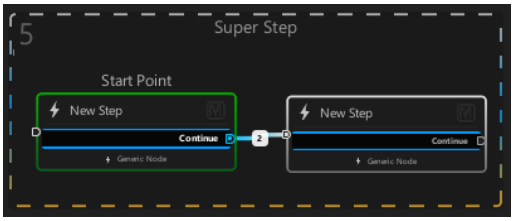


Figure 109. Default Group

You can add and remove nodes from groups at any time. However, the same node cannot be present in more than one group at the same time.

- To add a node to an existing group: drag the node over the group.
- To remove a node from a group: select the node, and then simultaneously hold Shift and drag the node out of the group.

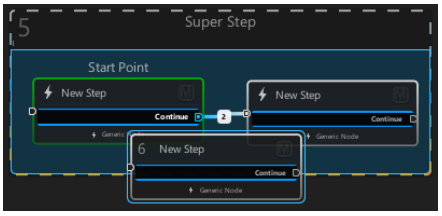


Figure 110. Add Node to Group

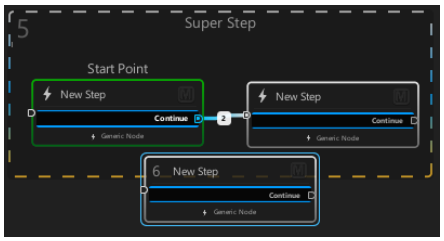


Figure 111. Remove Node from Group

2.13.2.4 *Hub Node (Join & Split)*

A hub node, as its name implies, waits for all the incoming transitions, and launches more execution flows in parallel.

Figure 112. Hub Node

No actions need to be taken for this node. In the Figure 110, the hub node waits for 2 transitions (0 waited out of 2) and launches 3 execution flows in parallel.

To create a Hub Node:

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

In the Procedure Editor window, right-click any part of the Graph View, and then click Create Hub Node.
By default, a newly created hub node has no input and output transitions, and the title is "Join & Split".

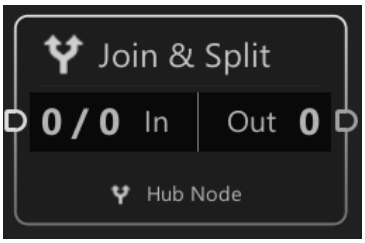


Figure 113. Hub Node

2.13.2.5 *Transition*

A Transition is a link between two Nodes. You can add multiple actions to be executed when the transition path is followed.

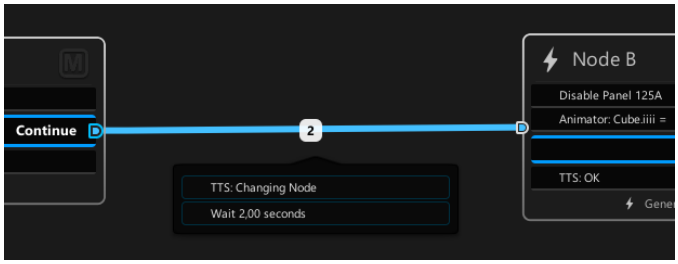


Figure 114. Transition with Actions

The set of transitions represents the navigation in the Graph.

To create the flow from one node to another, and set the execution order of the procedure, use transitions between nodes.

To make a transition:

- Click the output transition point of the node (Figure 115), and then drag the line to the input transition point of the step to connect.

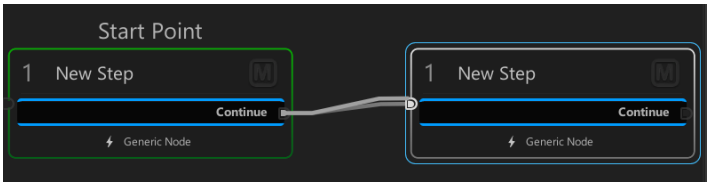


Figure 115. How to make a node transition

When a transition between nodes is valid, a preview of the transition is shown. Once a transition is established, it can be selected and loaded with actions if needed. If a transition is loaded with actions, a small "badge" is visible over the transition. The set of transition actions can be edited in the Procedure Inspector window.

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

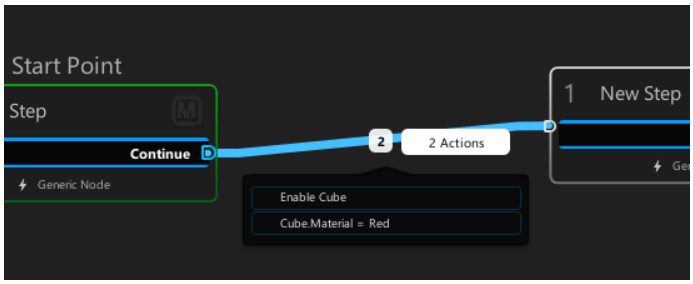


Figure 116. Transition overloaded with 2 actions

2.13.3 Procedure Inspector

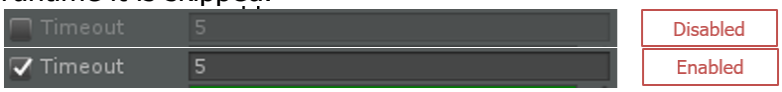
2.13.3.1 *Overview*

The Procedure Inspector window allows you to see the details of procedure items (Nodes, Groups, Transitions, etc.) and modify them. The window consists of a header and a body. The Procedure Inspector window opens automatically when you select a procedure item in the Graph View. Additionally, you can open the inspector by going to *WEAVR > Procedures > Inspector*

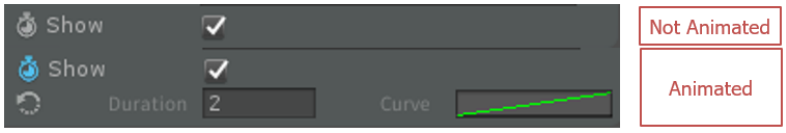
Each procedure item has its own inspector.

You can configure and edit many properties of procedure items in the inspector. Many properties are similar to default properties in Unity. However, some of them are quite different and require a better explanation:

- **Optional Value**: these properties can be enabled or disabled by a checkbox on their left. When enabled, the value is used, when disabled, the value cannot be edited and at runtime it is skipped.



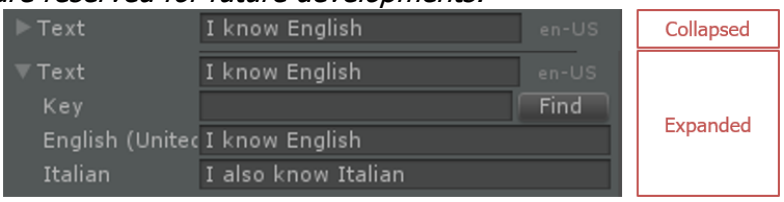
- **Animated Value**: these properties are animated according to their target values during the specified time (duration). To enable the animation, select the chronometer toggle ( ). The duration specifies how much time the transition takes for the property to reach its target value. The curve describes how to reach the target value. The Revert button ( ), when available and if enabled, animates the property to its original value when exiting the node.



- **Multilanguage Value**: these properties can have different values per language. The arrow on the left is for expanding/collapsing the property. When collapsed, the property shows its value in the currently set language (the language id can be seen on the right side). When expanded, all language values can be seen. *The "Key" field and "Find" button*

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

are reserved for future developments.



- **Optional Animated Value**: these properties are both optional and animated.

2.13.3.2 *Node Inspector*

A newly created node, by default, has the following inspector:

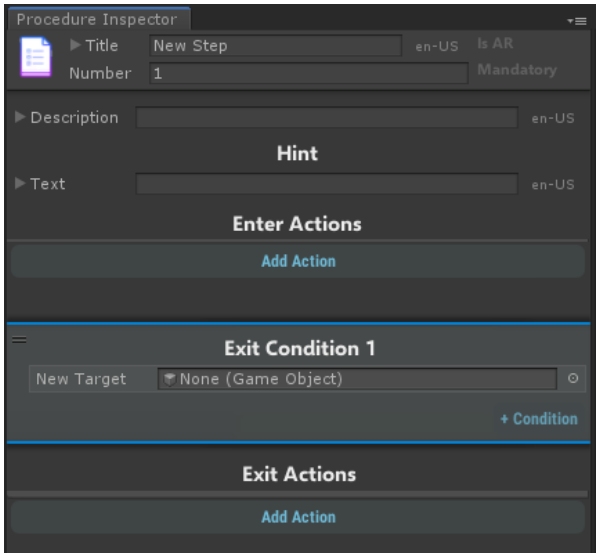


Figure 117. Default Node Inspector

The Node inspector has 6 sections:

- Head of a step inspector: consists of the step number (if it is a procedure step, otherwise it is disabled), the title of the node, and information whether the step is mandatory or not.
- Description: visible only if it is a procedure step; allows editing the description of the procedure step.
- Hint: used to guide a user in difficult tasks.
- Enter Actions: the list of actions which are executed when the node is reached.
- Exit condition(s): condition(s) that are used to exit the current step.
- Exit actions: the list of actions which are executed right after any of the exit conditions are evaluated to true.

When a node is a procedure step, its description field is visible, and the number field is enabled. The node inspector supports the copy-and-paste functionality.

In the upper-right corner of the inspector, there are two toggle buttons: "Pre-check" and "Mandatory". "Pre-check" checks the exit condition status before entering the step, while "Mandatory" toggles the step to be mandatory or not.

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

2.13.3.3 *Actions Inspector*

An action is the smallest element which can be executed by the system. It is defined in a context and can live only in a context. A context can be anything from a node to a transition. The context defines a boundary where execution elements, such as an action or a condition, can run and exchange data.

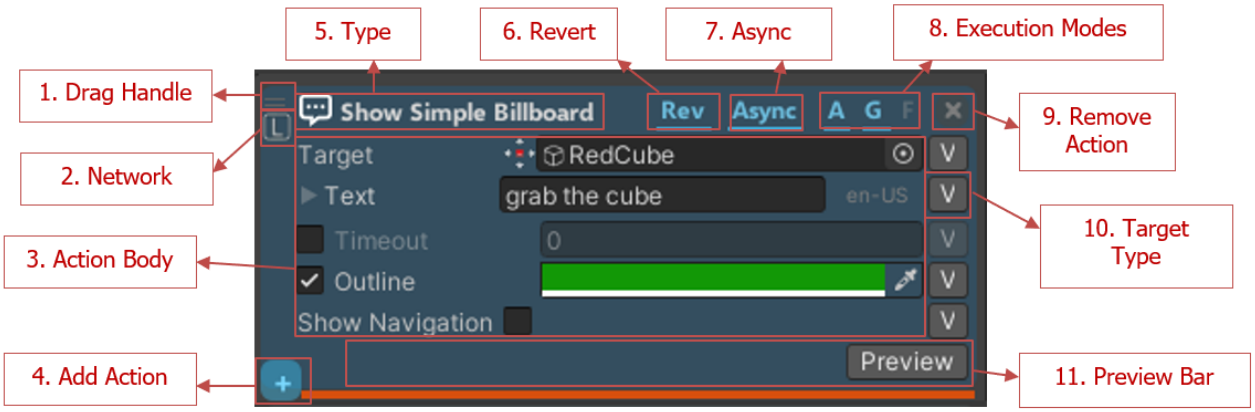


Figure 118. Actions Inspector

An action has the following elements:

1. Drag Handle: moves the action up and down the list.
2. Network: whether the action is applied locally or globally (across network).
3. Action Body: Properties of an action that are available for a modification.
4. Add Action: adds a new action after the current action.
5. Type: type of the action.
6. Revert: when enabled, reverses the changes to this action on exiting the context (Node, Transition, etc).
7. Async: when enabled, the action is executed asynchronously (in parallel with other actions).
8. Execution Modes: supported execution modes for this action.
9. Remove Action: Deletes the action from its context (Node, Transition, etc).
10. Target Type: whether to use a GameObject target or a Variable.
11. Preview Bar: previews the execution of the action. Preview bars differ for every action.

Actions are executed according to their specified execution modes. If you define the same execution mode) for the action and the procedure, the action will be executed.
You can move actions around the list (even from Enter Actions to Exit Actions) and copy them inside the same or various types of contexts.

You can quickly replace actions with other actions that belong to the same group. To do that, right-click the action Type, and then select the required action (see Figure 119).

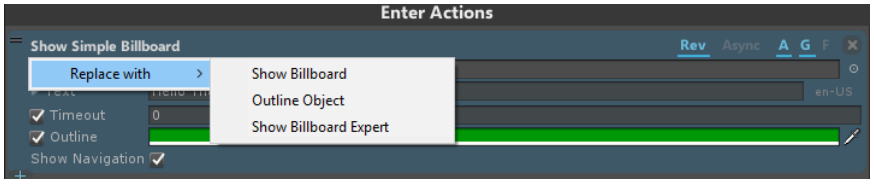


Figure 119. Replace action shortcut

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

2.13.3.4 *Conditions Inspector*

A condition is the smallest evaluation element of the system which is used to trigger the navigation throughout the procedure. A condition usually needs to be evaluated to true to trigger a navigation event. Conditions can be combined to get more complex logic expressions. Every node has a set of topmost conditions in an OR expression with each child having a transition to other nodes. Each child starts with an AND expression where its children can be any type of condition (e.g. AND expression, OR expression, etc.). Conditions can be moved and copied inside the same context, as well as between contexts. Every condition can be negated by using the NOT toggle on the left of each condition.

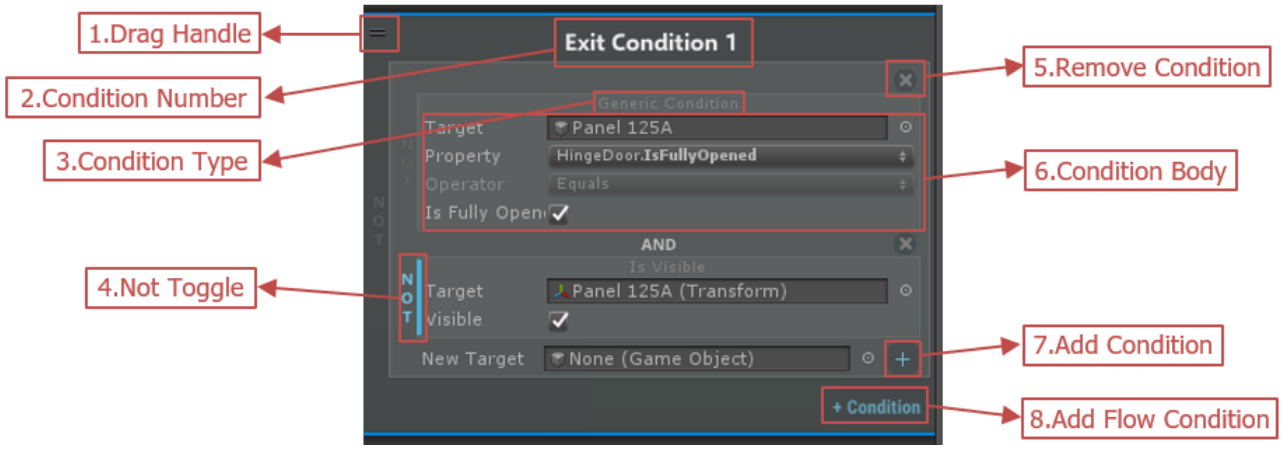


Figure 120. Example of conditions of a node

A condition has the following elements:

1. Drag Handle: moves the condition up and down the list.
2. Condition Number: order number of the conditionto be evaluated.
3. Condition Type: type of the condition. Implicit conditions do not show their types.
4. NOT Toggle: if selected, negates the condition. All conditions at any depth can be negated. However, only the topmost negated conditions are shown in Graph View.
5. Remove Condition: removes the condition.
6. Condition Body: the properties of conditions. Each type of condition has its own body.
7. Add Condition: adds a sibling condition to the condition immediately before
8. Add Flow Condition: adds a topmost condition (in the OR expression) with its own transition.

The list of Exit Conditions is explained later in the section.

2.13.3.5 *Transition Inspector*

The Transition inspector consists of a header and a body. Its header contains the buttons to reach the connecting nodes, while its body contains the list of actions to be executed when following the path of the transition.

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

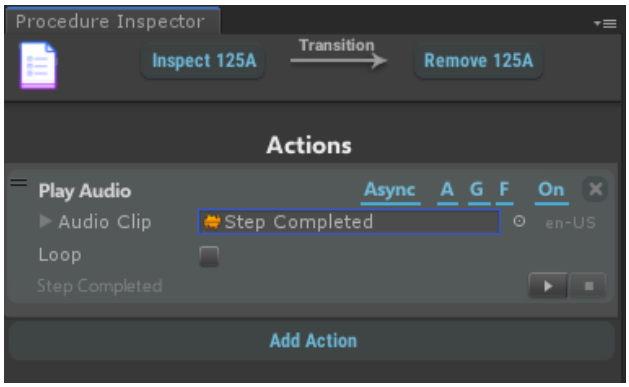


Figure 121. Transition Inspector

2.13.3.6 *Hub Node Inspector*

The Hub Node inspector consists of a header and a body. The header contains the title of the node, while the body contains the number of input connections and links (both transition and node) to output transitions.

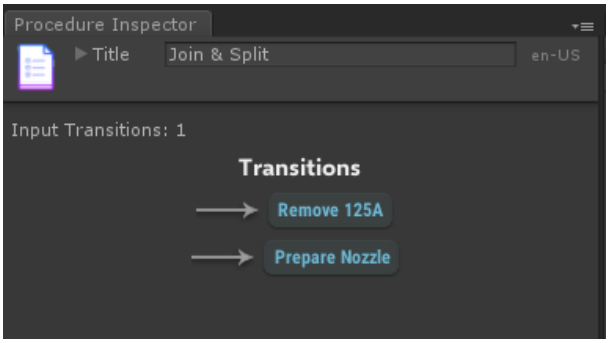


Figure 122. Hub Node Inspector

2.13.3.7 *Nodes Group Inspector*

The Nodes Group Inspector consists of a header and a body. The header contains the title (supports multilanguage) and the number, while the body contains the description (supports multilanguage) and the list of nodes. To display the Nodes Group Inspector, click any node.

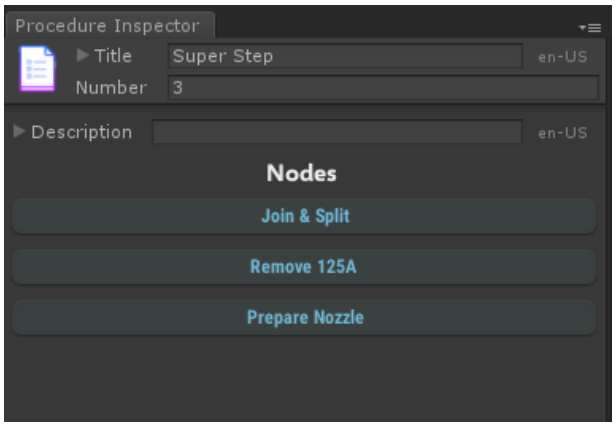


Figure 123. Group of Nodes (Super Step) Inspector

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

2.13.4 Actions

An Action informs you about something (for example, tells a user where to stand or what object to click), changes the position of a camera (for example, to have a closer view of an object), shows or hides an object from a user, and other. You can assign actions from an always growing list of actions: Animation, Audio, Camera, Canvas, Control Flow, Hints, and Objects.

2.13.4.1 *Animation*

The Animation group contains the following actions:

- Animations
- Set Animator Parameter
- Play Animation Clip
- Set Animator State

2.13.4.1.1 Animations

The Animations action is a container of Animation *Blocks.* The Animation Block is a simple movement, a change of an object setting, or other that you can add to other simple animations to create the needed full animation. The Block types are: Wait, Delta Move, Move to Target, Change Color and Texture, Change Material, and Change Transparency. Each Block type is explained later in the section. You can configure individual animation settings in Blocks or the whole animation in the Animation window. To open the Animation window, go to Window > Animation > Animation.

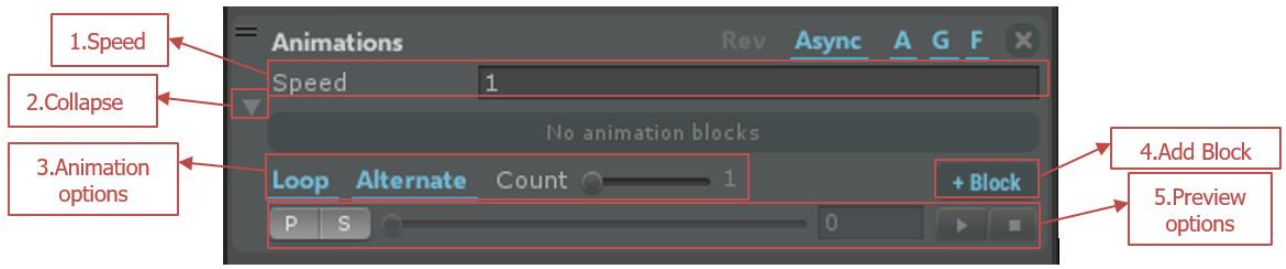


Figure 124. Animations settings

The Animations action contains the following settings:

1. Speed: speed of the whole animation (in m/s).
2. Collapse: Expands or collapses the blocks.
3. Animation options: define the number of times and the modality the whole animation is played.
    - Loop: when highlighted, the animation is played in a loop according to the following settings:
        - Alternate: if highlighted, the animation is reproduced alternately forward and backward (Figure 125).
        - Count: the animation is reproduced the indicated number of times. If you move the slider to the rightmost position, the animation will be reproduced infinite times.
4. Add block: creates a new Animation Block.
5. Preview Options: visualizes the animation in the scene without entering the Play mode. The options are as follows:
    - P: if clicked, the preview of the whole animation can be visualized in the scene.
    - S: if clicked, the directions, step numbers, transforms, etc. of the Animation Blocks appears in the scene and can be set directly in the scene. Figure 126 shows the example of a scene preview in case of the Delta Move animation; in this example, it is possible to change the rotation and translation of the object from the scene.

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

- Slider: moves to a particular episode of the animation.
- ▶ : plays the animation in the scene.
- ■ : stops the animation preview.

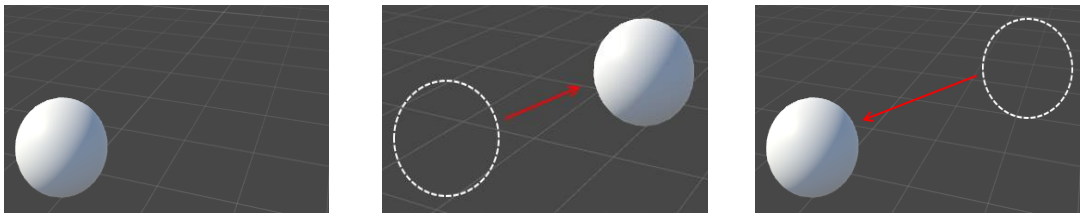


Figure 125. Example of alternate action

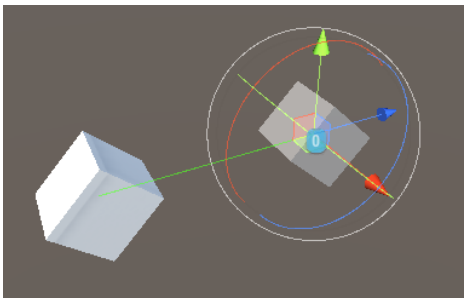


Figure 126. Example of preview in the scene of the Delta Move action

The Animations block allows you to create a animation that can involve one or more simple animations, using the available Animation Blocks.

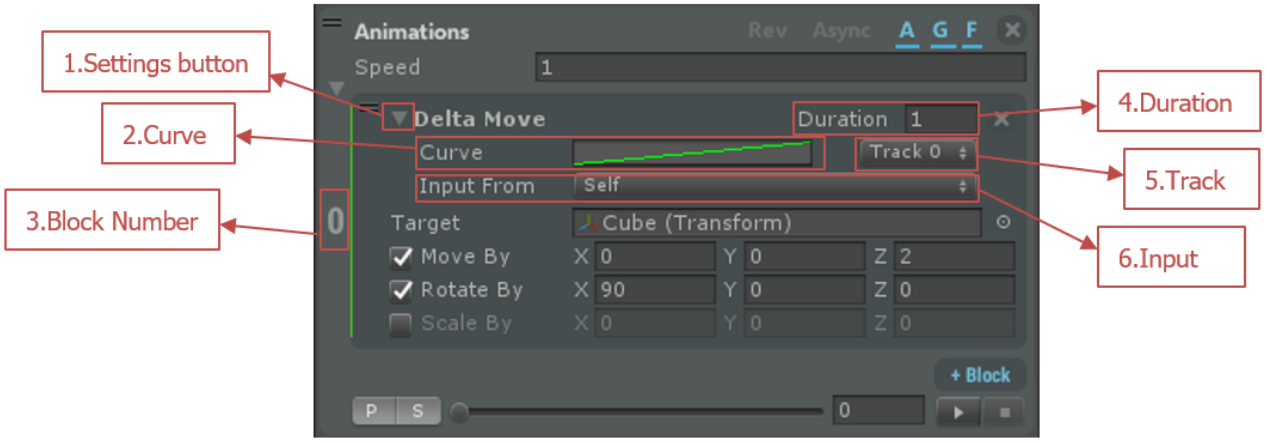


Figure 127. Animations Block settings

Settings:

Setting Name	Description
Settings button	Expands or collapses the block settings.
Curve	Displays an animation curve.
Block Number	Displays the number of a single Animation block. It is useful when the animation lines are shown in the scene (P and S buttons) to distinguish the Animation blocks. Figure 126 shows the example of the Delta Move block which is the number 0 (see the blue square).
Duration	Displays the duration of the Animation block (in seconds).
Track	Displays the track number of the Animation block of the same Animations action. Animations blocks that have the same track can be played in a sequence or one after the other, while Animations blocks

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

	with different tracks can be played simultaneously. The blocks with the same track number are highlighted with the same color to the left of the block. Figure 128 is an example of blocks with different track numbers: block 0 with the track 0 will start simultaneously with the block 1 with track 1, while block 2 (with track 1) will follow block 1.
Input	References the block target in different ways according to the indicated setting: ○ Self: the target is set manually and individually. ○ From Previous: the target is set automatically according to the previous block target, no matter of the track number. ○ From Previous in Track: the target is set automatically according to the previous block target having the same track. ○ From Block Index: the target is set automatically according to the block number indicated (a number list appears to the right of the setting).

