VIA-VR Editor

User Manual

Games Engineering Research Group, University of Würzburg

last updated: 2024-04-08

Contents

1 Introduction **5**
1.1 About . . . 5
1.2 Getting Started . . . 5
1.2.1 Installation . . . 7
1.3 How to Start . . . 13
1.3.1 Initial Configuration . . . 13
1.3.2 Welcome Screen . . . 13

2 Optimize Tab **14**
2.1 Introduction . . . 14
2.2 Optimization Process . . . 14
2.2.1 File Preparation . . . 14
2.2.2 Running the Preprocessor . . . 14
2.2.3 Optimization Feedback . . . 14
2.2.4 Previewing Optimized Files . . . 15
2.3 Saving Optimized Files . . . 15
2.4 Troubleshooting . . . 15

3 Objects Tab **16**
3.1 Overview . . . 16
3.2 Accessing the Spoke Editor . . . 16
3.3 User Interface Components . . . 16
3.4 Spawn Point . . . 16
3.5 Detailed Feature Usage . . . 17
3.5.1 Adding and Modifying Objects . . . 17
3.5.2 Object Selection . . . 17
3.5.3 Camera Movement . . . 17
3.5.4 Transform Gizmo . . . 17
3.5.5 Transformation Tools: Precision Manipulation of Virtual Assets . . . 17
3.5.6 Grid Tools: Enhancing Alignment and Placement Accuracy . . . 18
3.5.7 Lighting . . . 18
3.6 Saving Projects . . . 19
3.7 Conclusion . . . 19

4 Behavior Tab **20**
4.1 Meta Data Editor . . . 20
4.1.1 Introduction . . . 20

4.1.2 Requirements . . . 20
4.1.3 Opening the Meta Data Editor . . . 20
4.1.4 Using the MetaDataEditor . . . 20
4.1.5 Additional Features . . . 21
4.1.6 Troubleshooting . . . 21
4.2 Variable Editor . . . 22
4.2.1 Introduction . . . 22
4.2.2 Requirements . . . 22
4.2.3 Accessing the VariableEditor . . . 22
4.2.4 Adding a Variable . . . 22
4.2.5 Editing a Variable . . . 22
4.2.6 Deleting a Variable . . . 23
4.2.7 Saving Changes . . . 23
4.2.8 Additional Features . . . 23
4.2.9 Troubleshooting . . . 23
4.3 Events Editor . . . 23
4.3.1 Introduction . . . 23
4.3.2 Requirements . . . 24
4.3.3 Accessing the EventsEditor . . . 24
4.3.4 Adding an Event . . . 24
4.3.5 Configuring Events . . . 24
4.3.6 Configuring Event Parameters . . . 25
4.3.7 Managing Action Sequences . . . 25
4.3.8 Configuring Action Sequences . . . 26
4.3.9 Configuring Action Components . . . 26
4.3.10 Configuring If-Else Condition Components . . . 27
4.3.11 Event Description . . . 28
4.3.12 Removing an Event . . . 28
4.3.13 Saving Changes . . . 28
4.3.14 Additional Features . . . 28
4.3.15 Troubleshooting . . . 28

5 Articy Tab **30**

6 Character Tab **31**

7 Export Tab **32**
7.1 Introduction . . . 32
7.2 Accessing the Export Functionality . . . 32

7.3 Key Components 32
7.3.1 Packages Management 32
7.3.2 Supervisor Montior 32
7.3.3 Build Process 33
7.4 Spinner and Info Boxes 33
7.5 Practical Use 33
7.6 Installation Steps 34
7.6.1 Requirements 34
7.6.2 Step 1: Connect the Pico VR Device to Your Computer 34
7.6.3 Step 2: Transfer the APK File to the Pico VR Device 34
7.6.4 Step 3: Install the APK File Using Pico VR Device 34
7.7 Launching the Application 34

8 Supervisor Monitor 36
8.1 How to Connect 36
8.2 Panels 36

1 Introduction

1.1 About

The VIA-VR platform integrates several key software components to offer a comprehensive toolkit for creating VR experiences tailored to the medical domain. At its core, the platform features a scene editor, powered by Mozilla Spoke, which allows users to compose 3D scenes easily. The behavior editor is designed to enable users to define interactive elements and narrative flows without the need for coding, facilitating the creation of dynamic VR experiences. For character representation, an avatar engine supports the generation of realistic avatars through photogrammetry, enhancing the immersive quality of VR scenarios. Additionally, the platform incorporates articy:draft for detailed story crafting and a build pipeline that leverages Unity, streamlining the process of transforming project designs into executable VR applications. These components, together with a supervisor monitor for real-time adjustments and analytics, create a user-friendly yet powerful environment for healthcare professionals to develop VR applications for education and therapy.

Project website

1.2 Getting Started

To build a VR experience, you have access to several tools within this software. Once a project is loaded or created, you can select the active tab at the top of the screen. Each tab let's you work on a particular aspect of the VR experience.

- **Objects tab**: Allows you to contruct or modify the virtual environment by adding and positioning objects, lights, etc. It operates on a what-you-see-is-what-you-get (WYSIWYG) principle, facilitating the composition of scenes with drag-and-drop. To import objcets either drag them from the bottom panel into the scene view, or drag .glb objects from your filesystem into the scene view. The bottom panel also lets you search the Sketchfab database for objects.
- **Optimize tab**: Sometimes, imported objects are too complex in data. This can lead to a reduced performance of the VR experience. The optimize tab lets you optimize objects (.gltf files only). The resulting optimized objects can then be imported into the objects tab.
- **Behavior tab**: The behavior tab lets you associate metadata tags with objects to facilitate their identification and manipulation by other parts of the system or packages. It also lets you define variables. Variables are essential for storing data such as numbers, text, or boolean values (Yes/No values) that can be used for actions and events throughout the application. Lastly, it lets you manage events. Events are triggers that initiate actions based on specific conditions, enhancing interactivity and functionality in virtual environments.

- **Avatars tab**: Offers tools for importing avatars, including the ability to create new avatars using photogrammetry with a dedicated smartphone app. This feature allows for the generation of realistic character models from photos of real people.
- **Articy tab**: Integrates with articy:draft software to enable the crafting of engaging stories within VR experiences. This modified version of articy:draft allows for the synchronization of characters between the VIA-VR platform and the story editor.
- **Export tab**: The final project can be generated within the “Export” tab. The export can be invoked when you are ready to compile and generate your virtual interactive experience.

1.2.1 Installation

VIA-VR is based on various free software components. These must be installed on your system before VIA-VR itself can be installed. You may need to ask your IT admin to carry out these steps.