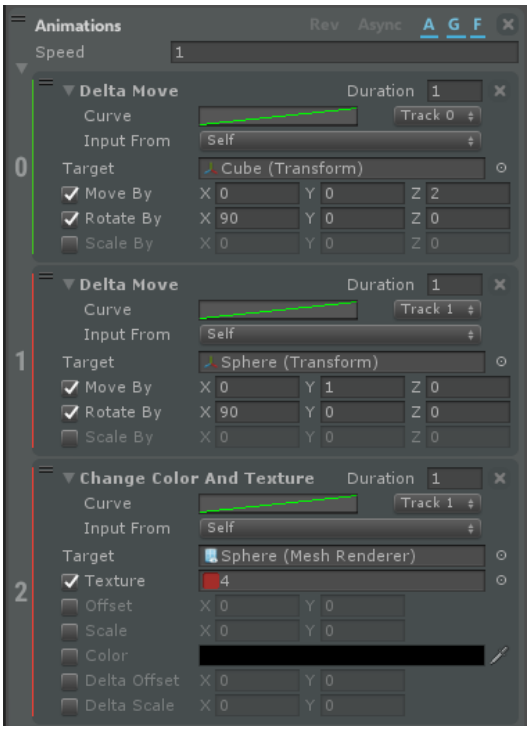


Figure 128. Example of Animation Blocks with different track number

Wait

The Wait block allows you to set a waiting time before continuing with step actions.

Figure 129. Wait

Settings:

Setting Name	Description
Duration	Displays the duration of the Wait block (in sec).
Curve	Displays the block animation curve (Figure 127).
Track	Displays the block track number (Figure 127).

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

Delta Move

The Delta Move block allows you to set a free delta movement of an object, translation, rotation, and scale of the object.
Figure 130. shows the sphere moving from its initial position to the final position according to the defined delta of translation (move) and rotation.

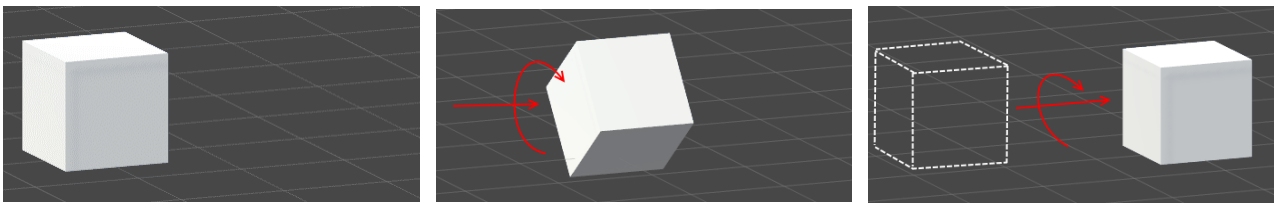


Figure 130. Example of translation and rotation with Delta Move animation

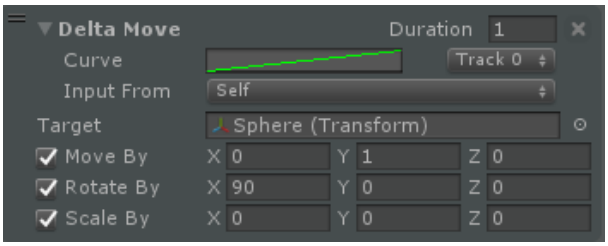


Figure 131. Delta Move

Settings:

Setting Name	Description
Duration	Displays the duration of the Delta Move block (in sec).
Curve	Displays the block animation curve (Figure 127).
Track	Displays the block track number (Figure 127).
Input From	References the block target (Figure 127).
Target	Object to animate.
Move By	Sets the translation deltas.
Rotate By	Sets the rotation deltas.
Scale	Sets the scale deltas.

Move to Target

The Move to Target block allows you to set the final position of the target towards another existent object transform in the scene.
Figure 132. shows how the sphere moves from the original position to the position of the cube.

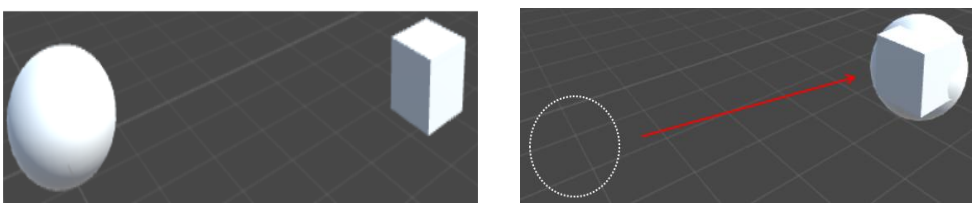


Figure 132. Example of move to target animation

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

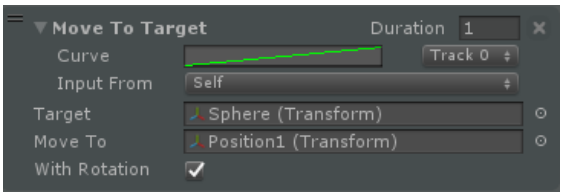


Figure 133. Move to Target

Settings:

Setting Name	Description
Duration	Displays the duration of the Move to Target block (in sec).
Curve	Displays the block animation curve (Figure 127).
Track	Displays the block track number (Figure 127).
Input From	References the block target (Figure 127).
Target	Object to animate.
Move To	Sets the destination of the target.
With Rotation	If enabled, the rotation of the object that will move matches the orientation of the destination transform.

Change Color and Texture

The Change Color and Texture block allows you to set the color and/or texture of the target object. This animation is related to the individual Set Color and Set Texture actions.

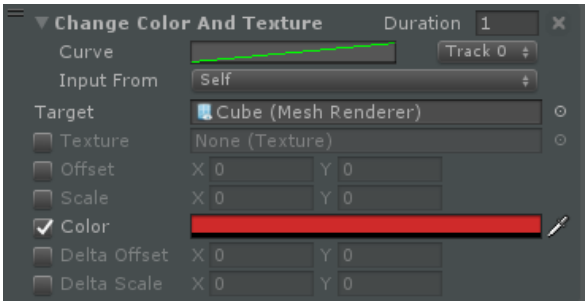


Figure 134. Change Color and Texture animation

Settings:

Setting Name	Description
Duration	Displays the duration of the Change Color and Texture block (in sec).
Curve	Displays the block animation curve (Figure 127).
Track	Displays the block track number (Figure 127).
Input From	References the block target (Figure 127).
Target	Object to animate.
Texture	If enabled, allows changing the texture of the target renderer.
Offset	If enabled, allows defining the offset to be applied to the main texture of the target renderer.
Scale	If enabled, allows defining the scale to be applied to the main texture of the target renderer.
Color	If enabled, allows selecting the color of the target.
Delta Offset	If enabled, allows defining the delta offset to be applied to the main texture of the target renderer.
Delta Scale	If enabled, allows defining the delta scale to be applied to the main texture of the target renderer.

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

Change Material

The Change Material block allows you to set the material of the target object. This animation is related to the individual Set Material action.

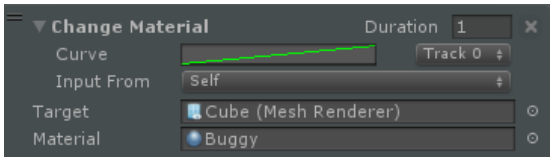


Figure 135. Change Material animation

Settings:

Setting Name	Description
Duration	Displays the duration of the Change Material block (in sec).
Curve	Displays the block animation curve (Figure 127).
Track	Displays the block track number (Figure 127).
Input From	References the block target (Figure 127).
Target	Object to animate.
Material	Material to be applied to the object.

Change Transparency

The Change Transparency block allows you to set the transparency of the object.

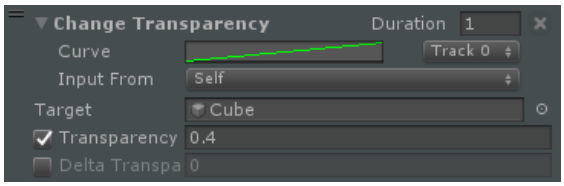


Figure 136. Change Transparency animation

Settings:

Setting Name	Description
Duration	Displays the duration of the Change Transparency block (in sec).
Curve	Displays the block animation curve (Figure 127).
Track	Displays the block track number (Figure 127).
Input From	References the block target (Figure 127).
Target	Object to animate.
Transparency	If enabled, the transparency of the object can be set. The value goes from 0 to 1, where 0 is invisible, and 1 is fully visible.
Delta Transparency	If enabled, the delta transparency of the object can be set. The value goes from 0 to 1, where 0 is invisible, and 1 is fully visible. Delta Transparency cannot be defined if Transparency is enabled.

2.13.4.1.2 Set Animator Parameter

The Set Animator Parameter block allows you to set the defined Parameter value of the target Animator.

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

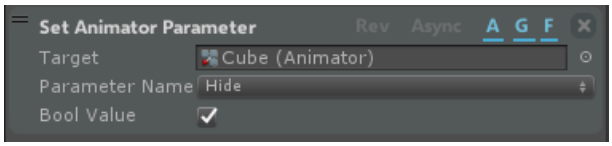


Figure 137. Set Animator Parameter

Settings:

Setting Name	Description
Target	Defines the Animator where to set the Parameter.
Parameter Name	Sets the parameter value.
Value	If enabled, sets the bool, float, or other value according to the Parameter type.

2.13.4.1.3 Play Animation Clip

The Play Animation Clip block allows you to assign an Animation Clip to the target object.

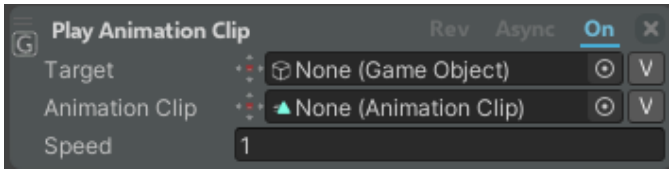


Figure 138. Play Animation Clip

Settings:

Setting Name	Description
Target	Object to be animated by the Animator.
Animation Clip	Clip to be played.
Speed	How fast the animation will be played. The negative value plays the animation backwards.

2.13.4.1.4 Set Animator State

The Set Animator State block allows you to set the current Animator State.

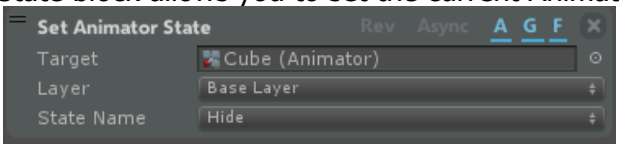


Figure 139. Set Animator State

Settings:

Setting Name	Description
Target	Target Animator.
Layer	Layer of the target Animator.
State Name	Animator state to be set as a target.

2.13.4.2 *Audio*

The Audio group contains the following actions:

- Play Audio
- Play Audio on Target
- Play Audio at Position
- Text to Speech

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

- Text to Speech on Target
- Text to Speech at Position
- Advanced

2.13.4.2.1 Play Audio

The Play Audio action plays the indicated audio on the background of the scene.

Figure 140. Play Audio properties in the Step Inspector window

Settings:

Setting Name	Description
Audio Clip	Audio clip to be reproduced.
Key:	Subject of further development.
Find	Subject of further development.
English ..	Different audio clips can be uploaded according to the defined languages in the procedure. If no audio clip is set, the main Audio Clip is played.
Loop	If enabled, the audio is reproduced in a loop (this parameter is invisible if the action is not Async).

2.13.4.2.2 Play Audio on Target

The Play Audio on Target action plays the indicated audio when you interact with the referenced Object of the scene.

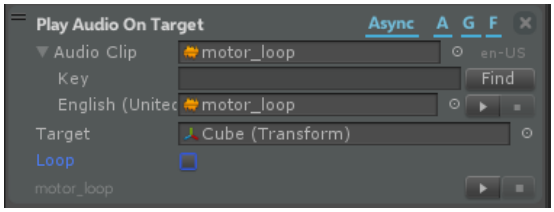


Figure 141. Play Audio on Target

Settings:

Setting Name	Description
Audio Clip	Audio clip to be reproduced.
Key:	Subject of further development.
Find	Subject of further development.
English ..	Different audio clips can be uploaded according to the defined languages in the procedure. If no audio clip is set, the main Audio Clip is played.
Target	Object for which the audio is played.
Loop	If enabled, the audio is reproduced in a loop (this parameter is invisible if the action is not Async).

2.13.4.2.3 Play Audio at Position

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

The Play Audio at Position action plays the indicated audio when you are in the needed position in the scene.

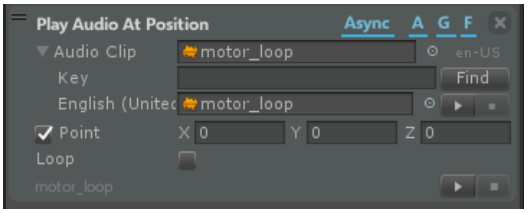


Figure 142. Play Audio at Position

Settings:

Setting Name	Description
Audio Clip	Audio clip to be reproduced.
Key:	Subject of further development.
Find	Subject of further development.
English ..	Different audio clips can be uploaded according to the defined languages in the procedure. If no audio clip is set, the main Audio Clip is played.
Point	Position in the coordinate system where to play the audio in respect to the center of the scene. If null, the audio is heard on the background of the scene.
Loop	If enabled, the audio is reproduced in a loop (this parameter is invisible if the action is not Async).

2.13.4.2.4 Text to Speech (TTS)

The Text to Speech action converts the text into audio that will be reproduced on the background of the scene.

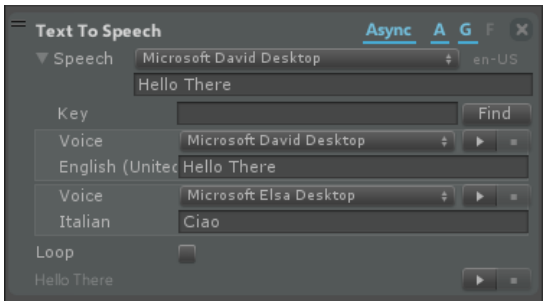


Figure 143. Text to Speech properties in the Step Inspector window

Settings:

Setting Name	Description
Text	Text to be converted into audio.
Key:	Subject of further development.
Find	Subject of further development.
Voice	Define the voice that reads the TTS (it can be different according to the language).
English ..	Different text can be written according to the defined languages in the procedure.
Loop	If enabled, the audio is reproduced in a loop (this parameter is invisible if the action is not Async).

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

NOTE
Windows Voice and Amazon TTS are both supported.
To use Amazon TTS, you should compile the WEAVR Settings and have a valid internet connection.

2.13.4.2.5 Text to Speech on Target

The Text to Speech on Target action converts the text into audio that will be reproduced from the referenced Object of the scene.

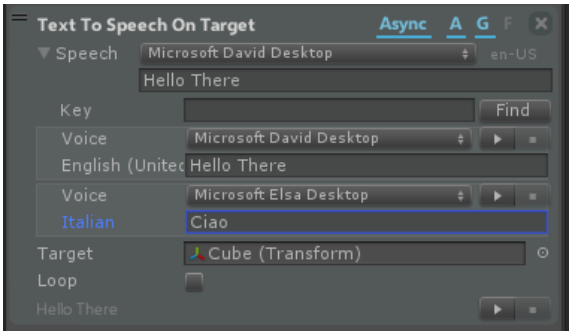


Figure 144. Text To SPeech On Target

Settings:

Setting Name	Description
Text	Text to be converted into audio.
Key:	Subject of further development.
Find	Subject of further development.
Voice	Define the voice that reads the TTS (it can be different according to the language).
English ..	Different text can be written according to the defined languages in the procedure.
Target	Object for which the audio is played. If null, the audio is heard on the background of the scene.
Loop	If enabled, the audio is reproduced in a loop (this parameter is invisible if the action is not Async).

2.13.4.2.6 Text to Speech at Position

The Text to Speech at Position action converts the text into audio when you are in the needed position in the scene.

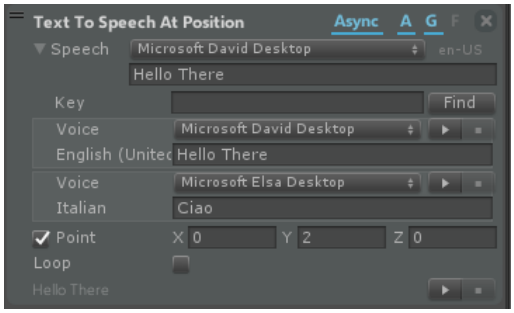


Figure 145. Text to Speech At Position

Settings:

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

Setting Name	Description
Text	Text to be converted into audio.
Key:	Subject of further development.
Find	Subject of further development.
Voice	Define the voice that reads the TTS (it can be different according to the language).
English ..	Different text can be written according to the defined languages in the procedure.
Point	Position in the coordinate system where to play the audio in respect to the center of the scene. If null, the audio is heard on the background of the scene.
Loop	If enabled, the audio is reproduced in a loop (this parameter is invisible if the action is not Async).

2.13.4.2.7 Advanced

Play Audio Clip Expert

The Play Audio Clip Expert action plays the indicated audio. You can customize the reproduction of the clip in different ways..

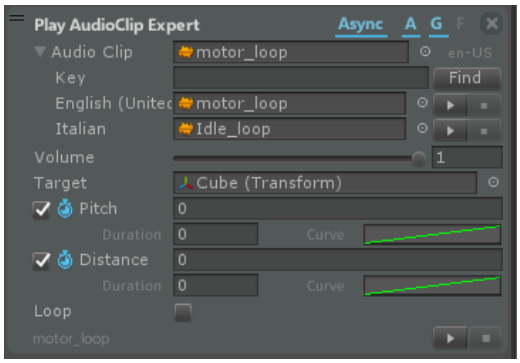


Figure 146. Play Audio Clip Expert

Settings:

Setting Name	Description
Audio	Audio clip to be reproduced.
Key:	Subject of further development.
Find	Subject of further development.
English ..	Different audio clips can be uploaded according to the defined languages in the procedure. If no audio clip is set, the main Audio Clip is played.
Volume	Volume of the audio source.
Target	Object for which the audio is played. If null, the audio is heard on the background of the scene.
Pitch	If enabled, the pitch of the audio can be customized. To animate the pitch, enable ⏱, and then define the following: • Duration: how long the animation lasts. • Curve: Animation curve.
Distance	If enabled, the audio becomes 3D, and the distance is the maximum distance from which you can hear the audio. To animate the distance, enable ⏱, and then define the following:

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

	• Duration: how long the animation lasts. • Curve: Animation curve.
Loop	If enabled, the audio is reproduced in a loop (this parameter is invisible if the action is not Async).

Text to Speech Expert

The Text to Speech Expert action plays the indicated TTS. You can customize the reproduction of the TTS in different ways.

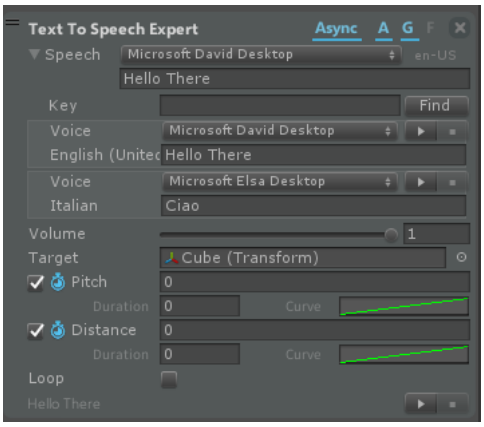


Figure 147. Text to Speech Expert

Settings:

Setting Name	Description
Text	Text to be converted into audio.
Key:	Subject of further development.
Find	Subject of further development.
Voice	Define the voice that reads the TTS (it can be different according to the language).
English ..	Different text can be written according to the defined languages in the procedure.
Volume	Volume of the audio source.
Target	Object for which the audio is played. If null, the audio is heard on the background of the scene.
Pitch	If enabled, the pitch of the audio can be customized. To animate the pitch, enable ⏱, and then define the following: • Duration: how long the animation lasts. • Curve: Animation curve.
Distance	If enabled, the audio becomes 3D, and the distance is the maximum distance from which you can hear the audio. To animate the distance, enable ⏱, and then define the following: • Duration: how long the animation lasts. • Curve: Animation curve.
Loop	If enabled, the audio is reproduced in a loop (this parameter is invisible if the action is not Async).

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

2.13.4.3 *Camera*

The Camera group contains the following actions:

- Camera Follow Target
- Move Camera
- Set Camera Orbit Target

2.13.4.3.1 Camera Follow Target

The Camera Follow Target action allows you to set the camera to follow one object in real time when the target is moving.

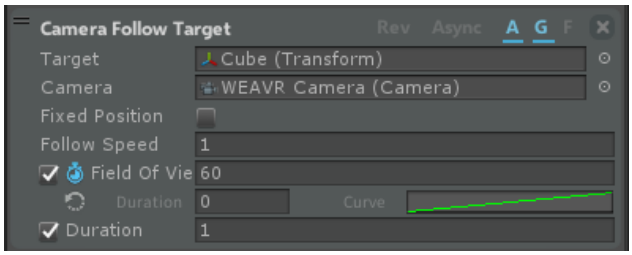


Figure 148. Camera Follow Target in the Step Inspector window

Settings:

Setting Name	Description
Target	Object to be followed by the camera.
Camera	Camera to follow the target.
Fixed Position	If enabled, the camera keeps its current position and rotates only to observe the target from its position.
Follow Speed	Speed of the camera when following its target (in m/s).
Field of View	If enabled, the field of view of the camera can be customized. To animate the distance, enable ⏱, and then define the following: • Duration: how long the animation lasts. • Curve: Animation curve.
Distance	If enabled, the audio becomes 3D, and the distance is the maximum distance from which you can hear the audio. To animate the distance, enable ⏱, and then define the following: • Duration: how long the animation lasts. • Curve: Animation curve. To revert the Field of View on exit, click ↶.
Duration	Time to follow the target (in sec).

2.13.4.3.2 Move Camera

The Move Camera action allows you to move the WEAVR Camera from one position to the other.

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

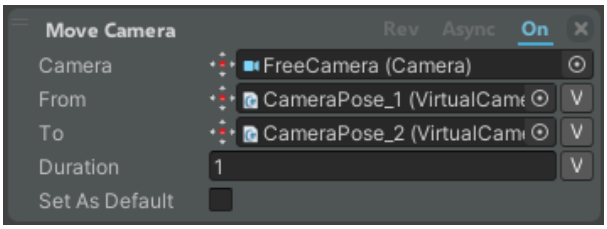


Figure 149. Move Camera properties in the Step Inspector window

Settings:

Setting Name	Description
Camera	Camera to be moved.
From	Initial point from where to start moving the camera (if empty, the current position of the camera is taken). The camera pose is created by the Camera Manager window.
To	Destination point of the camera that is created using the Camera Manager window.
Duration	Duration of the camera movement.
Set as Default	If enabled, the "To" position is used to reset the camera view in the scene. To reset a position, click the Reset Position button in the WEAVR Player.

2.13.4.3.3 Set Camera Orbit Target

The Set Camera Orbit Target action allows you to set the target around which the camera can zoom and rotate around its pivot point in OpS procedures. This action is associated with the Camera Orbit button that enables or disables the camera movement.
You can configure the camera movement parameters in the Camera Orbit component.

Figure 150. Set Camera Orbit Target properties in the Step Inspector window

Settings:

Setting Name	Description
Target	Game object around which the camera zooms and rotates.
Center on Target	If enabled, the camera is centered on the target.
Camera Distance	If enabled, the distance from the camera to the target.
Focus Type	Hint of the camera on which to focus: Bounds or Pivot Point.
Is Default Target	If enabled, focuses on this target as on a default one.

2.13.4.4 *Canvas*

The Canvas group contains the following actions:

- Set Text
- Set Text Color
- Set Image
- Play Video
- Advanced

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

2.13.4.4.1 Set Text

The Set Text action allows you to set the text and color of any canvas in the scene.

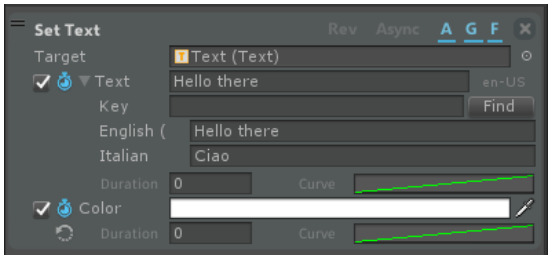


Figure 151. Set Text

Settings:

Setting Name	Description
Target	Target text component in which to change the text.
Text	If enabled, the text can be changed.
Key:	Subject of further development.
Find	Subject of further development.
English ..	Different text can be written according to procedure languages. If no other text is set, the main text is taken. To animate the text appearance, click ⏱, and then define the following: • Duration: how long the animation lasts. • Curve: Animation curve.
Color	If enabled, you can customize the color that is applied to the text. To animate the color change, enable ⏱, and then define the following: • Duration: How long the animation lasts. • Curve: Animation curve. To revert the color on exit, click ↶.

2.13.4.4.2 Set Text Color

The Set Text Color action allows you to set the color of the text of any canvas.

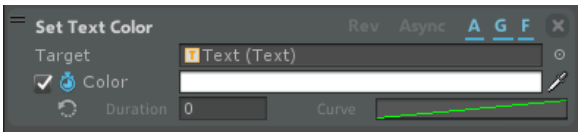


Figure 152. Set Text Color

Settings:

Setting Name	Description
Target	Target Text component in which to change the text.
Color	If enabled, allows customizing the color that is applied to the text. To animate the color change, enable ⏱, and then define the following: • Duration: How long the animation lasts. • Curve: Animation curve. To revert the color on exit, click ↶.

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

2.13.4.4.3 Set Image

The Set Image action allows you to set the image sprite and color of any canvas.

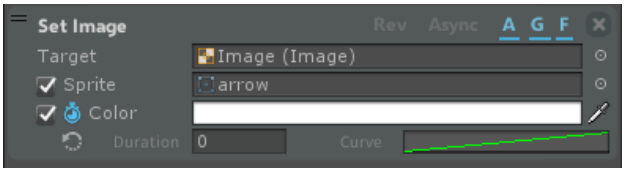


Figure 153. Set Image

Settings:

Setting Name	Description
Target	Target Image component to change the image.
Sprite	If enabled, allows changing the sprite to visualize on the target.
Color	If enabled, allows customizing the color that is applied to the sprite. To animate the color change, enable ⏱, and then define the following: • Duration: How long the animation lasts. • Curve: Animation curve. To revert the color on exit, click ↶.

2.13.4.4.4 Play Video

The Play Video action allows you to set the video to be played on the Video Player of any canvas of the scene.

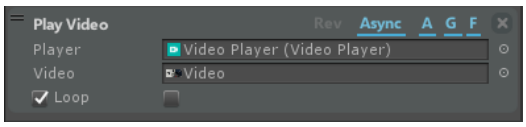


Figure 154. Play Video

Settings:

Setting Name	Description
Player	Video Player in which to play the video.
Video	Video clip to play.
Loop	If enabled, allows playing the video in a loop.

2.13.4.4.5 Advanced

Set Text Expert

The Set Text Expert action allows you to customize the Text of any canvas in the scene.

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

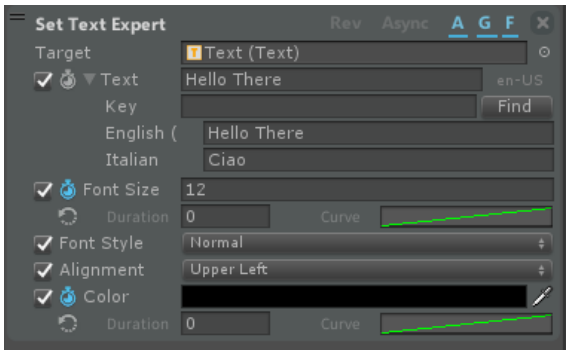


Figure 155. Set Text Expert

Settings:

Setting Name	Description
Target	Target Text component in which to change the text.
Text	If enabled, the text can be changed.
Key:	Subject of further development.
Find	Subject of further development.
English ..	Different text can be written according to procedure languages. If no other text is set, the main text is taken. To animate the text appearance, click ⏱, and then define the following: • Duration: how long the animation lasts. • Curve: Animation curve.
Font Size	If enabled, allows customizing the font size that is applied to the text. To animate the size change, enable ⏱, and then define the following: • Duration: How long the animation lasts. • Curve: Animation curve. To revert the size on exit, click ↶.
Font Style	If enabled, allows changing the font style.
Alignment	If enabled, allows changing the alignment.
Color	If enabled, allows customizing the color that is applied to the text. To animate the color change, enable ⏱, and then define the following: • Duration: How long the animation lasts. • Curve: Animation curve. To revert the color on exit, click ↶.

Set Image Expert

The Set Image Expert action allows you to customize the Image of any canvas in the scene.

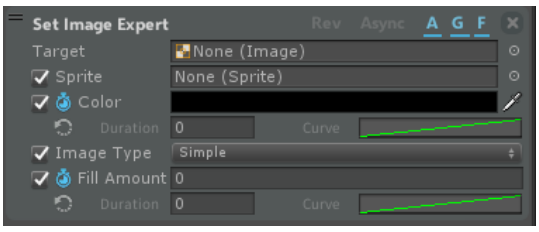


Figure 156. Set Image Expert

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

Settings:

Setting Name	Description
Target	Target Image component to change the image.
Sprite	If enabled, allows changing the sprite to visualize on the target.
Color	If enabled, allows customizing the color that is applied to the sprite. To animate the color change, enable ⏱, and then define the following: • Duration: How long the animation lasts. • Curve: Animation curve. To revert the color on exit, click ↶.
Image Type	If enabled, allows changing the image type.
Fill Amount	If enabled, allows customizing the fill amount of the sprite (if the Image Type property supports filling the sprite). To animate the fill amount, enable ⏱, and then define the following: • Duration: How long the animation lasts. • Curve: Animation curve. To revert the fill amount on exit, click ↶.

Play Video Expert

The Play Video Expert action allows you to customize the Video of the Video Player in any canvas in the scene.

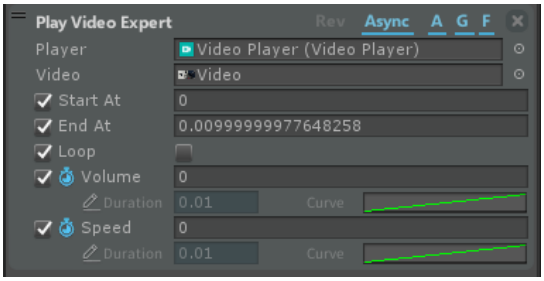


Figure 157. Play Video Expert

Settings:

Setting Name	Description
Player	Video Player in which to play the video.
Video	Video clip to play.
Start At	Starting point of the video (in sec). The video proper duration should be considered.
End At	Ending point of the video (in sec). The video proper duration should be considered.
Loop	If enabled, allows playing the video in a loop.
Volume	If enabled, allows customizing the volume of the audio in a video clip. To animate the volume change, enable ⏱, and then define the following: • Duration: How long the animation lasts. • Curve: Animation curve. To revert the volume on exit, click ↶.

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

Speed	If enabled, allows customizing the speed of the video clip. To animate the speed change, enable ⏱, and then define the following: • Duration: How long the animation lasts. • Curve: Animation curve. To revert the speed on exit, click ↶.

2.13.4.5 *Control Flow*

The Control Flow group contains the following actions:

- Wait Time
- Wait All Async Actions
- Stop All Async Actions
- Set Variable
- Set Variable From Object

2.13.4.5.1 Wait Time

The Wait Time action waits a certain amount of time between two actions in a sequence of one step.

Figure 158. Time properties in the Step Inspector window

Settings:

Setting Name	Description
Wait Time	Define the waiting time (in seconds).

2.13.4.5.2 Wait All Async Actions

The Wait All Async Actions action waits until all the parallel actions end. It is mainly used as an exit action.

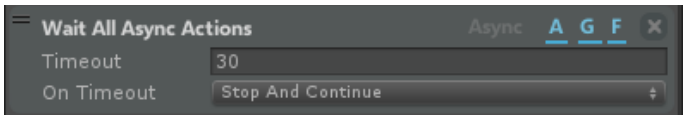


Figure 159. Wait All Async properties in the Step Inspector window

Settings:

Setting Name	Description
Timeout	Define the amount of time (in sec) before the On Timeout property is performed.
On Timeout	Define the action to be performed when the Timeout property expires: • Stop and Continue: waits until all asynchronous actions stop before continuing with the procedure. • Continue: does not wait until all asynchronous actions stop; continues the procedure.

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

2.13.4.5.3 Stop All Async Actions

The Stop all Async Actions action stops all asynchronous action that are executed in parallel. It is mainly used as an exit action.

Figure 160. Stop All Async properties in the Step Inspector window

2.13.4.5.4 Set Variable

The Set Variable action creates or sets, if already existent, a variable according to the indicated value. All the Variables created in the scene and procedures are stored in the Variable window (Section 2.13.7).

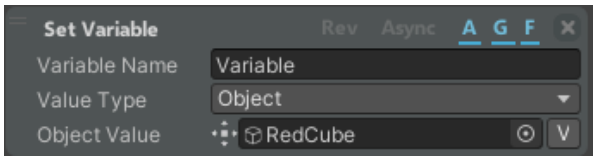


Figure 161. Set Variable

Settings:

Setting Name	Description
Variable Name	Enter the name of a variable.
Value Type	Select the type of a variable.
… Value	Define the variable value. To use the other variable value, click V.

2.13.4.5.5 Set Variable From Object

The Set Variable From Object action creates or sets, if already existent, a variable according to the status of an object in a scene. All the Variables created in the scene and procedures are stored in the Variable window (Section 2.13.7).

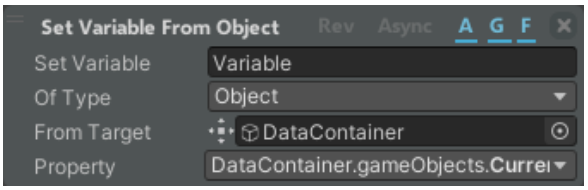


Figure 162. Set Variable From Object

Settings:

Setting Name	Description
Set Variable	Enter the name of the variable.
Of Type	Define the variable type.
From Target	Define the GameObject in a scene.
Property	Select the property of the component of the Target Object according to which the variable is set.

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

2.13.4.6 *Hints*

The Hints group contains the following actions:

- Show Simple Billboard
- Show Billboard
- Outline Object
- Hide Billboard and Outline
- Show Notification
- Advanced

> NOTE
> The billboard settings are explained in the manual. The same parameters can be customized for a single action in Show Billboard Expert.

2.13.4.6.1 Show Simple Billboard

The Show Simple Billboard action allows you to set the billboard text (and other parameters according to the used Billboard Sample), the outline, and the target from which it pops up. The Billboard Sample that is used in the action is the one referenced in the Billboard Manager (Section 2.8.1).

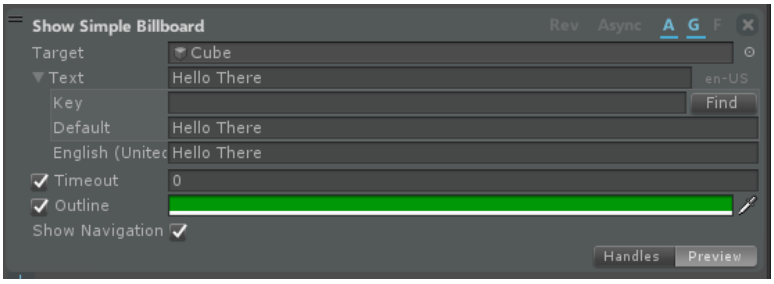


Figure 163. Show Simple Billboard

Settings:

Setting Name	Description
Target	Define the object from which the billboard and/or the outline will be shown.
Text	Enter the text for the billboard.
Key:	Subject of further development.
Find	Subject of further development.
English ..	Different text can be written according to procedure languages. If the text is null, the main text is taken.
Timeout	If enabled, allows defining the amount of time to show the billboard/outline before disappearing.
Outline	If enabled, allows defining the outline color for the target object. If not enabled, the object is outlined until another command removes it.
Show Navigation	If enabled, shows the navigation hint to the target object.
Preview	If enabled, shows the preview of the billboard on the target object in the scene (Figure 164).

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

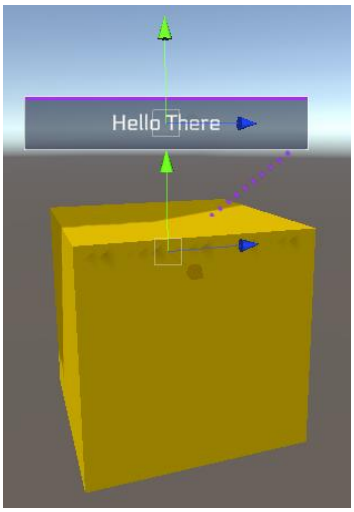


Figure 164. Billboard Handles

2.13.4.6.2 Show Billboard

The Show Billboard action allows you to set the billboard text (and other parameters according to the used Billboard Sample, see Section 2.8.1), the outline, and the target from which it pops up.

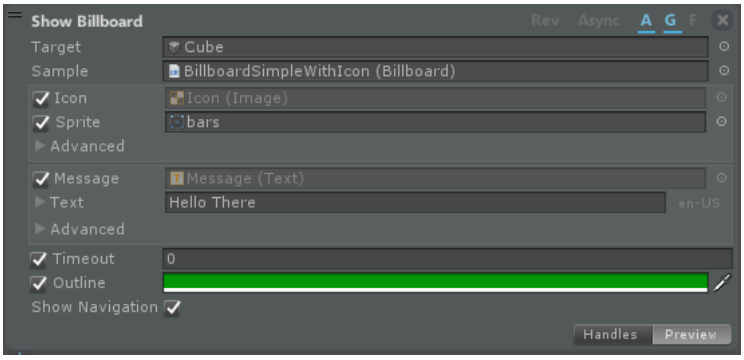


Figure 165. Show Billboard

Settings:

Setting Name	Description
Target	Define the object from which the billboard and/or the outline will be shown.
Sample	Define the Billboard sample that will be taken as a reference.
Image, Text…	Define parameters of canvas elements to be customized. Parameters depend on the indicated Sample.
Timeout	If enabled, allows defining the amount of time to show the billboard/outline before disappearing.
Outline	If enabled, allows defining the outline color for the target object. If not enabled, the object is outlined until another command removes it.
Show Navigation	If enabled, shows the navigation hint to the target object.
Preview	If enabled, shows the preview of the billboard on the target object in the scene.
Handles	If enabled, the Start Point and End Point transforms appear in the scene. You can move both transforms in the scene to define the correct location of the billboard.

-

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

2.13.4.6.3 Outline Object

The Outline Object action allows you to set the color of the outline of a target. The outline settings are explained in the manual.

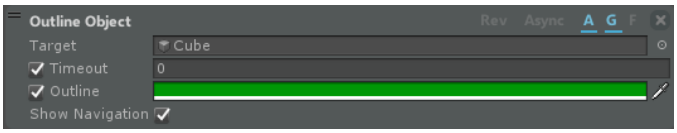


Figure 166. Outline Objects

Settings:

Setting Name	Description
Target	Define the object that will show the outline.
Timeout	If enabled, allows defining the amount of time to show the outline before disappearing.
Outline	If enabled, allows defining the outline color for the target object. If not enabled, the object is outlined until another command removes it.
Show Navigation	If enabled, shows the navigation hint to the target object.

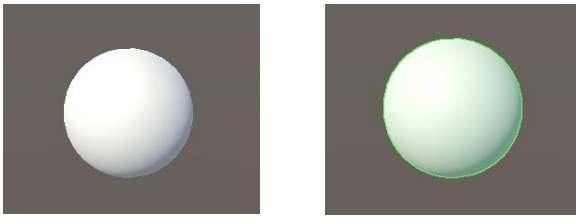


Figure 167. Example of outlining action

2.13.4.6.4 Hide Billboard and Outline

The Hide Bllboard and Outline action hides the billboard and/or outline of the target object .

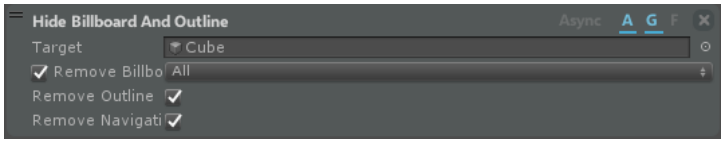


Figure 168. Hide Billboard and Outline

Settings:

Setting Name	Description
Target	Define the object from which to remove the outline.
Remove Billboard	If enabled, billboards are removed from the target object according to the selected filter: • All: all billboards are removed. • By Sample Type: billboards based on the selected Billboard Sample are removed. ▪ Billboard Sample: focused Billboard Sample whose generated billboard is removed. • Last One: last generated billboard is removed.
Remove Outline	If enabled, the outline of the target object is removed.
Remove Navigation	If enabled, removes the navigation hint to the target object.

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

2.13.4.6.5 Show Notification

The Show Notification action allows you to set a message type and text that will appear as a notification attached to the controller for the defined duration.

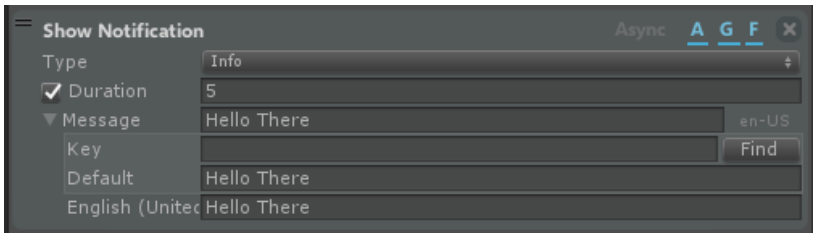


Figure 169. Show Notification

Settings:

Setting Name	Description
Type	Define the type of notification that will appear: • Info: information message (Figure 170). • Warning: warning message (Figure 170). • Error: error message (Figure 170). • Custom: customized message. ▪ Image: image to be used for a custom notification.
Duration	If enabled, define the time to show the notification message.
Message	Enter the text message of the notification.
Key:	Subject of further development.
Find	Subject of further development.
English ..	Different text can be written according to procedure languages. If the text is null, the main text is taken.

Figure 170. Notification message, Info type (left) Warning type (centre) and Error type (right)

2.13.4.6.6 Advanced

Show Billboard Expert

The Show Billboard Expert action allows you to set the billboard on the target object with further customizations of the Billboard Sample parameters (the available settings which you can customize are explained in Section 2.8.1).

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

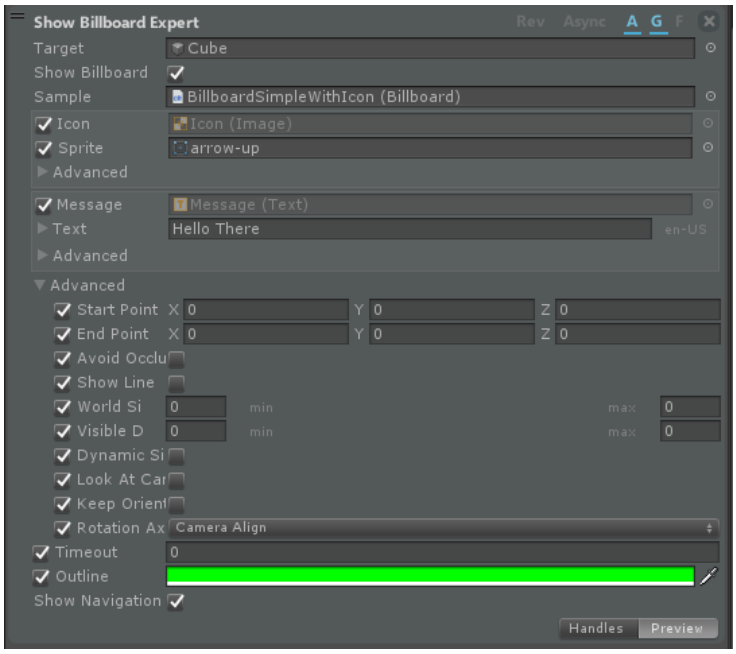


Figure 171. Show Billboard Expert

Settings:

Setting Name	Description
Target	Define the object from which the billboard and/or the outline will be shown.
Show Billboard	If enabled, the billboard appears on the object.
Sample	Define the Billboard sample that will be taken as a reference.
Image, Text…	Define parameters of canvas elements to be customized. Parameters depend on the indicated Sample. Section 2.13.4.4.5 explains in detail the element settings.
Start Point Offset	If enabled, the start position of the billboard line can be set in respect to the transform of the focused object.
End Point Offset	If enabled, the end position of the billboard line can be set in respect to the transform of the focused object.
World Size Limit	Controls the maximum and minimum billboard dimensions.
Visible Distance	Define the maximum and minimum distance of the camera to see the billboard.
Dynamic Size	If enabled, the billboard dimensions change according to the camera distance from them.
Look at Camera	If enabled, the orientation of the billboard moves with the camera location.
Keep Orientation	If enabled, the billboard keeps the same orientation to the focused object and does not rotate.
Rotation Axis	If enabled, allows selecting the rotation of the billboard: ▪ Up: billboard rotates only on the vertical axis. ▪ Free: billboard rotates on three axes. ▪ Camera Align: billboard rotates in alignment with the camera view.
Timeout	If enabled, allows defining the amount of time to show the billboard/outline before disappearing.
Outline	If enabled, allows defining the outline color for the target object. If not enabled, the object is outlined until another command removes it.

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

Show Navigation	If enabled, shows the navigation hint to the target object.
Preview	If enabled, shows the preview of the billboard on the target object in the scene.
Handles	If enabled, the Start Point and End Point transforms appear in the scene. You can move both transforms in the scene to define the correct location of the billboard.

Hide Billboard Expert

The Hide Billboard Expert action hides the billboard and/or outline of the target object.

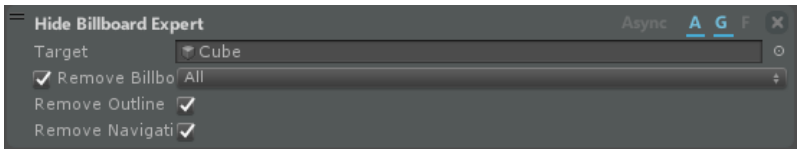


Figure 172. Hide Billboard and Outline

Settings:

Setting Name	Description
Target	Define the object from which to remove the outline.
Remove Billboard	If enabled, billboards are removed from the target object according to the selected filter: • All: all billboards are removed. • By Sample Type: billboards based on the selected Billboard Sample are removed. ▪ Billboard Sample: focused Billboard Sample whose generated billboard is removed. • Last One: last generated billboard is removed.
Remove Outline	If enabled, the outline of the target object is removed.
Remove Navigation	If enabled, removes the navigation hint to the target object.

2.13.4.7 *Object*

The Object group contains the following actions:

- Toggle Object
- Toggle Component
- Toggle All Components
- Set Generic Value
- Set Material
- Set Color
- Set Texture
- Set Color and Texture
- Call Method
- Advanced

2.13.4.7.1 Toggle Object

The Toggle Object action enables or disables the target GameObject in a scene.

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

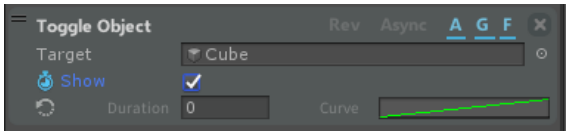


Figure 173. Toggle Object

Settings:

Setting Name	Description
Target	Define which the object to show or hide.
Show	If enabled, shows the object.
⏱	If enabled, animates the action: • Duration: how long the animation lasts. • Curve: animation curve.
↶	If enabled, reverts the action on exit.

2.13.4.7.2 Toggle Component

The Toggle Component enables or disables the indicated component of the target object.

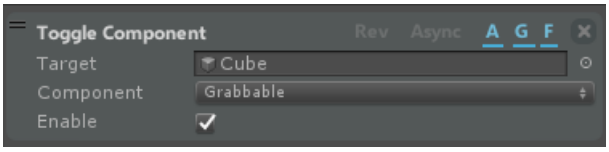


Figure 174. Toggle Component

Settings:

Setting Name	Description
Target	Define the object in which to toggle the component.
Component	Select the component to be toggled.
Enable	If enabled, activates the component.

2.13.4.7.3 Toggle All Components

The Toggle All Components action enables or disables the selected component in all objects that have this component. Objects of all the scenes are considered.

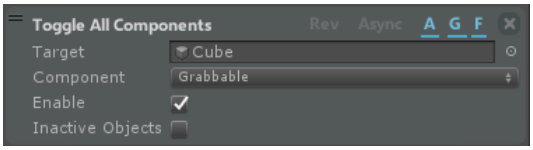


Figure 175. Toggle All Components

Settings:

Setting Name	Description
Target	Define a sample target in which to toggle the component.
Component	Select the component to be toggled.
Enable	If enabled, activates the component for all objects that have it in a scene.
Inactive Objects	If enabled, the component is considered in inactive objects as well.

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

NOTE
The Interaction Controller Group component (Section 2.9.3.1) can be used also to enable or disable the Interaction Controller behavior (Section 2.9.3.2) of the group of listed objects.

2.13.4.7.4 Set Generic Value

The Set Generic Value action sets different attribute values of the target object. The values are property of the component of the target object. WEAVR Interactions section explains all the component values and their meanings.

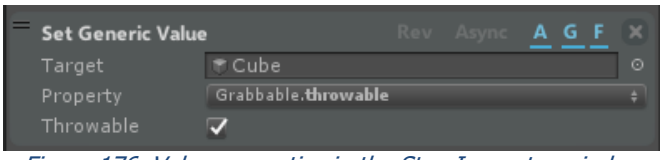


Figure 176. Value properties in the Step Inspector window

Settings:

Setting Name	Description
Target	Define the object in which to change the value.
Property	Select the property to set. The list pf available properties depends on the components associated to the GameObject.
Value	If enabled, sets the value of the selected property. The value can be a bool, float, string, or other.

2.13.4.7.5 Set Material

The Set Material action sets the material of the target object.

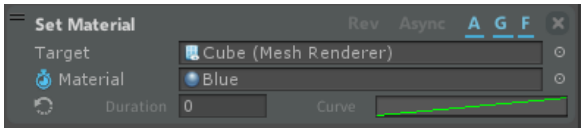


Figure 177. Set Material

Settings:

Setting Name	Description
Target	Define the object in which to change the material.
Material	Define the material to be applied to the object.
⏱	If enabled, animates the action: • Duration: how long the animation lasts. • Curve: animation curve.
↶	If enabled, reverts the material on exit.

2.13.4.7.6 Set Color

The Set Color action sets the color of the target object.

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

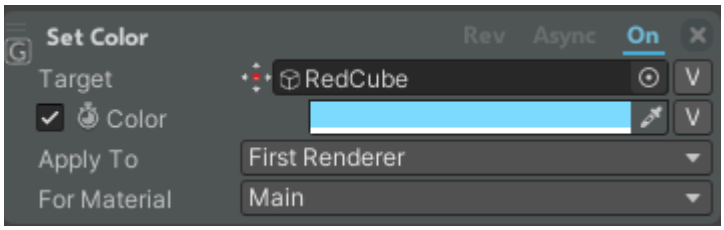


Figure 178. Set Color

Settings:

Setting Name	Description
Target	Define the renderer for which to change the color.
Color	If enabled, applies the selected color to the main material of the target renderer.
⏱	If enabled, animates the action: • Duration: how long the animation lasts. • Curve: animation curve.
↶	If enabled, reverts the color on exit.
Apply To	Applies the color to the selected mesh: • First renderer: to the first mesh that is found in the Object or its children. • First Level Children: to all children of the first level. • All Children: to all children in the hierarchy.
For Material	Define the material to which to apply the color: • Main: the first listed material. • At Index or Main: the material with the corresponding index or the first existing material. • At Index or Last One: the material with the corresponding index or the last existing material . • At Index or None: the material with the corresponding index or existing to none • All Materials: all the listed materials.

2.13.4.7.7 Set Texture

The Set Texture action sets the texture of the target object.

Figure 179. Set Texture

Settings:

Setting Name	Description
Target	Define the renderer for which to change the texture.
Texture	If enabled, changes the texture of the target renderer.

2.13.4.7.8 Set Color and Texture

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

The Set Color and Texture action sets the color and texture of the target object.

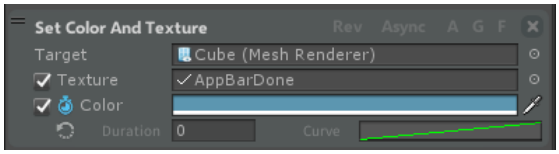


Figure 180. Set Color and Texture

Settings:

Setting Name	Description
Target	Define the renderer for which to change the color and texture.
Texture	If enabled, changes the texture of the target renderer.
Color	If enabled, applies the selected color to the main material of the target renderer. To animate the action, enable ⏱, and then define the following: • Duration: How long the animation lasts. • Curve: Animation curve. To revert the color on exit, click ↶.

2.13.4.7.9 Call Method

The Call Method action expands the Set Generic Value actions because it calls the methods of the Target component.

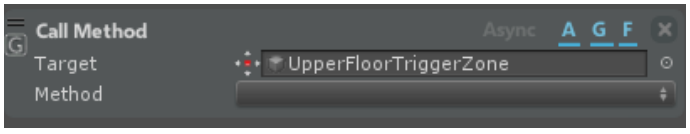


Figure 181. Call Method

Settings:

Setting Name	Description
Target	Define the target GameObject.
Method	Select which specific method of the target GameObject to call.

2.13.4.7.10 Advanced

Set Color and Texture Expert

The Set Color and Texture Expert action allows you to customize the color and texture of the target object.

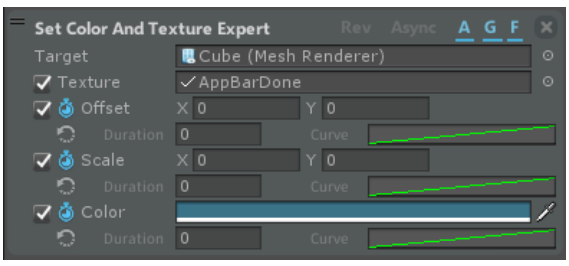


Figure 182. Set Color and Texture Expert

Settings:

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

Setting Name	Description
Target	Define the renderer for which to change the color and texture.
Texture	If enabled, changes the texture of the target renderer.
Offset	If enabled, applies the offset to the main texture of the target renderer. To animate the action, enable ⏱, and then define the following: • Duration: How long the animation lasts. • Curve: Animation curve. To revert the offset on exit, click ↶.
Scale	If enabled, applies the scale to the main texture of the target renderer. To animate the action, enable ⏱, and then define the following: • Duration: How long the animation lasts. • Curve: Animation curve. To revert the scale on exit, click ↶.
Color	If enabled, applies the selected color to the main material of the target renderer. To animate the action, enable ⏱, and then define the following: • Duration: How long the animation lasts. • Curve: Animation curve. To revert the color on exit, click ↶.

2.13.5 Exit Conditions

An Exit Condition is an evaluation element that needs to be true to trigger a navigation event and go on with the procedure flow.
Conditions can be combined to get more complex logic expressions. You can assign Exit Conditions to steps. The major component settings are explained in Section 2.9, while the conditions are explained in the following paragraphs.
Exit Conditions contain the following conditions:

- Generic Value
- Generic Comparison
- Condition Valid For
- Distance Between Object
- Is Execute Mode
- Object Is in Area
- Object Is Active
- Object Is Visible by Camera
- Value Changed
- Visually Inspect
- Simple Visual Inspection
- Query Method
- Button Is Pressed
- Get Variable
- Variable Value Changed
- Specific OpS Exit Conditions

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

2.13.5.1 *Generic Value*

The Generic Value exit condition depends on the components that are associated with the target object. The condition takes into consideration properties and available values of the referenced component.
Figure 183 shows an example of Generic Condition. The condition becomes true when the object Cube is grabbed (the referenced component property of the object is Grabbable, Section 2.9.2.1).

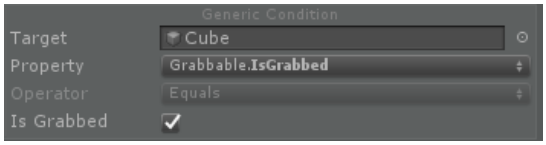


Figure 183. Generic Condition, example of an exit condition

Settings:

Setting Name	Description
Target	Define the object.
Property	Select the component and its target property.
Operator	Select the arithmetical operator that is used to perform the comparison.
Value	Define the value to compare.

2.13.5.2 *Generic Comparison*

The Generic Comparison exit condition compares property values of two target objects.

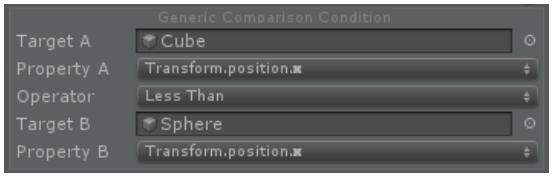


Figure 184. Generic Comparison Condition

Settings:

Setting Name	Description
Target A	Define the first object from which to take the value for the comparison.
Property A	Select the property value of the first comparison object.
Operator	Select the arithmetical operator that is used to perform the comparison.
Target B	Define the other object from which to take the value for the comparison.
Property B	Select the property value of the second comparison object.

2.13.5.3 *Condition Valid For*

Condition Valid For is a container for Exit Conditions that should persist for a certain duration before changing the step.

Figure 185. Condition Valid For, exit condition

Settings:

Setting Name	Description

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

Valid For	Define for how long (in sec) the conditions should be valid before changing the step.
Progressive	If highlighted, the condition slowly resets its progress to 0 when the condition child is false.
+	Adds an exit condition.

2.13.5.4 *Distance Between Object*

The Distance Between Object exit condition checks the distance between two object transforms.

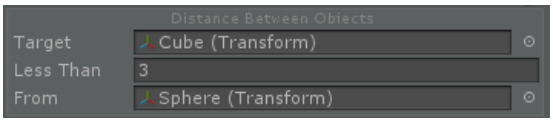


Figure 186. Distance Between Objects

Settings:

Setting Name	Description
Target	Define the object A from which start calculating the distance.
Less Than	Define the maximum distance between the two objects to make the condition true.
From	Define the object B to which calculate the distance.

2.13.5.5 *Is Execute Mode*

The Is Execute Mode condition checks the play mode of the procedure: automatic, guided, or feedback. It does not involve a target object.

Figure 187. Current Execution Mode, exit condition

Settings:

Setting Name	Description
Mode	Define the procedure mode to make the condition true.

2.13.5.6 *Object is in Area*

The Object Is in Area exit condition becomes true if the target object is in the referenced area (Collider).

Figure 188. Is in Area, exit condition

Settings:

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

Setting Name	Description
Target	Define the object that should be in the area.
Area	Define the area to check against (collider).
When	Select when to trigger the condition: Enters Area or Leaves Area.
How	Select what to check to consider the condition true: • Pivot: pivot point of the target. • Boundig Box: bounding box of the target intersects the area. • Full Body: full body of the target is inside the area.

2.13.5.7 *Object Is Active*

The Object Is Alive exit condition checks if the target object is/is not active in hierarchy.

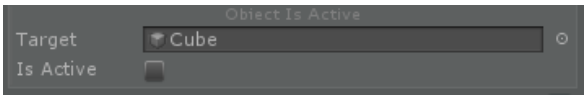


Figure 189. Object Is Active

Settings:

Setting Name	Description
Target	Define the object to check.
Is Active	If enabled, the exit condition becomes active.

2.13.5.8 *Object Is Visible by Camera*

The Object Is Visible by Camera condition is true if the target object is/is not visible by the indicated camera.

Figure 190. Is Visible by Camera, exit condition

Settings:

Setting Name	Description
Target	Define the object to be visible.
Camera	Define the camera that should see the object. Otherwise, all cameras in the scene are checked.
Visible	If enabled, the object becomes visible.

2.13.5.9 *Value Changed*

The Value Changed condition checks if the indicated property value, proper of the target object components, is changed.

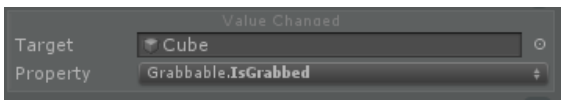


Figure 191. Value Changed, exit condition

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

Settings:

Setting Name	Description
Target	Define the object whose value will be checked on changes.
Property	Select the property value whose change is checked.

2.13.5.10 *Visually Inspect*

The Visually Inspect condition becomes true if the target is visualized by the defined camera according to the indicated time and distance.

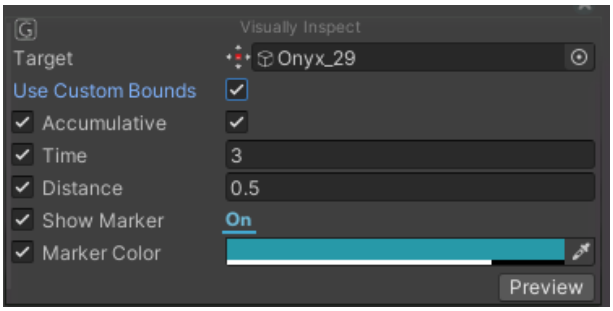


Figure 192. Visually Inspected, exit condition

Settings:

Setting Name	Description
Target	Define the object to be inspected from the player.
Use Custom Bounds	If enabled, the inspection marker (Figure 193) can be customized according to the preferred position and dimension; otherwise the inspector marker depends on the Target bounding box centre and dimensions: • Preview: if enabled, the inspector marker is visualized in the scene (Figure 193). • Handles: if enabled, the inspector marker can be moved to the desired position and scaled directly from the scene. Furthermore, the inspection distance can be modified from the scene according to the radius of the sphere that is visualized around the Target (Figure 193). • Reset: if enabled, the position of the inspector marker is set to the centre of the Target.
Accumulative	If enabled, the inspection time is not reset to zero when the visual inspection is lost before the minimum time set.
Time	Define the required inspection time.
Distance	Define the minimum distance from the Inspector camera to the center of the Target.
Show Marker	If enabled, the inspector marker is showed according to the highlighted procedure mode.
Marker Color	If enabled, allows selecting the Inspector Marker color

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

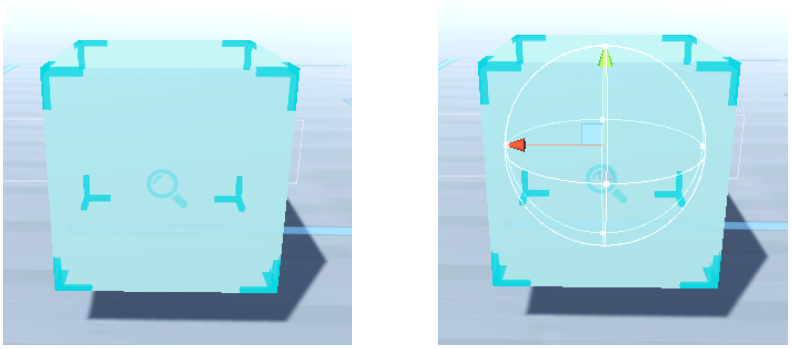


Figure 193. Inspection marker preview (left) and Inspection marker Handles (right)

2.13.5.11 *Simple Visual Inspection*

The Simple Visual Inspection is a simplified version of the Visually Inspect condition. This condition is true if the target is visualized by the defined camera according to the indicated time and distance.

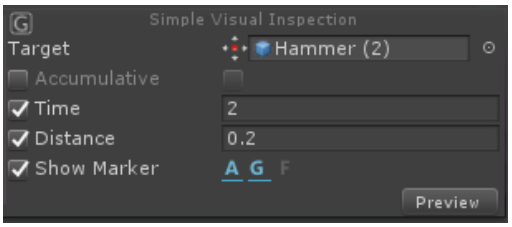


Figure 194. Simple Visual Inspection

Settings:

Setting Name	Description
Target	Define the object to be inspected from the player.
Accumulative	If enabled, the inspection time is not reset to zero when the visual inspection is lost before the minimum time set.
Time	Define the required inspection time.
Distance	Define the minimum distance from the Inspector camera to the centre of the Target.
Show Marker	If enabled, the inspector marker is showed according to the highlighted procedure mode.

2.13.5.12 *Query Method*

The Query Method exit condition depends on components associated with the target object. It takes into consideration methods and available values of the referenced component.

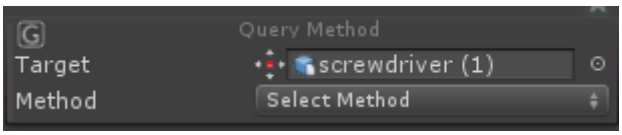


Figure 195.Query method

Settings:

Setting Name	Description
Target	Define the object from which to take the value for comparison.
Method	Select the method that is used to check the target value.

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

2.13.5.13 *Button Is Pressed*

The Button Is Pressed exit condition becomes true when you press the target Button.

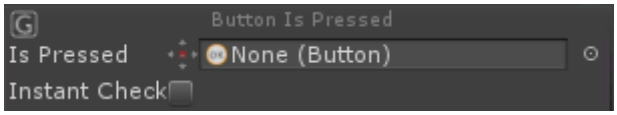


Figure 196. Button Is Pressed

Settings:

Setting Name	Description
Is Pressed	Define the button to be checked.
Instant Check	If enabled, the On Click event should be triggered more than once.

2.13.5.14 *Get Variable*

The Get Variable condition is verified when a variable value equals the target value.

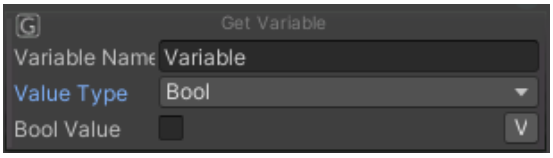


Figure 197. Get Variable

Settings:

Setting Name	Description
Variable Name	Enter the name of the variable to be interrogated.
Value Type	Select the type of the variable to be interrogated.
… Value	If enabled, the target value is used.

2.13.5.15 *Variable Value Changed*

The condition is true when the value of the indicated variable has changed.

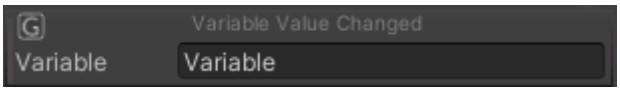


Figure 198. Variable Value Changed

Settings:

Setting Name	Description
Variable	Enter the name of the variable.

2.13.5.16 *Specific OpS Exit Conditions*

A simple OpS procedure is a linear flow of steps. You can advance the procedure by clicking the Next Step button and reproduce the previous step by clicking the Previous Step button.
Exit conditions that control these basic OpS buttons are explained in this section.

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

NOTE
How to reference the Next Step and Previous Step buttons is explained in Section 2.13.8.2.

2.13.5.16.1 Null Exit Condition

The Null Exit Condition controls if the Next Step button is clicked in case the step is Mandatory, and the exit condition is left empty,.

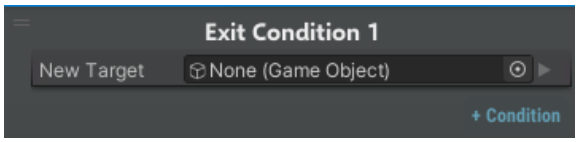


Figure 199. Null Exit Condition

2.13.5.16.2 On Previous

The On Previous condition is triggered when you click the Previous Step button.

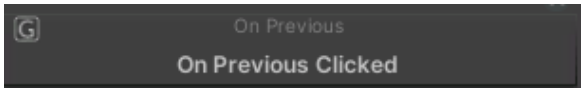


Figure 200. On Previous Clicked

2.13.6 Camera Manager

The Camera Manager wizard allows users to manage (list, check, create, or remove) camera positions to use them later in the Procedure Editor window. The poses of Virtual Cameras are used as objects of the Camera Move action (Section 2.13.4.3.2).

Camera manager
To open the Camera Manager window, go to WEAVR > Utilities > Camera Manager.

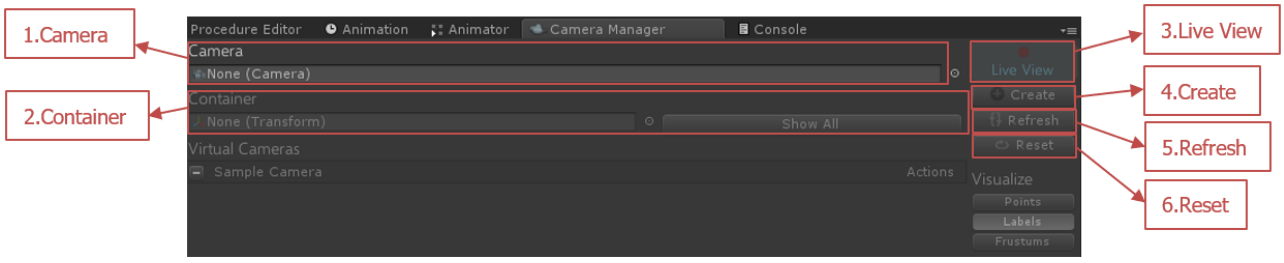


Figure 201. Camera Manager window

Settings:

Setting Name	Description
Camera	Define the camera that is used during the procedure.
Container	Define the folder in the hierarchy where Virtual Cameras will be created as child objects. Show All: when the field if empty, and a folder in the Hierarchy containing Virtual Camera exists, it references it as container folder.
Live View	Visualizes camera settings and actions.

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

Create	Creates a Virtual Camera with the current scene view and the referred Camera Pose in the Hierarchy window. To enable the button, click the Live View button.
Refresh	Refreshes the Virtual Camera List.
Reset	Resets selected virtual cameras to their initial scene views.
Controls	Configure settings that refer to all Virtual Cameras to be created. You can modify these settings later in the Virtual Camera component of the created Camera Pose in the Hierarchy (Figure 204). ○ Is Orthographic: If enabled, the Virtual Camera view is orthographic (camera renders objects uniformly with no sense of perspective). • Size: View the angle of an orthographic camera. The orthographic size is half the size of the vertical viewing volume. ○ Field of View: Define the width of the Camera's view angle, measured in degrees along the local Y axis. ○ Clipping Plane: Define the distance from the camera to the start (Near) and stop (Far) rendering.
Actions	○ To view the scene as it is seen by the Virtual Camera, click 👁. ○ To save the new position of the Virtual Camera in the scene, click ●. ○ To delete the Virtual Camera, click ✕.
Visualize	Visualizes points, labels, or frustums in the camera scene and preview (Figure 203): ○ Points: Center point of the virtual camera view. ○ Label: Camera label. ○ Frustum: Near and Far properties of the Clipping Plane setting together with the planes that are defined by the Field of View of the camera.

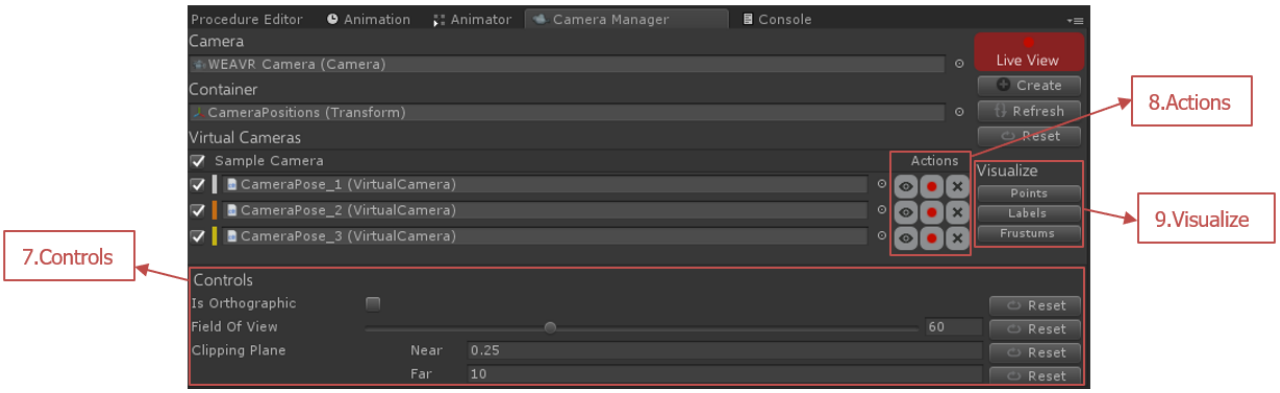


Figure 202. Example 2 of Camera Manager window

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

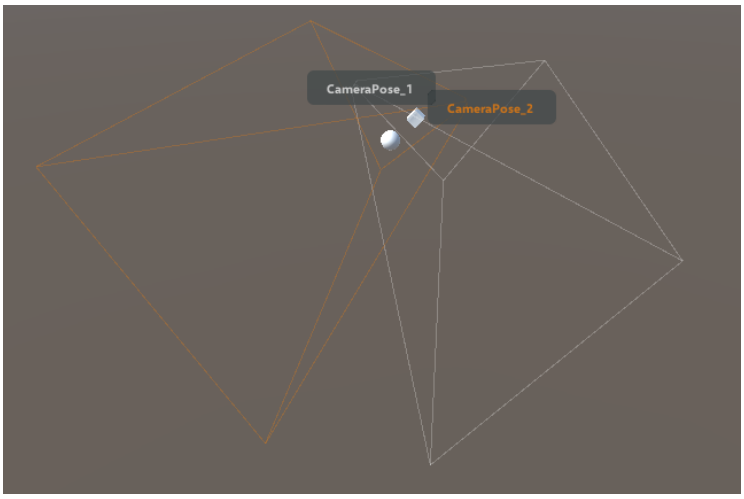


Figure 203. Example of Virtual Camera visualization in the scene

Virtual Camera component

The Virtual Camera component allows you to define the settings of Camera Positions.

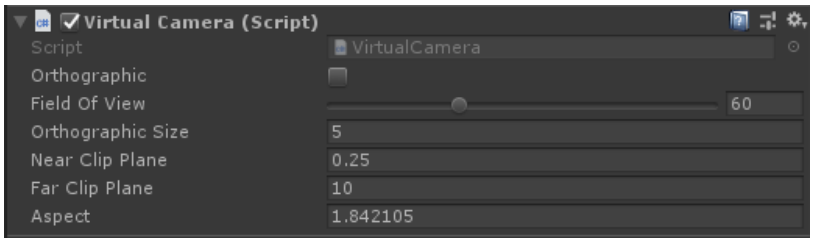


Figure 204. Virtual Camera component of the Camera Pose

Properties:

Setting Name	Description
Is Orthographic	If enabled, the Virtual Camera view is orthographic (camera renders objects uniformly with no sense of perspective).
Field of View	Define the width of the Camera view angle that is measured in degrees along the local Y axis.
Orthographic Size	Define the view angle of an orthographic camera. The orthographic size is half the size of the vertical viewing volume.
Near Clip Plane	Define distances from the camera to the start of rendering.
Far Clip Plane	Define distances from the camera to the stop of rendering.
Aspect	Define the camera aspect ratio (by default, it is the screen aspect ratio).

2.13.7 Variables

The Variables wizard is an instantiated runtime that lists all created variables and their values. You can use it to debug the variables system.
Components, Actions, and Exit Conditions involving the variable system are explained in Section 2.9.8, Section 2.13.4.5 and Section 2.13.5.

To open the Variables window, go to WEAVR > Diagnostics > Variables.

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

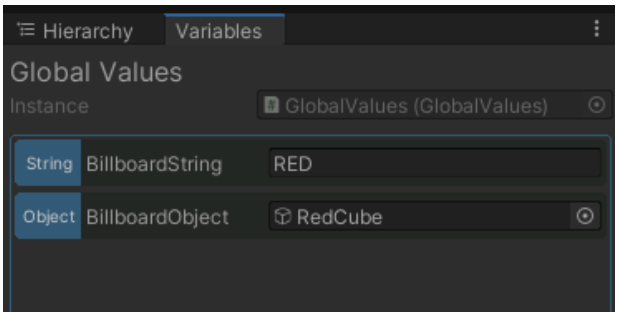


Figure 205. Variables wizard

2.13.8 Testing

2.13.8.1 *Procedure Editor Debug*

You can test how the procedure is executed if the button In Test of the Procedure Editor window is ON (Section 2.13.2.1).
As the procedure actions and steps are executed, the procedure graph is highlighted in different colors according to the outcome of single actions and the current action.
The following figure shows an example of procedure testing:

- When actions are executed, they are highlighted in green. Past actions, step transitions, and transition actions turn dark-green, while current actions are in light-green.
- When exit conditions are executed, they are highlighted in blue.
- If an action has not been executed for any reason, an exclamation mark appears at the right side of the action. The exclamation mark is colored in orange or red according to the error gravity. To read the definition of an error, hover over the exclamation mark.
- Current and previous steps are outlined in yellow, while future steps are white .

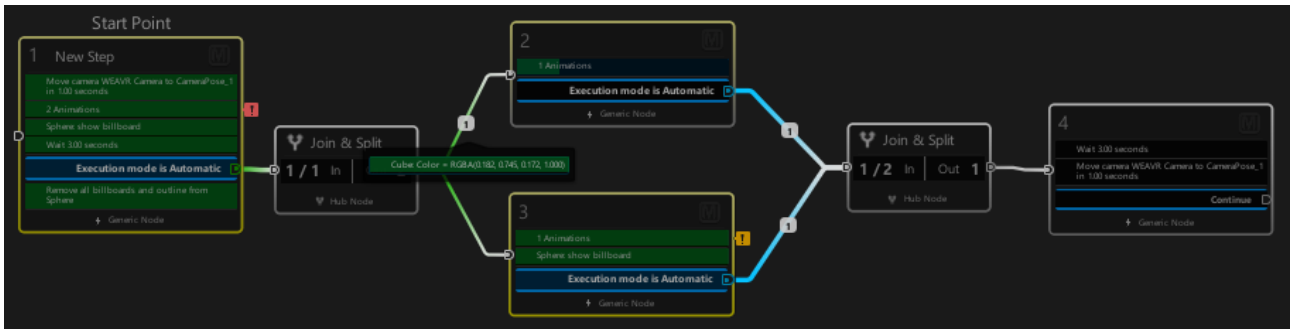


Figure 206. Example of procedure testing

2.13.8.2 *Procedure Test Panel*

Procedure Test Panel can be used in OPS applications that do not use the WEAVR Player (Section 4), to control the procedure flow. This component references the "next step" and "previous step" buttons that are used to advance in the procedure flow (Section 2.13.5.16), and the canvas texts that are used to visualize the current step information.

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

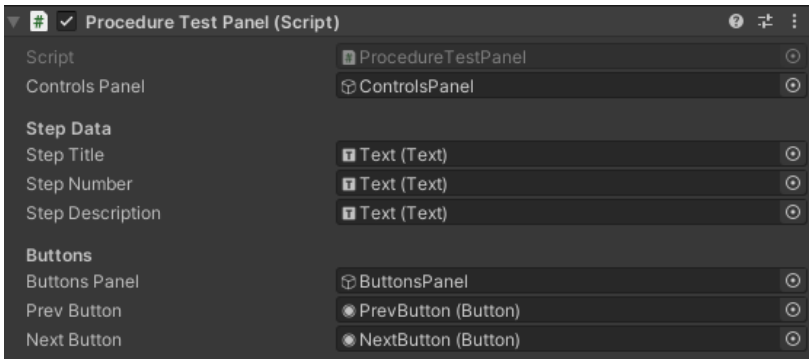


Figure 207. Procedure Test Panel

Properties:

Setting Name	Description
Controls Panel	Define the panel owner of procedure elements.
Step Title	Define the text used to visualize the current step title.
Step Number	Define the text used to visualize the current step number.
Step Description	Define the text used to visualize the current step description.
Buttons Panel	Define the parent of the "Next Step" and "Previous Step" buttons.
Prev Step	Define the button that is used to reproduce the previous step of the procedure.
Next Step	Define the button that is used to advance in the procedure flow.

NOTE
The Procedure Test Canvas prefab of a canvas that references Procedure Test Panel elements can be found in WEAVR/Essential/Assets/Prefabs/Editor.
This prefab is automatically loaded in the scene hierarchy when the procedure is tested in the Editor to reproduce the WEAVR Player canvas, but it is not part of the build.

2.13.9 Catalogues

In the Procedure Defaults window, you can set up defaults for each procedure element from the Execution Modes settings to Exit Conditions (Figure 208).
To open the Procedure Defaults window, go to WEAVR > Procedures ✇ Catalogues

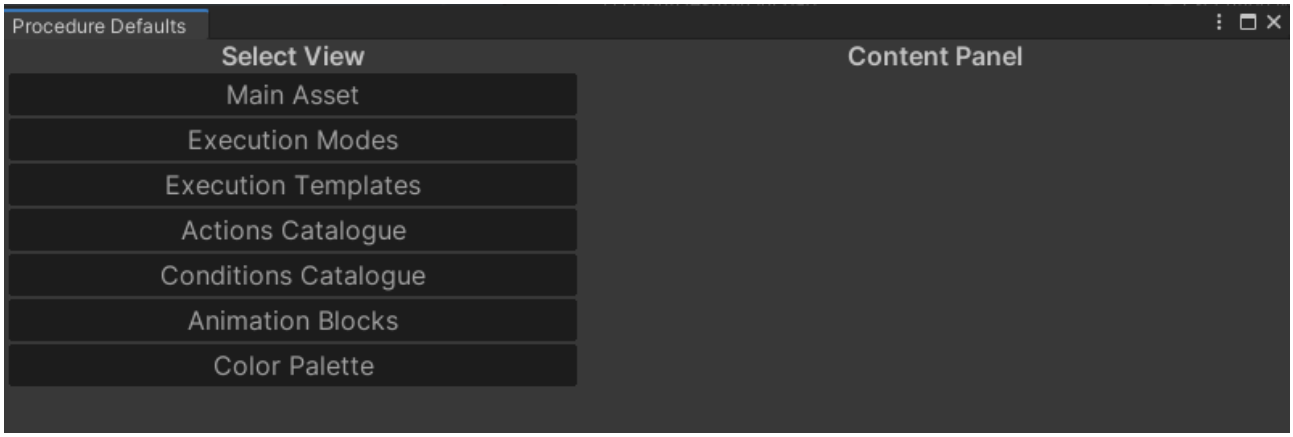


Figure 208. Procedure Defaults

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

In the Select View section, you can see a list of groups of elements that are involved in the Procedure Editor. When a group in the Select View list is highlighted, its elements are listed in the Content Panel side; you can change the default settings from the content panel. So, when the element is created in the Procedure Editor, its settings are set as required.
Figure 209 shows the defaults of animation blocks. The available elements are listed (Wait, Delta Move, etc.) and can be customized according to the required defaults.

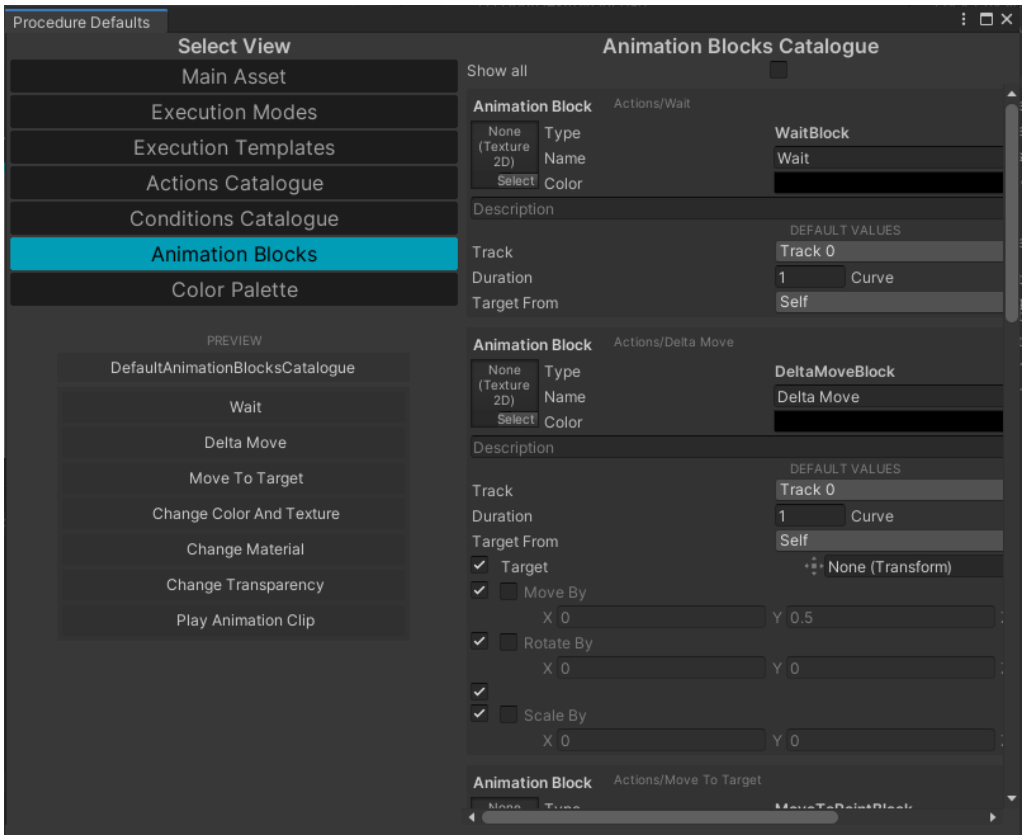


Figure 209. Procedure Defaults, Animation Blocks example

2.13.10 Procedure Upload

When the procedure is ready, you can create the application, and then either upload it to the WEAVR Player or create as a general Unity build (Section 5.1).

To upload the created procedure to the WEAVR Player (Section 4), follow these steps:

1. Make sure that the procedure is not in a test mode:
   - On the Procedure Editor toolbar (Section 2.13.2.1), disable the "In Test" button.
   - In the Hierarchy window > WEAVR > Procedure Runner (Figure 210), make sure that Start When Ready and Current Procedure are disabled.

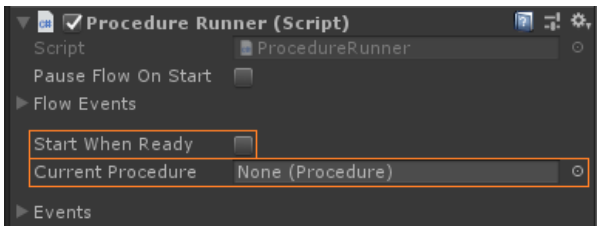


Figure 210. Procedure Runner

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

2. Select the required Platform from the Unity Build Settings (File > Build Settings). The selected platform is indicated with the Unity symbol. To change the platform, select the required platform, and then click the Switch Platform button. Make sure that platform tools are installed (For more information, refer to the Unity manual: https://docs.unity3d.com/Manual/PublishingBuilds.html).

3. Open the Upload window by clicking Upload Procedure in the Procedure Editor toolbar (Section 2.13.2.1).

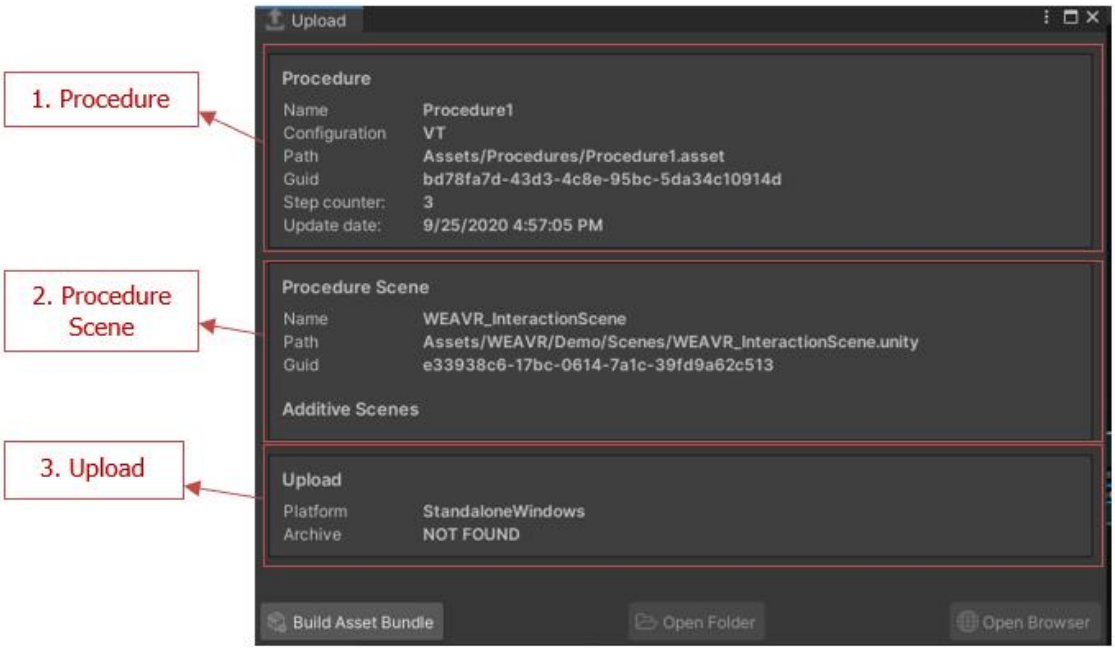


Figure 211. Upload Procedure window

Properties:

Setting Name	Description
Procedure	Data of the procedure to be deployed: • Procedure name • Procedure path • Scene identification • Number of steps of the procedure • Last update date of the procedure
Procedure Scene	Data of the procedure scene: • Scene name • Scene path • Scene identification
Upload	Information regarding the deploy: • Build platform (defined in step 2). • Folder of the asset bundle. When the procedure is built for the first time, this property is defined as NOT FOUND. It is created and referenced automatically the first time the Built is executed.

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

4. Build the procedure by clicking the Build Asset Bundle button.

A new folder is created in the project during the build to contain the necessary files. The build, in fact, is saved in*AssetBundles > [platform name] > [procedure name_procedure identification]*
The following necessary files are stored in this folder:

- Scene file
- Procedure file
- Preview.json (with information related to the procedure)
- Accessory files

Once, the procedure has been built, the Asset Bundle Folder (point 3 Figure 211) setting automatically takes the new folder reference., You can open the folder by clicking the Open Folder button (Figure 212).

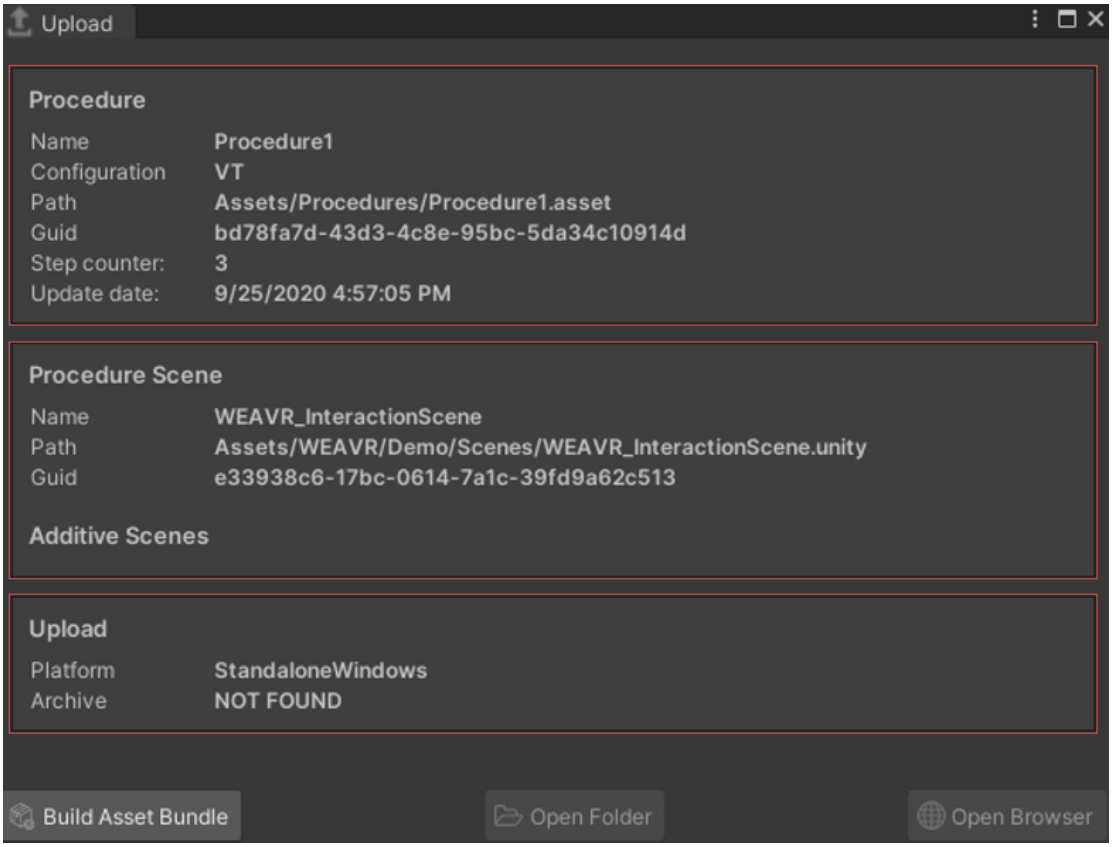


Figure 212. Upload Procedure window (2)

You can rebuild the procedure as many times as it is necessary using the Rebuild Asset Bundle button (Figure 212). The new build overwrites the old procedure.

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com |website: www.txtgroup.com | Tel: +49 30 293620

2.13.10.1 *Scene Controller (Multiscene Application)*

To load more scenes additively, add the Scene Controller component to a GameObject of the main scene.
This component allows you to load more scenes additively when required from the procedure uploaded to the WEAVR Player.

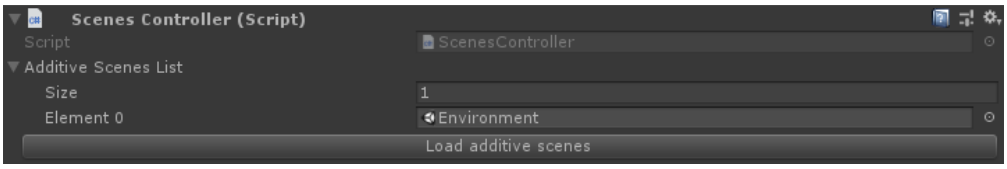


Figure 213. Scene Controller

Settings:

Setting Name	Description
Additive Scenes List	Lists scenes to be loaded additively. • Size: Number of scenes to be added to the main scene. • Element: Scene to be added.
Load additive scene	Adds scenes from the list to the Editor scene.

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

3. WEAVR MANAGER

The Manager module collects and distributes content and basic functionalities to instructors and managers.

3.1 Architecture

The architecture of the WEAVR Manager is showed in the following schema, where each service performs different tasks:

- Identity microservice: enables the authentication and the authorization through a role-based approach, furthermore it contains the information of the users in a relational database
- Main services: manage all the functionality of procedures except file management, data are saved in a relational database
- File Provider: manage all the files from WEAVR Manager browser to the WEAVR Player, it has different implementations depending on the layer where files are saved (ex: local, blob…)
- Analytics microservice: allows the write of XAPI format feed coming from the WEAVR Player
- LMS integration microservice: enable a communication from/to the LMS. The LMS can trigger actions in the WEAVR Manager or can be called by the WEAVR Manager itself.
- Multiplayer service: enable collaborative scenario thanks to a Photon server that enable a data exchange through the different WEAVR Players
- PLM service: enable data resources exchange and process them to Augmented Reality WEAVR player

Identity, Main, File and Analytics microservices can communicate with each other thanks to an Event Bus manager that enables an asynchronous exchange of information.
In front of Main, File and Analytics microservices is placed an API Gateway that act as proxy in order to not interact directly with the services and enable a load balancing.

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com |website: www.txtgroup.com | Tel: +49 30 293620

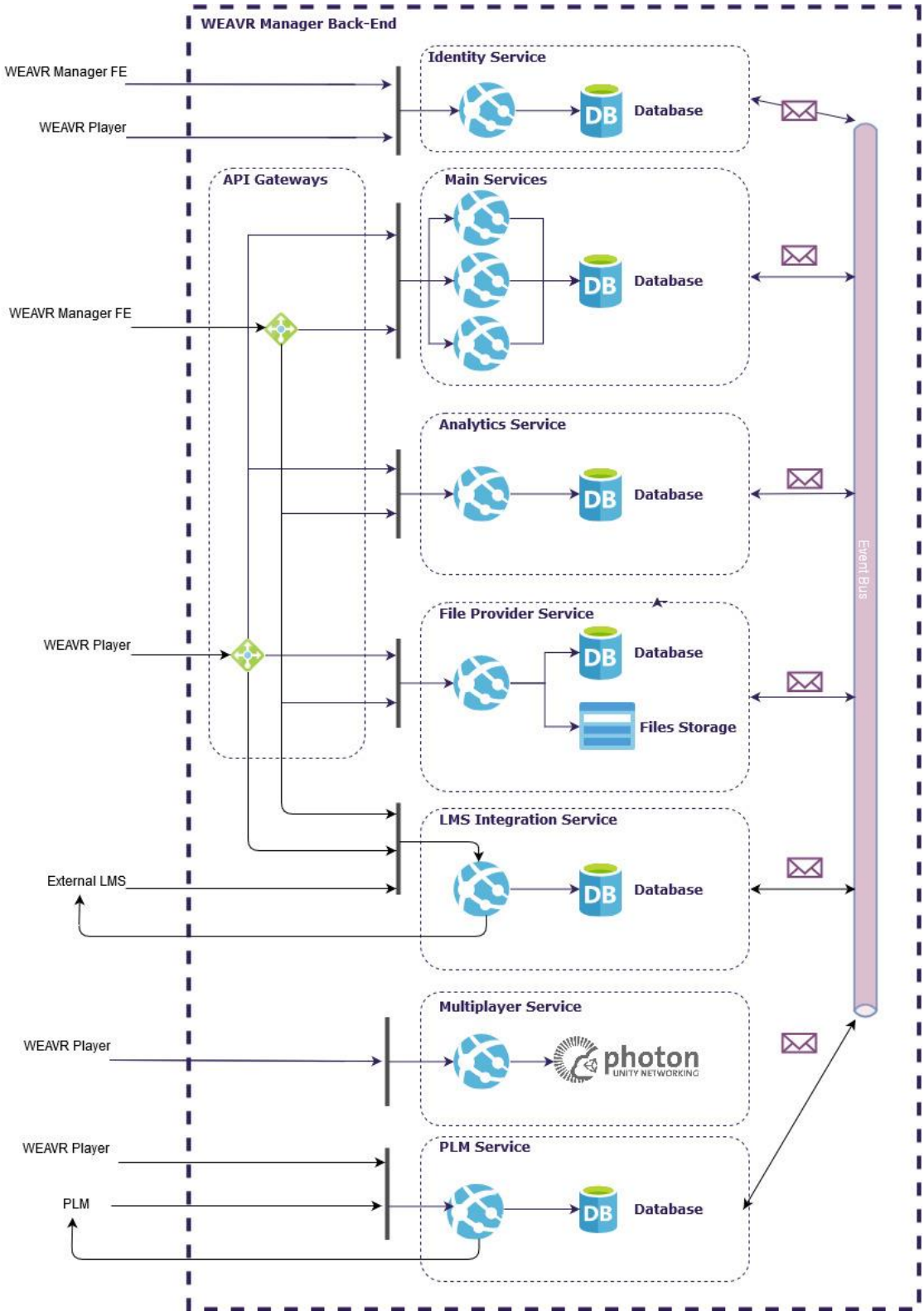


Figure 214.WEAVR Manager Architecture

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

3.2 Login

The login page (Figure 215.Login Page8) is showed in order to allow only authenticated user to access the system. Login also allow to understand the Role of the User that is log in into. Username/Email and Password are mandatory fields.

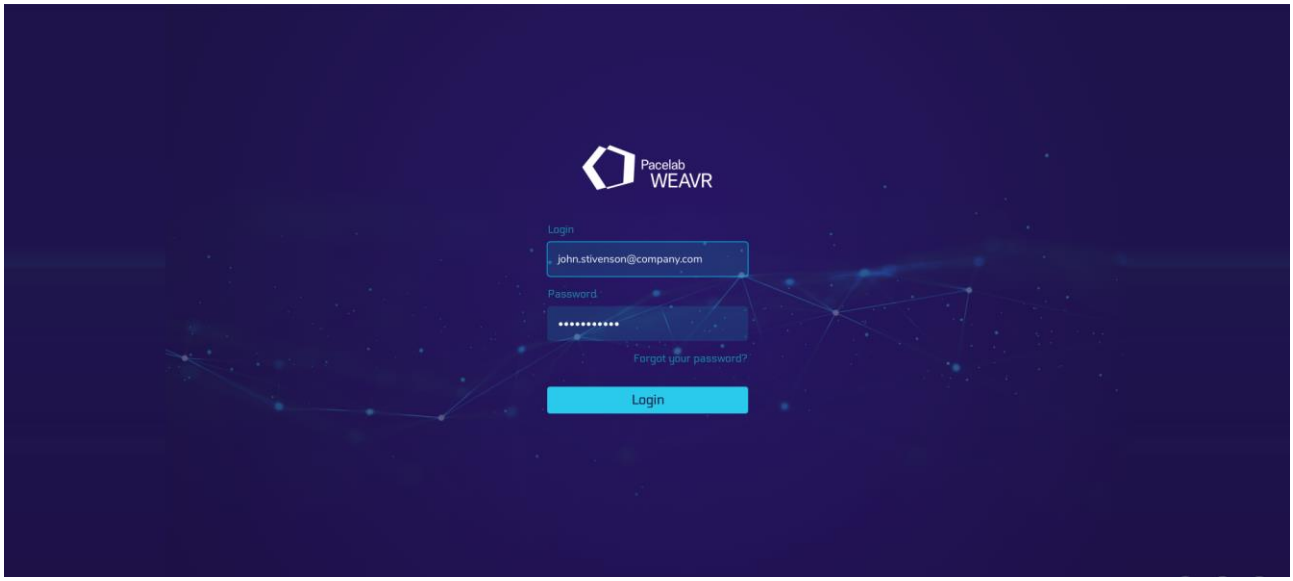


Figure 215.Login Page

After logging in the User List page (Figure 216.First Page) is showed to the actor that is using the platform.

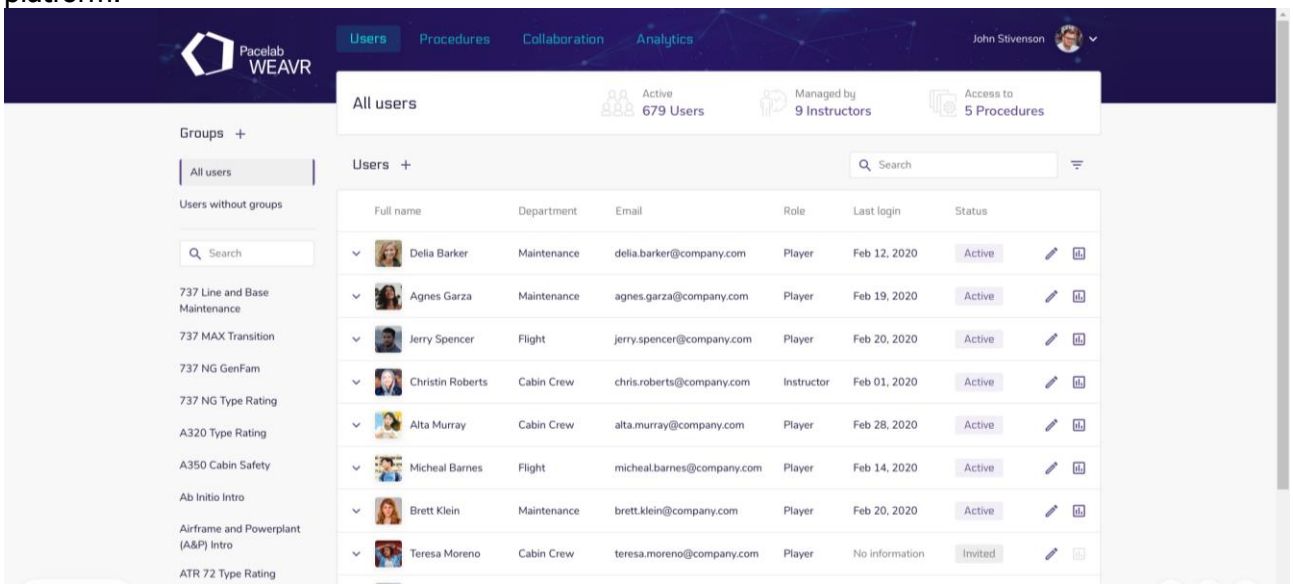


Figure 216.First Page

3.3 Header Information

All pages has a contextual header (Figure 217) that allow to understnd information about the page that user has open. Furthermore an image with user name is showed in order to be able to interact with user options (edit profile, logout).

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

Figure 217.Header

3.4 Content Management

The page of the dashboard relative to the management of the lessons allow to create, edit and delete the content information. It is also possible to upload a new version of an existing lesson, a new content for a specific version or a new lesson.

3.4.1 List of Procedures

The page contains the list of the lessons added into the platform, mains information about procedures are showed directly in the current page as well as the history of uploading. A lesson can be deleted, edited (adding it into group or associating it to a user).
The list is created with a paginator in order to switch easily through the pages, supported by a search functionality.

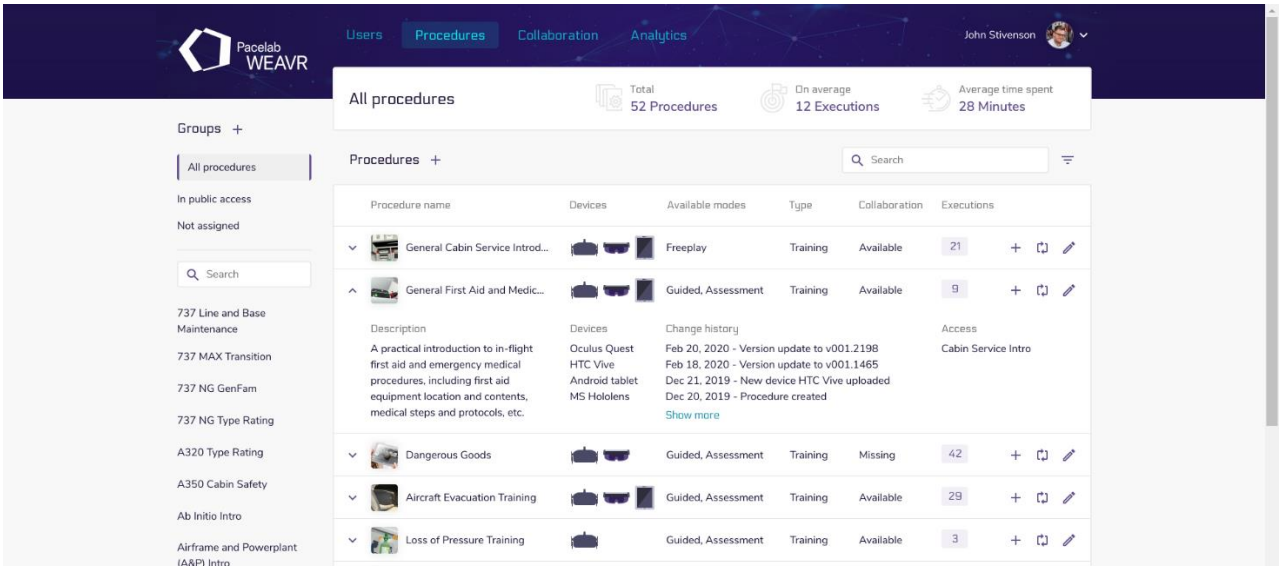


Figure 218. List of Procedures

3.4.2 Upload Procedure

To upload a new procedure, a new version of an existing procedure or a new platform for a specific version, the steps to follow are the same. First of all upload the "preview.json", the file of the procedure and the file of the scene that are autogenerated when you create a procedure inside the "AssetBundles" folder.

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

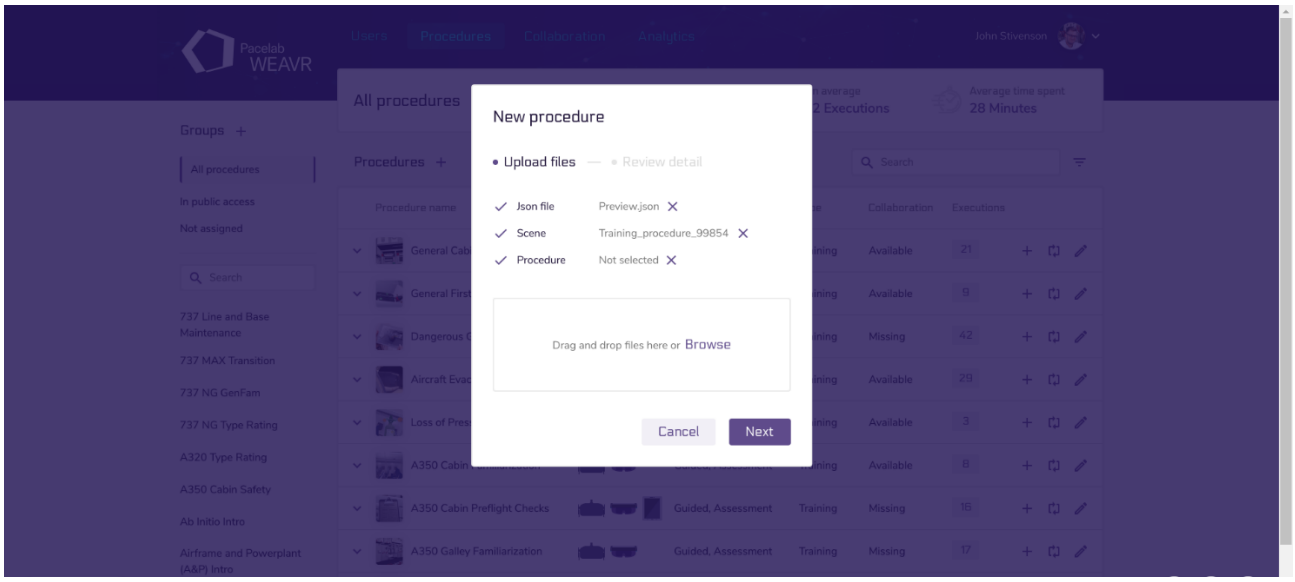


Figure 219.Upload procedure

After clicking on next will be showed the summary of the procedure that is uploaded. User can change mains information directly by the popup.

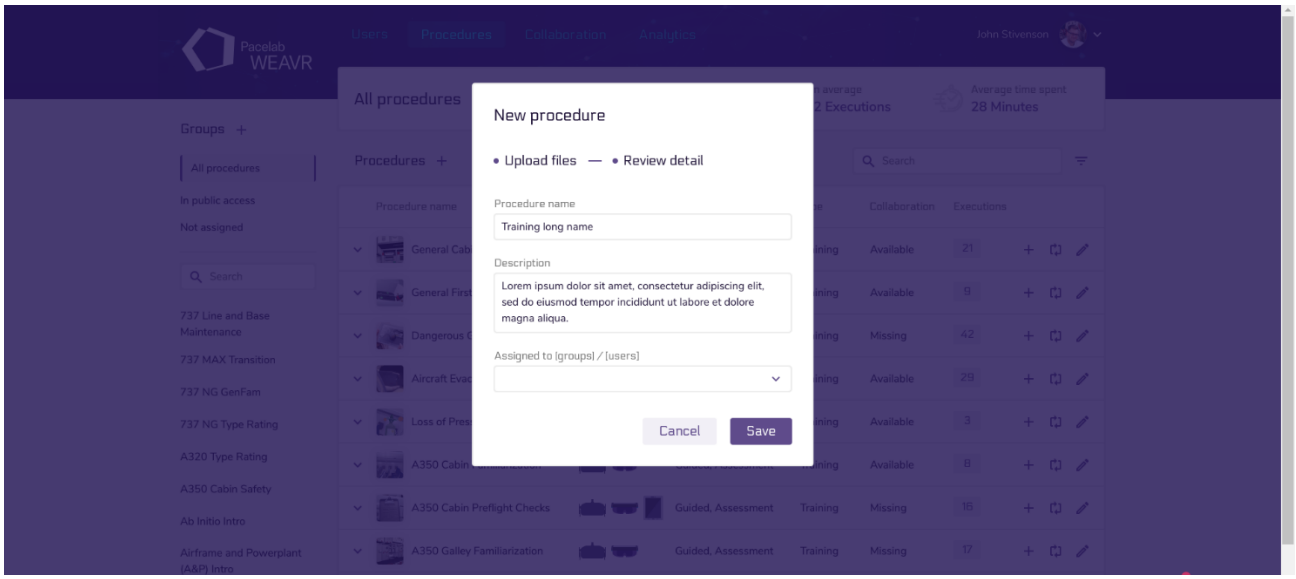


Figure 220.Save procedure

Clicking on "Save" the procedure files will be copied into files repository and the procedure will be available to be used.

3.5 Users Management

The page of the dashboard relative to the management of the users allow to invite, edit and delete the user information.

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

3.5.1 List of Users

The page contains the list of the users registered to the platform, mains information about users are showed directly in the current page. User can be deleted or edited (adding him into group or associating to him procedures).
The list is created with a paginator in order to switch easily through the pages, supported by a search functionality.

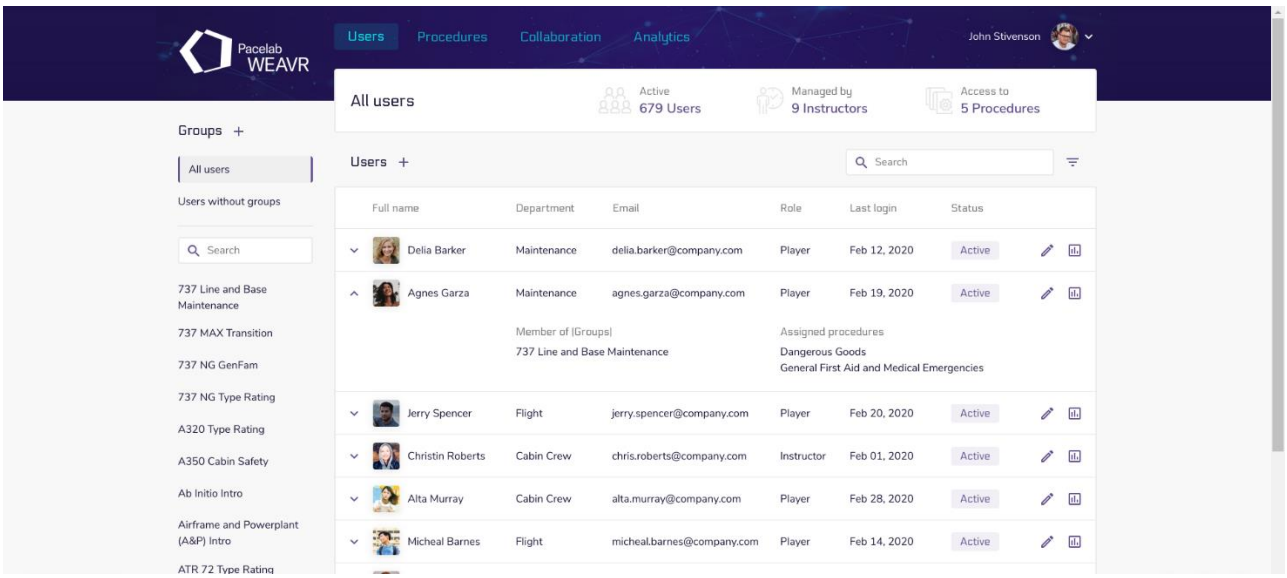


Figure 221.List of Users

3.5.2 Invite new User

Clicking on the "+" icon a new user can be invited to the platform, user will receive an email with link where user can enter first password and enters last missing information.

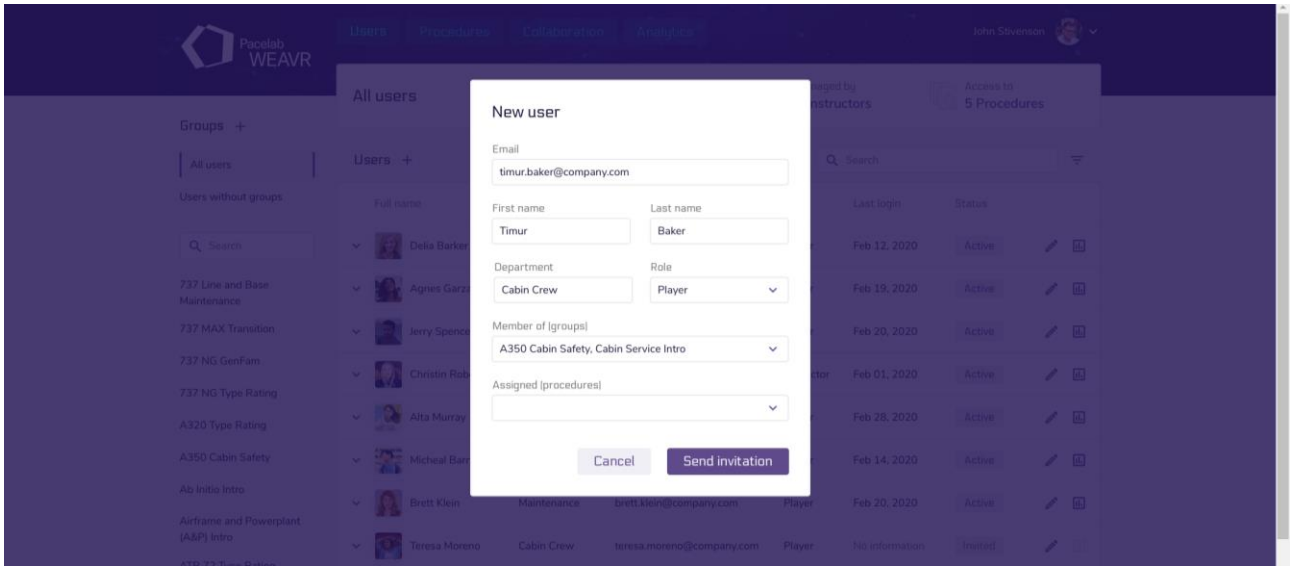


Figure 222.Create User

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

3.6 Group Management

The page of the dashboard relative to the management of the groups allow to add, edit and delete the groups information.

3.6.1 List of groups

All the pages have on the left the "Groups" list (Figure 223) in order to be able to filter the content showed on the main page. Furthermore, user can filter groups thanks to search textbox.

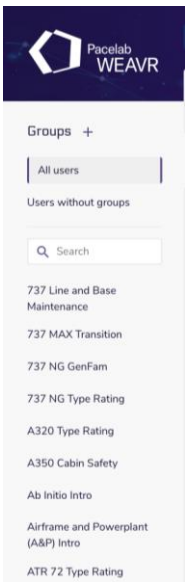


Figure 223.Groups List

3.6.2 Create new Group

Clicking on the "+" icon the user can create a new group with chosen name.

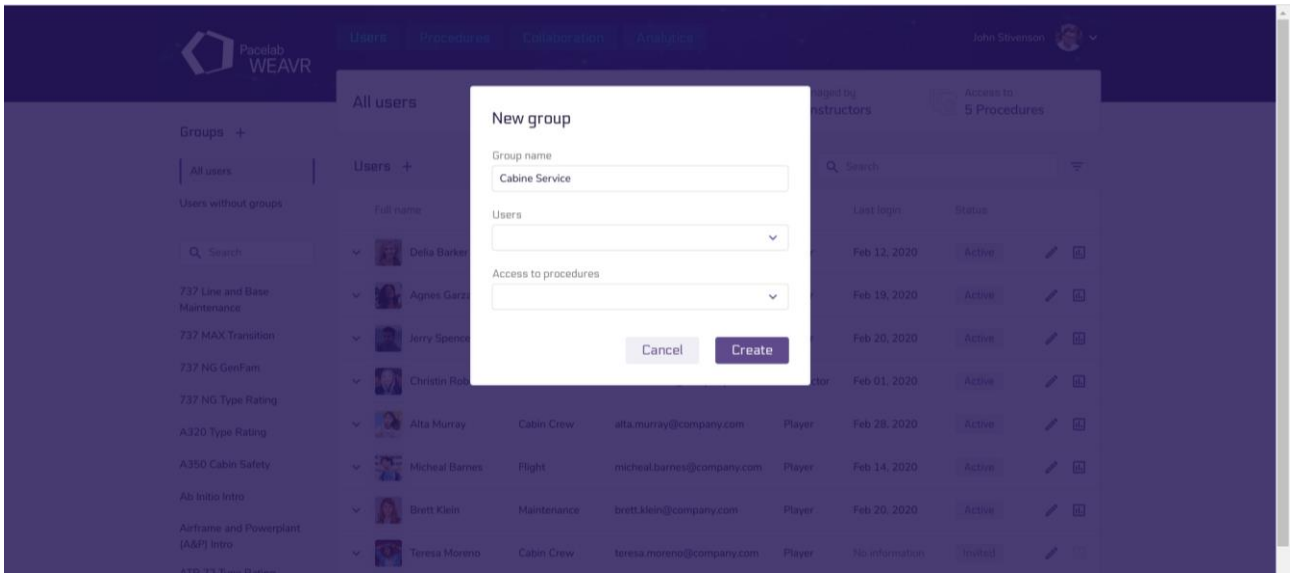


Figure 224.Group Create

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

3.7 Report

The page of the dashboard relative to the analytics enables instructors to understand the knowledge of students thanks to analytics feeds created directly form the executions of the WEAVR Player.

3.7.1 List of User Report

The page contains the analytics created by the executions of user using WEAVR Player filtered by date range or search textbox.

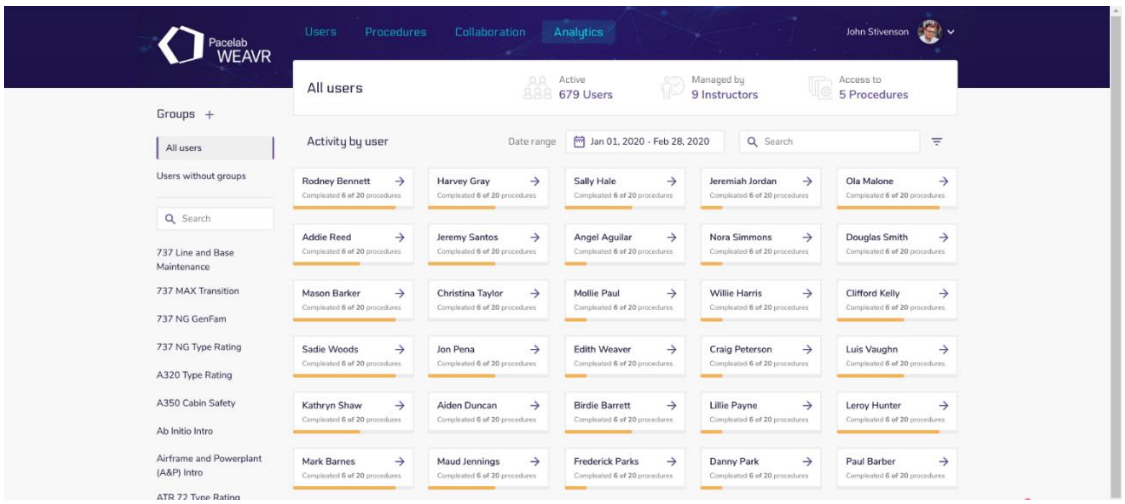


Figure 225.List of User Analytics

3.7.2 View User Report

Clicking on the user the information of him will be shown

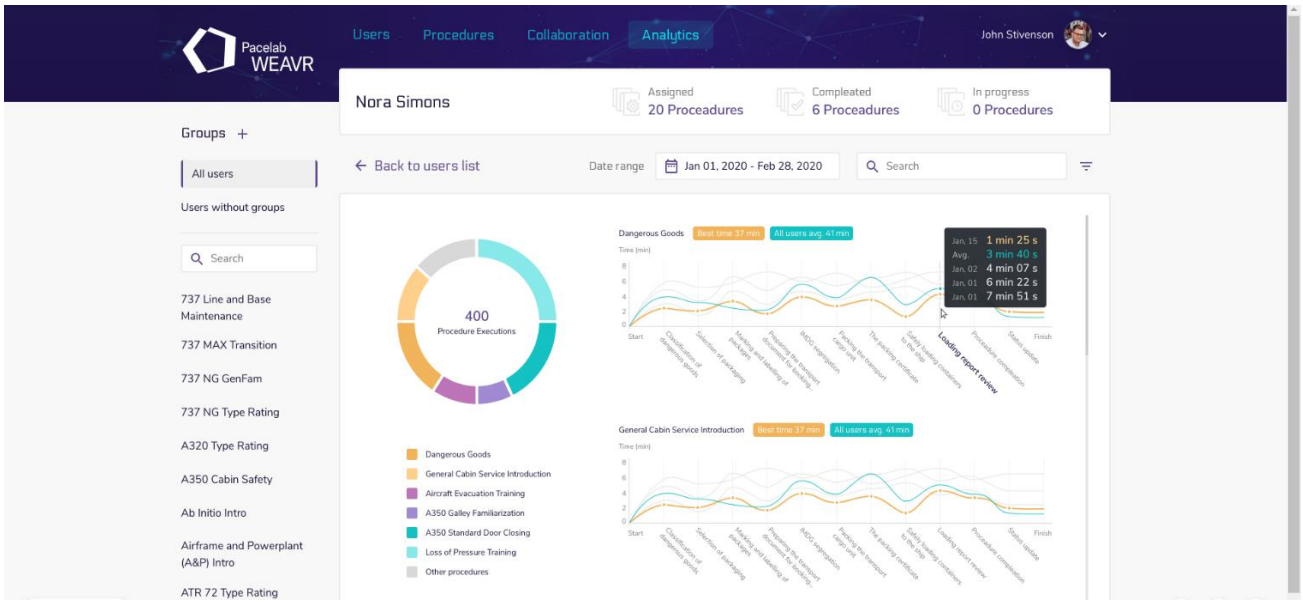


Figure 226.User Analytics Details

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

3.8 Collaborations (Multiplayer)

Collaboration enables students to create collaboration sessions directly from WEAVR Player. Other student can join session in order to cooperate and learn from colleagues.

The collaboration features are as follows:

- No need of collaboration room creation: the procedure itself can be shared by users
- User can join other user's session at any time during execution
- User can be invited by other users to join the procedure execution
- When user is disconnected, it is possible to re-join the session
- It is possible to set a minimum number of users for the procedure execution: the procedure will start only when the minimum number of connected users is reached.
- It is possible to define in WEAVR Creator which objects or which part of the procedure can be shared among users

3.9 Remote instruction and support

Instructors can monitor and help users during procedure execution directly from the WEAVR Manager.

- Instructor can monitor the procedure execution by displaying what the user is doing in a dedicated window
- Instructor can add indications to student during the procedure execution:
    - Pinpoint objects of interest
    - Add various indicators such as arrows, lines, circles, etc.
    - Use a laser pointer
    - Create billboards with custom text over objects of interest
- Instructor can join the procedure session in order to assist users during the execution

3.10 LMS Integration

LMS Integration enables the exchange of information between an external LMS about:

- Users information
- Users group
- Assigned lessons
- SCORM data
    - Time spent for each step
    - Time used to finish the lesson
    - Result of the exercise/lesson

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

3.10.1 LMS Data

Analytic data are all the feeds that WEAVR Player sends to the WEAVR Manager.
Normally, feeds are to be intended as user interactions, but that behaviour can be customized as a service in order to track different operations of the user and other not related to him.

The default analytic connected to procedure flow are:

- Start of Procedure
- End of Procedure
- Start of Step
- End of Step

The default analytic connected to user interactions are:

- Door component: Open and Close
- Key component: Lock and Unlock
- Connect component: Connect and Disconnect

3.10.2 WEAVR Player Connection

WEAVR Player communicates with WEAVR Manager thanks to Open API exposed by the WEAVR Manager. All the APIs are secured from unauthenticated access and can be used only after performing a login.

3.11 PLM Integration

PLM integration provide methodologies and interface that allow to manage, integrate and make available the data associated with the Mixed Reality Headset.
The PLM shall be configured with the features and processes necessary for the management of new information and the channels that allow to be used by shop floor applications.
It is possible to transfer from PLM the project data (3D models and/or instructions) and then be able to process them in Mixed Reality player.
The PLM shall be modified with a publication mechanism of the project data resources towards WEAVR Manager and/or Mixed Reality player.
Microsoft Hololens (or equivalent devices) viewers must be integrated into the company procedures and into company connectivity (Wi-Fi).

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

4. WEAVR PLAYER

4.1 Launch Content

The WEAVR Player can manage and execute any procedure it is allowed access to. The Player can be essentially classified in three types:

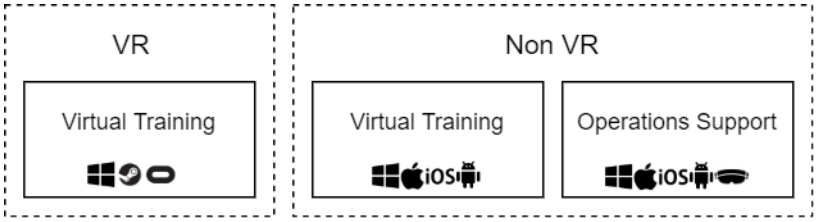


Figure 227. WEAVR Player Overview

- **VR Virtual Training (VR-VT)**
  - The VR Player is basically tailored to handle only Virtual Training procedures and only in Guide or Feedback modes, since there is no sound use case for Operations Support in VR. The navigation is performed either by physical movement of the user or by a teleportation mechanism, meanwhile the interactions are performed by virtually "touching" the interactive objects and pressing the relative buttons on the device controllers.

- **Non-VR Virtual Training (VT)**
  - The Non-VR Player can execute any Virtual Training procedure in any modes (Automatic, Guided and Feedback). It is loosely based on an FPS (First Person Shooter) mechanics to navigate the world and the interaction is handled by point and click mechanics for Guided and Feedback modes. The Automatic mode behaves more like an Operations Support procedure where the user cannot navigate nor interact with any object and is just presented to the procedure steps by means of an automatic camera movement and triggered animations.

- **Non-VR Operations Support (OpS)**
  - The Non-VR Player can also execute the Operations Support procedures. These procedures do not require any world navigation nor interaction with any virtual objects, thus the user is only presented with a few options for these procedures: the navigation throughout the procedure (previous and next steps), camera orbiting (rotating the camera around focused objects) and AR functionality (only on supported devices).

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

4.2 Standalone Content

The WEAVR player can also work as standalone player, without connection to the WEAVR Manager. In this case all the content is uploaded on the player application, as well as the rules for user authentication.

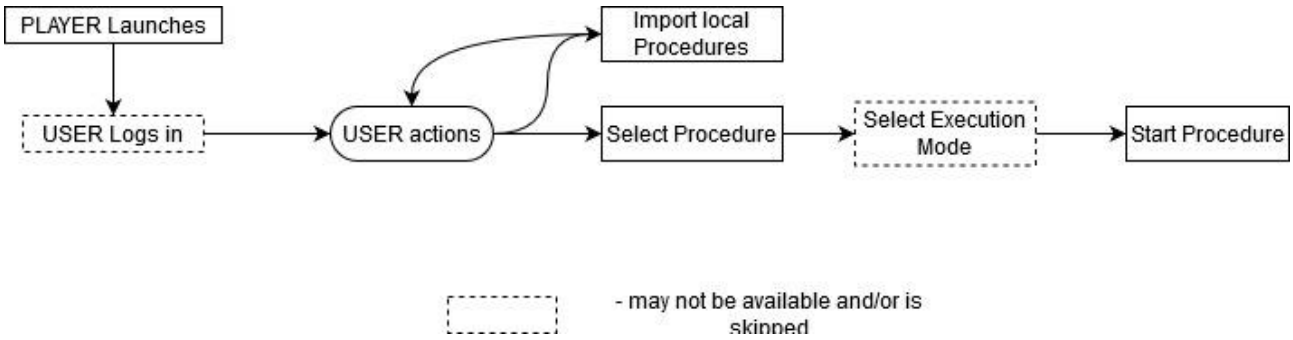


Figure 228.WEAVR Player standalone Workflow

1. Player launches and asks the user to log in (not always as the login is done automatically afterwards)
2. User enters credentials and logs in
3. User has a set of options to select from:
    a. Import local procedures (persisted in specific folder designated for the import)
    b. Select and execute a procedure
        i. Select the Execution Mode (if applicable)
            1. Start the procedure
            2. Share the procedure and start it (if the procedure can be shared)
        ii. Join a shared procedure (if there are any users who are sharing the procedure)

The same standard set of buttons as for the Manager synced Player is available.

Once a procedure is successfully finished, the user will be prompted with a dialog box asking whether the procedure should be restarted or not.

4.2.1 Build Standalone application

You can build a standalone application of the WEAVR Player. To attach the created procedure to the WEAVR Player and build the application, you must:

1. Open the WEAVR Player scene in:
   Assets/WEAVR/Player/Assets/Scenes

2. In Hierarchy find the Builtin Procedures component (Figure 229) in:
   PLAYER → Model → BuiltInProcedures

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com |website: www.txtgroup.com | Tel: +49 30 293620

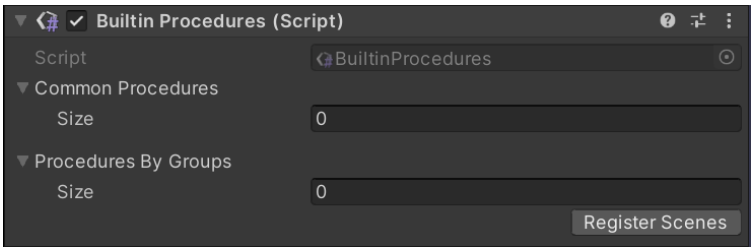


Figure 229. Builtin Procedures

3. Add the procedures required in "Common Procedures" list. Figure 230 shows an example of attached procedure.

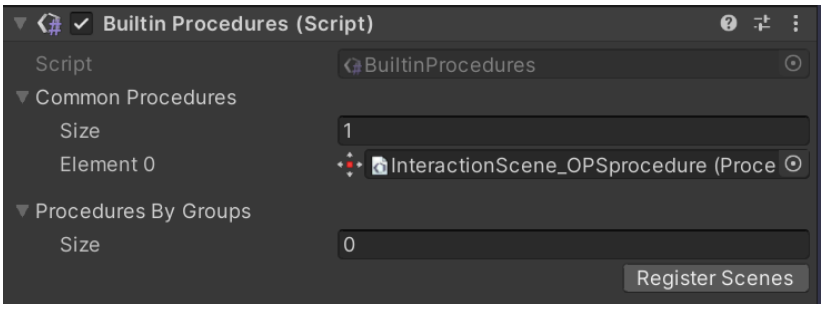


Figure 230. Example of attached procedure in Builtin Procedures

4. Click "Register Scenes" button to load the involved scenes in the "Scenes In Build" list of the Scenes Settings.

5. Open the settings window:
Toolbar menu → File → Build Settings…

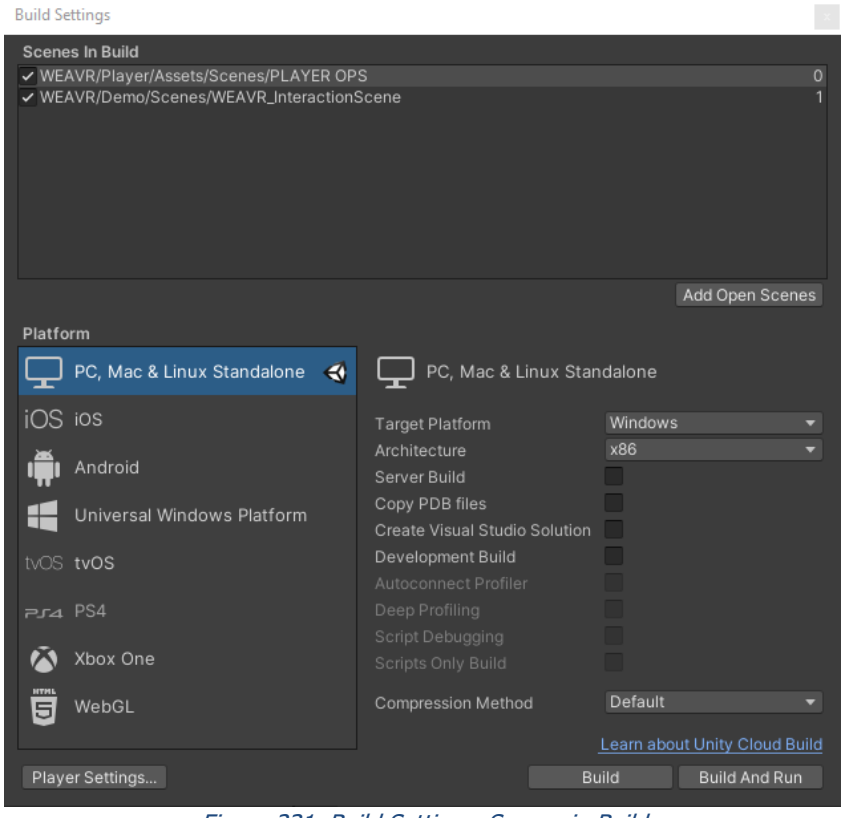


Figure 231. Build Settings, Scenes in Build

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

6. Select the required Platform. The platform selected is indicated with the Unity symbol. To change the platform select the required one and click on "Switch Platform" button, making sure that the platform tools are installed.

Start the build by clicking on the "Build" button

4.3 WEAVR Manager Connection

In case the WEAVR player is connected to the WEAVR Manager, the typical workflow is as follows:

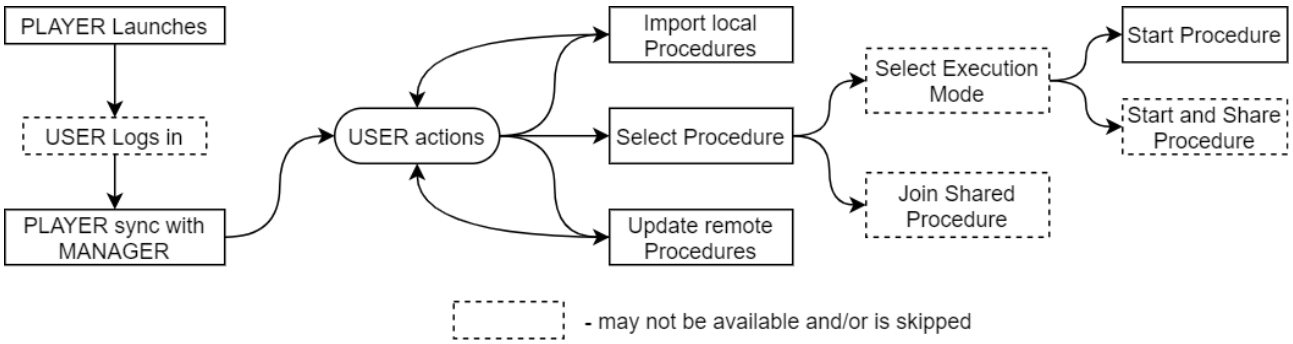


Figure 232.WEAVR Player Manager Connected Workflow

1. Player launches and asks the user to log in (not always as the login is done automatically afterwards)
2. User enters credentials and logs in
3. Player synchronizes the list of procedures with the WEAVR Manager
4. User has a set of options to select from:
    a. Import local procedures (persisted in specific folder designated for the import)
    b. Update remote procedures (update or download new available procedures)
    c. Select and execute a procedure
        i. Select the Execution Mode (if applicable)
            1. Start the procedure
            2. Share the procedure and start it (if the procedure can be shared)
        ii. Join a shared procedure (if there are any users who are sharing the procedure)

Throughout the procedure execution some buttons will be available, those are usually procedure agnostic. The standard set of buttons are:

- *Change Execution Mode* (for VT only): allows the user to change how the procedure is executed during its execution.
- *Replay Step* (for VT only): in case the user is blocked and cannot proceed, this button will repeat all the helping actions of the step (Text-To-Speech, Billboards showing, etc.) even when the execution does not provide this help (e.g. Feedback execution mode).
- *Restart Procedure*: restarts the currently running procedure.
- *Change Language*: changes the procedure execution language during the execution itself. This, depending on the running step, may prompt a step replay.

Once a procedure is successfully finished, the user will be prompted with a dialog box asking whether the procedure should be restarted or not.

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

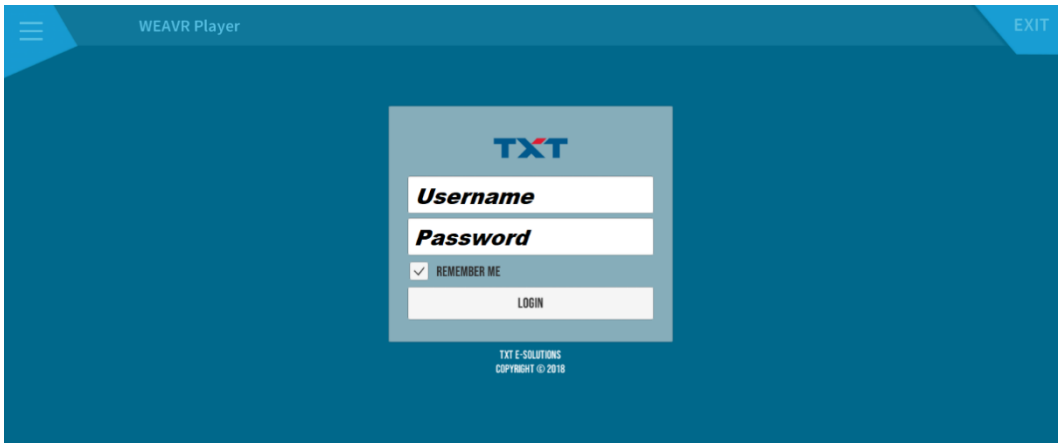
4.3.1 Content download from WEAVR Manager

As explained in the above flowchart, due to the synchronization with the WEAVR Manager, WEAVR Player can download content from the Manager, allowing user to play specific content assigned by the instructor.
During the phase of synchronization, a list of available content is exchanged between Player and Manager. From this list, the user can either remove, download or update procedures. When downloading or updating a procedure, the Manager sends to the Player a package containing both the procedure and the scene (environment the procedure is played in), thus allowing a completely empty Player to evolve steadily with new content.
The downloaded content is usually persisted until the user instructs the Player to remove it. Every piece of content is categorized and synchronized based on user's group and whether the user was authorized to access that content.

4.4 Immersive Virtual Reality Player

To start to use the VR player you need to log in. The login page automatically appears on the pc display and on the headset display appears a black view with text "LOGIN REQUIRED REMOVE YOUR HEADSET".

Figure 233. Headset black view



PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

Figure 234. Login page VR Player

After a standard login on your pc the headset shows you the Player Room, a white environment where you can launch your procedure with the help of your VR controller. Your VR controller usually has the buttons configuration described in Figure 235

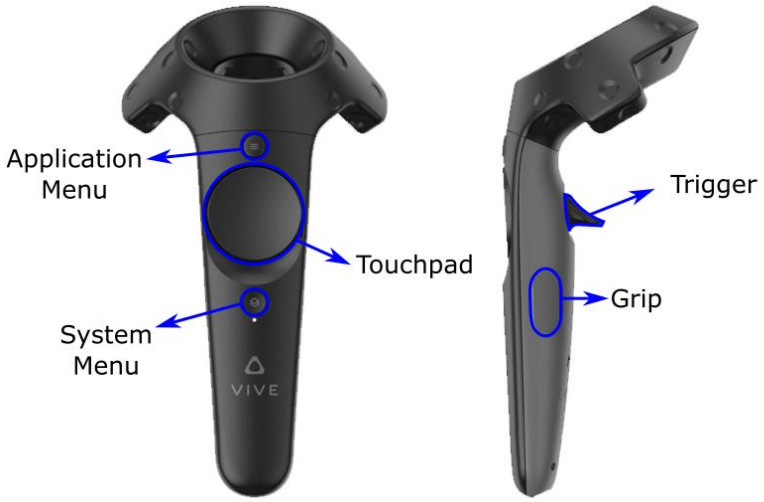


Figure 235. Controller buttons

With the Application Menu button, you can open the controller menu, a window on top of your controller where you can interact with the other controller that becomes a hand usable as a pointer. With the index finger you can touch the buttons on the controller menu and you can click them with pressing the trigger button. The menu on the controller is shown in the Figure 236

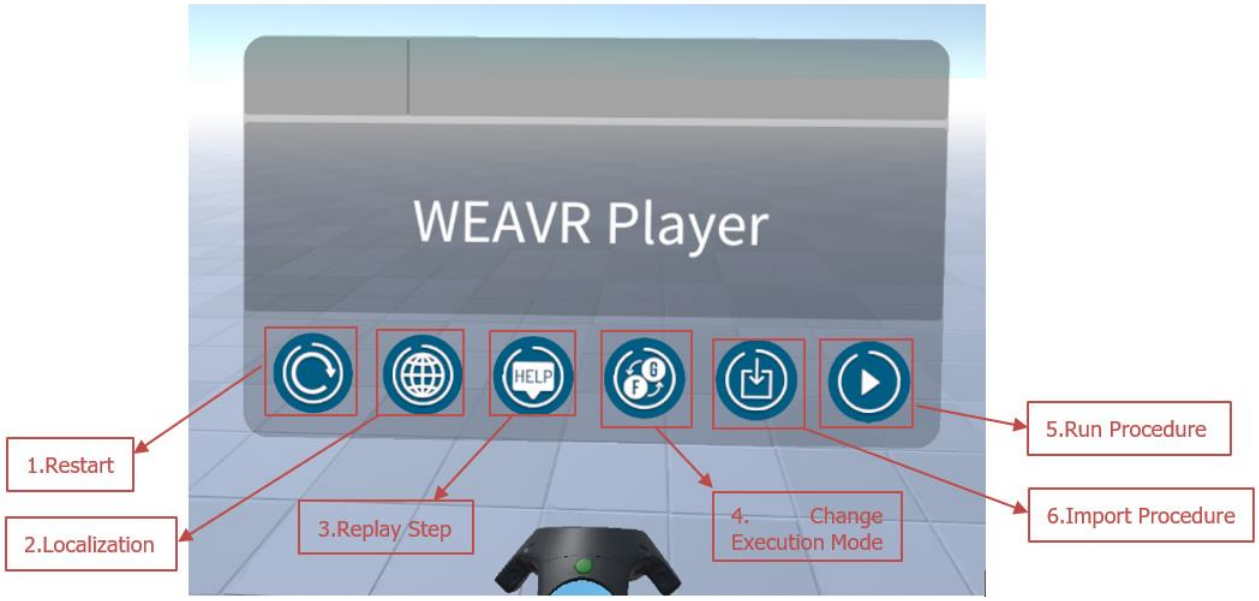


Figure 236. WEAVR Player controller menu

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

1. Restart: the procedure that is currently running will restart from the beginning. This button is enabled when a procedure is running

2. Localization: from the Localization popup (Figure 237) the user can choose the procedure language from the ones available (the procedure languages are set in Create Procedure Wizard, Section 2.13.1). This button is enabled when a procedure is running

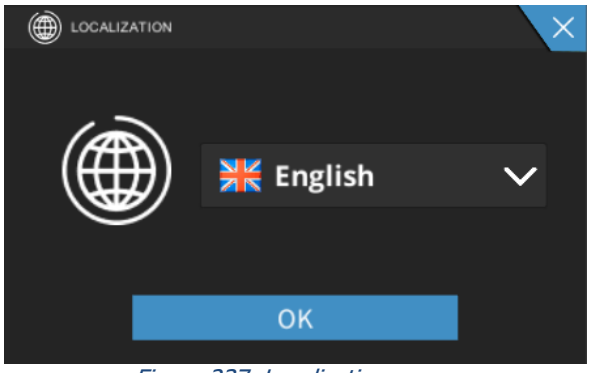


Figure 237. Localization popup

3. Replay Step: shows the hint actions (billboard, outline, wait, TTS) of the current step only

4. Change Execution Mode: switch from feedback to guided the execution mode of the procedure, and viceversa

5. Import Procedure: all the procedures in the Persistent Data Path (https://docs.unity3d.com/ScriptReference/Application-persistentDataPath.html) are imported in the player and ready to be run (see Section 2.13.10)

6. Run Procedure: when you click on this button a carousel with the available procedures is shown. The carousel shows a set of procedures like the one shown in Figure 238

Figure 238. Procedures carousel tile

The header of the tile shows the name of the procedure and a button with an icon that explains if the procedure is available on the device or needs to be downloaded. With a click on the icon with the cloud image you can download the procedure. The footer shows a

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

variable number of buttons which represent the available mode to start the procedure. With a click on one of these buttons, the carousel disappears and the procedure starts to run in the selected mode.
When a procedure is running, the controller menu shows the same buttons of the Figure 236 but it adds information about step title, description and number as shown in Figure 239.

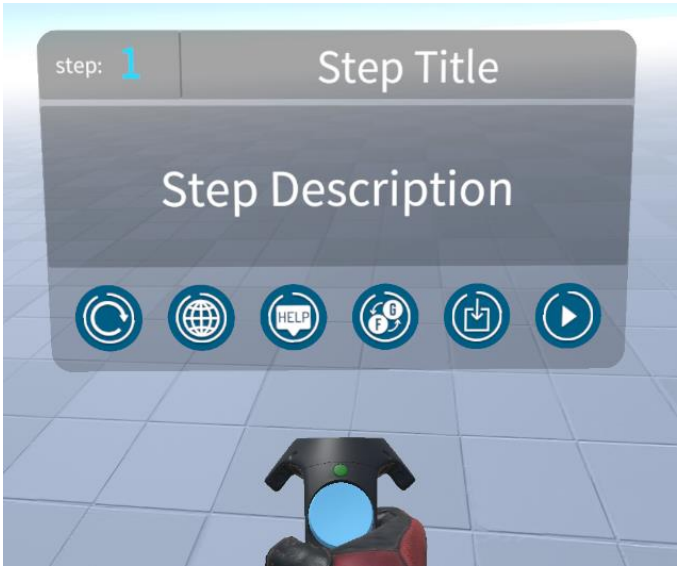


Figure 239. Step controller menu during procedure running

4.5 PC/laptop and mobile/tablet Player

The non-VR content can be played on both pc/laptop and mobile/tablet devices, by using the relevant version of the WEAVR player.

To start using the Player you need to log in. The login page automatically appears on the display.

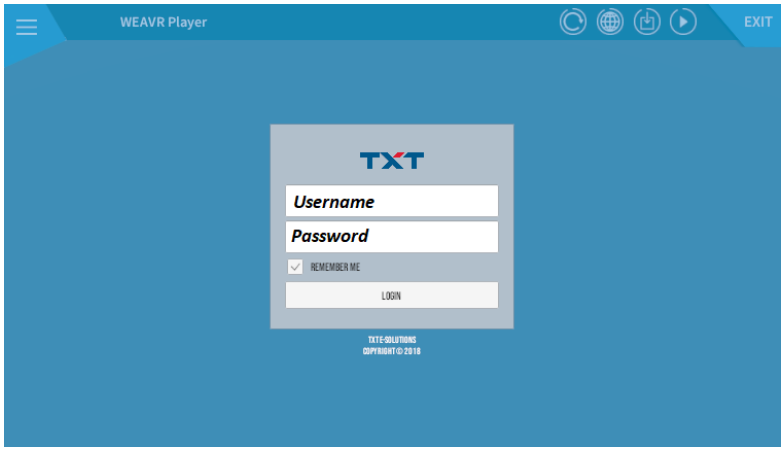


Figure 240. Login page Player

After the log in, the Player main page appears. From the main page the user is able to import, set the language, run and restart the procedure.

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com |website: www.txtgroup.com | Tel: +49 30 293620

4.5.1 Player main page

Figure 241. Player main page

1. Menu: menu popup appears; it contains the username and logout button

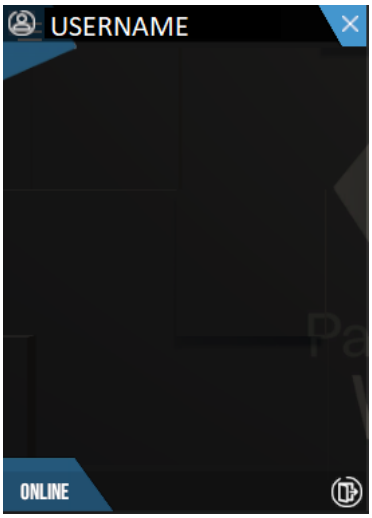


Figure 242. Player Menu

2. Exit: quit the application

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

4.5.2 Player main buttons

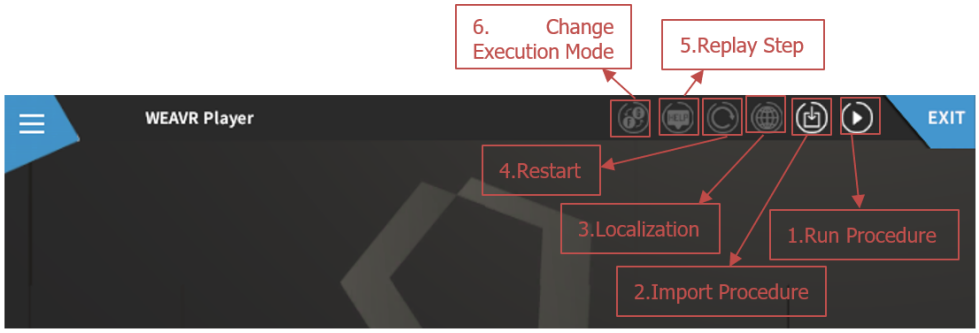


Figure 243. Player main buttons

1. Run Procedure: from the Run Procedure popup the user is able to choose which procedure to run, from the ones imported in the Player, and the execution mode

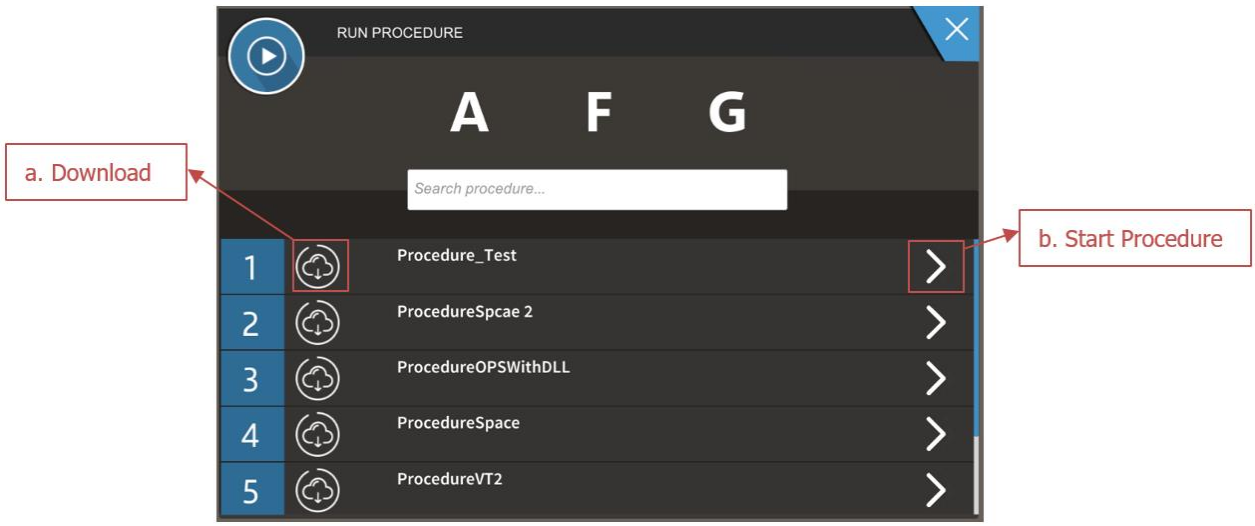


Figure 244. Run Procedure popup

a. Download: in case of WEAVR Manager connection, download the procedure before running it. The procedures downloaded are stored, the download is required only the first time.

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

b. Start Procedure: the procedure can be started by pressing on the procedure title in the list. A popup will appear with the procedure name, procedure description and available operation modes (

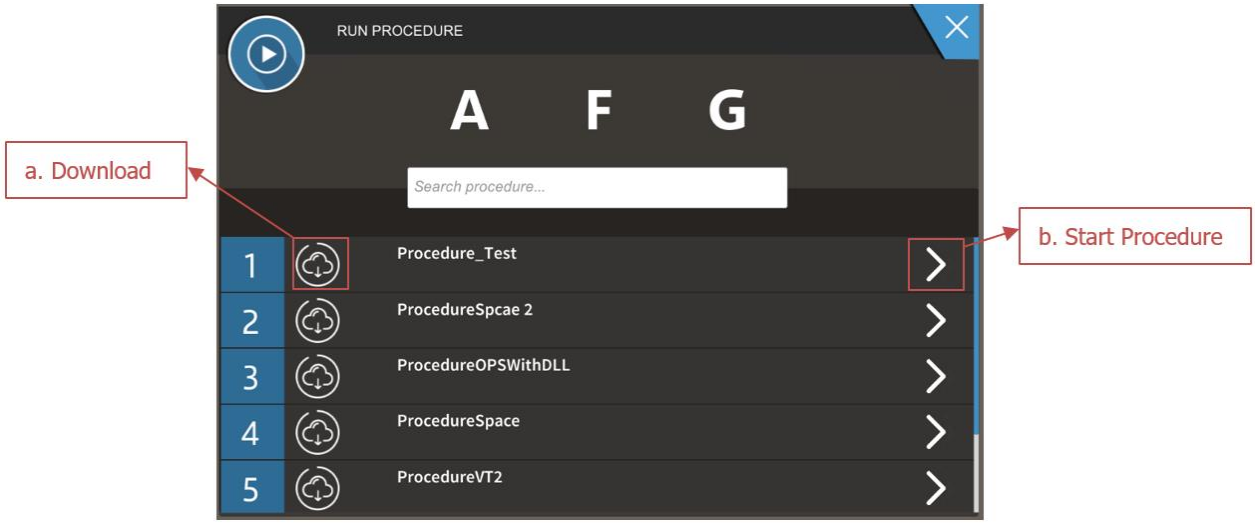


Figure 244). With a click on one of the procedure modes buttons you can start the procedure and the scene is automatically opened.

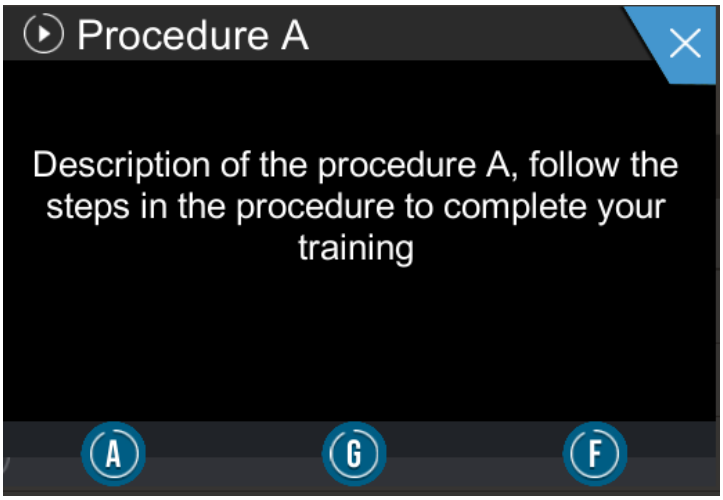


Figure 245. Popup to run the procedure

2. Import Procedure: all the procedures in the Persistent Data Path (https://docs.unity3d.com/ScriptReference/Application-persistentDataPath.html) are imported in the player and ready to be run
3. Localization: from the Localization popup (Figure 246) the user can choose the procedure language from the ones available (the procedure languages are set in Create Procedure Wizard, Section 2.13.1). This button is enabled when a procedure is running

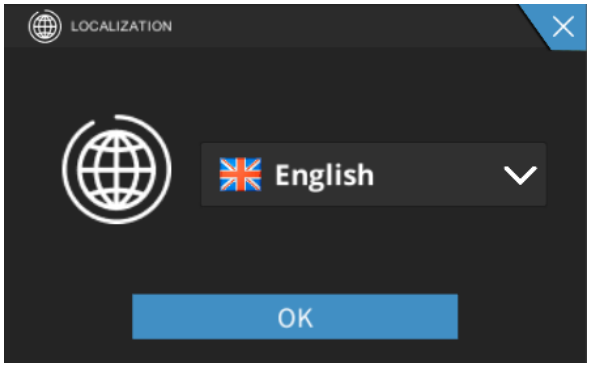


Figure 246. Localization Popup

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

4. Restart: the procedure that is currently running will restart from the beginning. This button is enabled when a procedure is running

5. Replay Step: shows the hint actions (billboard, outline, wait, TTS) of the current step only

6. Change Execution Mode: switch from feedback to guided the execution mode of the procedure, and viceversa

4.5.3 Content fruition UI

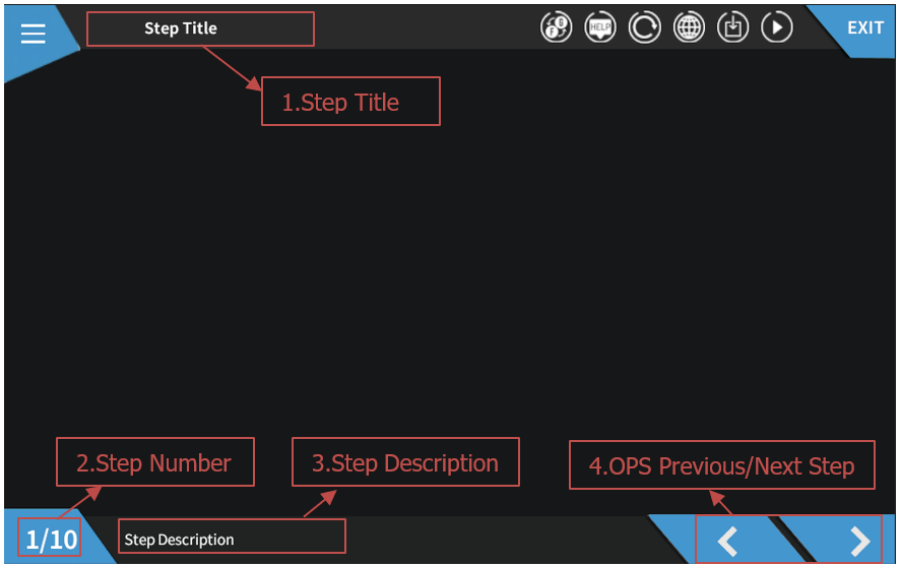


Figure 247. Player Procedure UI, common canvas

1. Step Title: title of the current step as indicated in the Step Inspector

2. Step Number: number of the current step as indicated in the Step Inspector

3. Step Description: Description of the current step as indicated in the Step Inspector

4. OPS Previous/Next Step: available only for OPS (Operation Support) procedures, these buttons are used to change the step forward and backward

4.6 Augmented Reality Player

The WEAVR player can be ran on Augmented Reality devices (helmet, tablet). In this case the OPS (Operation Support) content is fruited in Augmented Reality mode.

The player functionalities and UI are the same described in the chapter above with some additional functionality. Augmented devices can be essentially divided in 2 categories:

- See-thru devices (e.g. Microsoft's Hololens)
- AR camera devices (e.g. Supported Tablets and Smartphones)

Consequently, the Player have a slightly different usage for these two categories.

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

4.6.1 WEAVR AR for See-through devices

The Player has a panel which is placeable into the real world and some 3D models required by the procedure. The panel shows all the relevant information for the procedure status (e.g. current step, current step description, etc.) as well as some control buttons (e.g. Next Step, Previous Step, etc.). The 3D models can be moved, scaled and/or rotated at will and are used either overlaid over real objects (e.g. a section of an engine replacing that same physically missing part of the engine, thus having a physical engine with a virtual part attached to it) or as pure AR objects (e.g. the whole engine on a workbench). These models are then manipulated and/or animated depending on the procedure.

4.6.2 WEAVR AR for Camera Devices

It is the standard Player with some additional buttons:

- Camera Orbit: this button is more like a toggle locking on and off the camera to orbit around a specified object during procedure editing. The user can rotate the camera to see the object of interest from different angles.
- AR: this button, when available, enables the AR functionality and allows the user to place the object of interest over the physical world. This functionality varies depending on the context, object and device used. It can be static, which overlays the object (although semi-transparent) over what the device's camera sees. It can also be dynamic where the device scans the physical environment for a good placement for the object, and once placed, the object can be interacted with and the user is free to move around the object.

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com |website: www.txtgroup.com | Tel: +49 30 293620

5. ADVANCED FEATURES

5.1 Build Standalone Application

When the project (and the procedure) is ready, the application can be created. To build the application without using the WEAVR player, a few passages should be followed:

1. Open the settings window: *Toolbar menu → File → Build Settings…*
2. Check if there is the project scene in the "Scenes in Build" list. If not click on "Add Open Scenes"
3. Select the required Platform. The platform selected is indicated with the Unity symbol. To change the platform select the required one and click on "Switch Platform" button, making sure that the platform tools are installed.
4. Start the build by clicking on the "Build" button

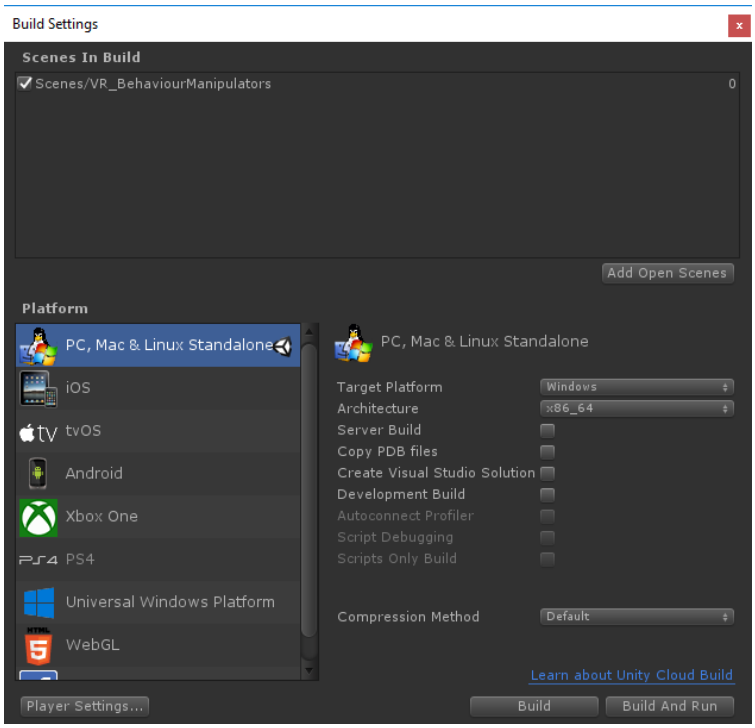


Figure 248. Build Settings

For more information, refer to the Unity manual: https://docs.unity3d.com/Manual/PublishingBuilds.html.

5.1.1.1 *Attach procedure*

To attach the procedure to the Build there are two possibilities:

a) The procedure starts by default when launching the application
   - Hierarchy window → WEAVR → Procedure Runner (Figure 249)→ Enable: Start When Ready and insert the correct procedure in Current Procedure

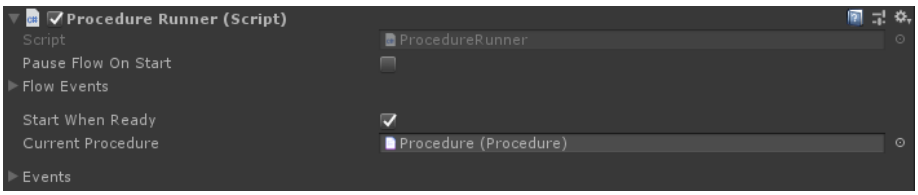


Figure 249. Procedure Runner

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

b) The procedure starts by event
   - Use the Procedure Runner component as event. Figure 250 shows an example of event that launch the current procedure inserted in the Procedure Runner component.

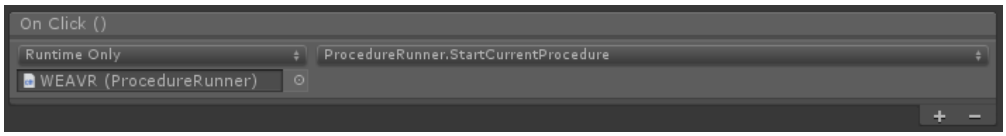


Figure 250. Example of Procedure Runner by event

5.1.1.1.1 Procedure Launcher

This component allows to launch a procedure belonging to a different scene. The new scene and procedure will start with the chosen execution mode.

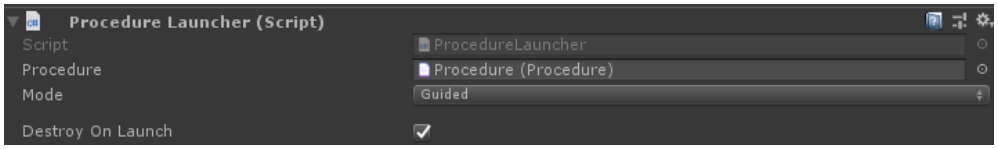


Figure 251. Procedure Launcher

- Procedure: procedure to launch
- Mode: execution mode required for the procedure
- Destroy On Launch: whether to destroy the GameObejct owner once the scene and procedure have been launched.

NOTE
To work, the scenes owner of the procedures launched need to be added to the Build Settings "scenes in build" list.

WARNING
Make sure this component is added to an empty GameObject, to avoid errors in case it is destroyed.

5.1.1.2 *Setup for multiple scenes for Standalone Application*

To load more scenes additively, a GameObejct in the main scene needs the Scene Loader and the Toggle On Scene Load components.

5.1.1.2.1 Scene Loader

This component should be included in all the scenes to load additively to the main one; it is called by event from Toggle On Scene Load (Section 5.1.1.2.2) contained in the main scene.

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

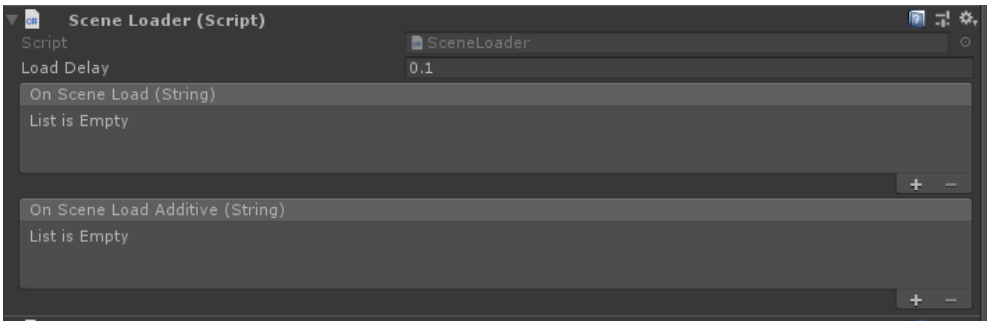


Figure 252. Scene Loader

- Load delay: delay time before loading the additive scene

5.1.1.2.2 Toggle On Scene Load

This component is included in the main scene (as separated GameObject) to call by event the Scene Loader component of all the scenes that should be added.

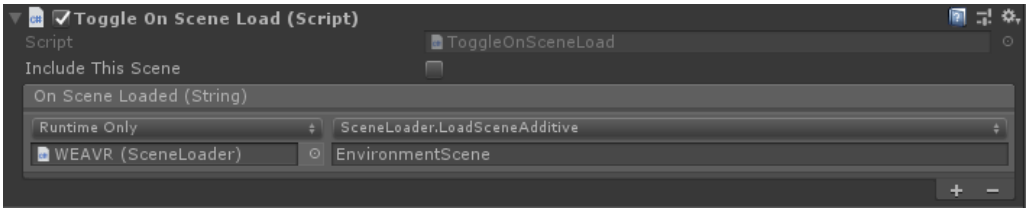


Figure 253. Toggle On Scene Load

- Include This Scene: whether to trigger the event when this GameObject scene is loaded

Figure 253 shows how to call the Scene Loader component (Section 5.1.1.2.1) of the scenes to add to the main one.

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

5.2 WEAVR Simulation Hub

WEAVR Simulation Hub (SimHub) is a module designed to agnostically establish connections between various simulation endpoints, keep the information exchange between them and allow shared memory data exchange where needed. The SimHub is NOT a Plug&Play system and should be tailored to every simulation endpoint accordingly. The main components of the SimHub are described as follows:

- **SimHub Server**: establishes connections between SimHub Clients using a Publish/Subscribe paradigm.
- **SimHub Client**: exchanges data with other SimHub Clients using the message protocol provided by the SimHub framework. Every client needs to provide a data representation file (normally an Interface Control Document **ICD**) to the server in order to receive a correct connection with other clients.
- **Shared Memory**: handlers for shared memory management. Data can be exchanged locally (on the same host) between SimHub Client and customer's simulation.

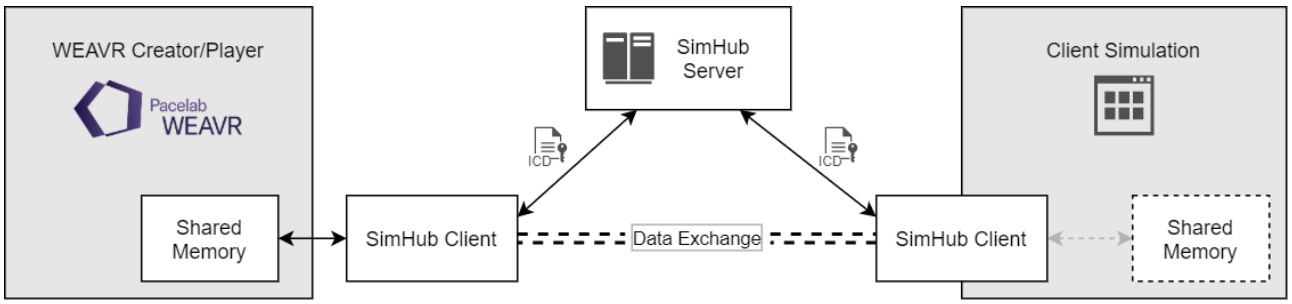


Figure 254. WEAVR Simulation Hub Overview

The WEAVR Creator uses the ICD file to define the variable bindings and those can be used throughout the procedures and/or by interactions. E.g. a cockpit switch with three states can bind to a variable and depending on the binding type (read, write or read/write) can either update its state based on variable value (read), update the variable value based on its state (write) or both (first read then write). This way a virtual cockpit can be created with all its components and then later have those components bind to simulation variables. The procedure can as well interact with variables by either directly reading their values or writing them, thus allowing a control flows driven by the simulation variables.

The WEAVR Player instead doesn't require the ICD file but it uses the Shared Memory to exchange data between SimHub Client and its bindings and/or running procedure. It is therefore mandatory that the ICD file used by the SimHub Client remain structurally the same as the one used in the Creator. The Player uses custom-built optimizations, such as various granularity access options, to optimize reading/writing data in shared memory.

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com | website: www.txtgroup.com | Tel: +49 30 293620

5.2.1 SimHub Server

The SimHub Server keeps track of connected SimHub Clients and handles any new connection and/or disconnection from clients. Upon a connection from a new client, the server receives the data file (ICD) from the client and based on received file instructs the client to which other clients it should connect to have all the requested data and informs other clients of the newly connected one. Virtually the clients exchange data between them using the provided message protocol.
The SimHub Server is a standalone application and can be deployed on any host, even on one where other SimHub Clients reside. The server uses a TCP connection for the clients' management and optionally an UDP connection for data exchange.

5.2.2 SimHub Client

The SimHub Client manages the data exchange between other clients, but only those who can provide the required data. It also can access the shared memory to read/write data used by the host simulation, which is also the preferred data exchange method between the SimHub Client and the simulation.
The SimHub Client comes in two flavors:

- As a standalone application ready to be used As-Is. The host simulation is however responsible for providing data to the client by using the shared memory as well as retrieving the data provided by the client (read: shared memory synchronization).
- As an SDK to be integrated into the simulation application. This allows the host simulation to take full control of the data exchange and skip entirely the shared memory synchronization thus removing entirely the sync overhead. The SimHub Client will handle connection and data exchange with other clients.

5.2.3 Shared Memory

The suite of WEAVR Simulation Hub contains also a dedicated library for shared memory access (although fully functional only on Windows Platforms). This library leverages shared memory use thus freeing the developer from writing boilerplate code for shared memory access. It also allows other non-C++ applications to access the shared memory. E.g. the WEAVR itself (which is written in C#) uses this library to access the shared memory.

PACE Aerospace Engineering and Information Technology GmbH Am Bahnhof Westend 13 | D-14059 Berlin | Germany mail: paceinfo@txtgroup.com |website: www.txtgroup.com | Tel: +49 30 293620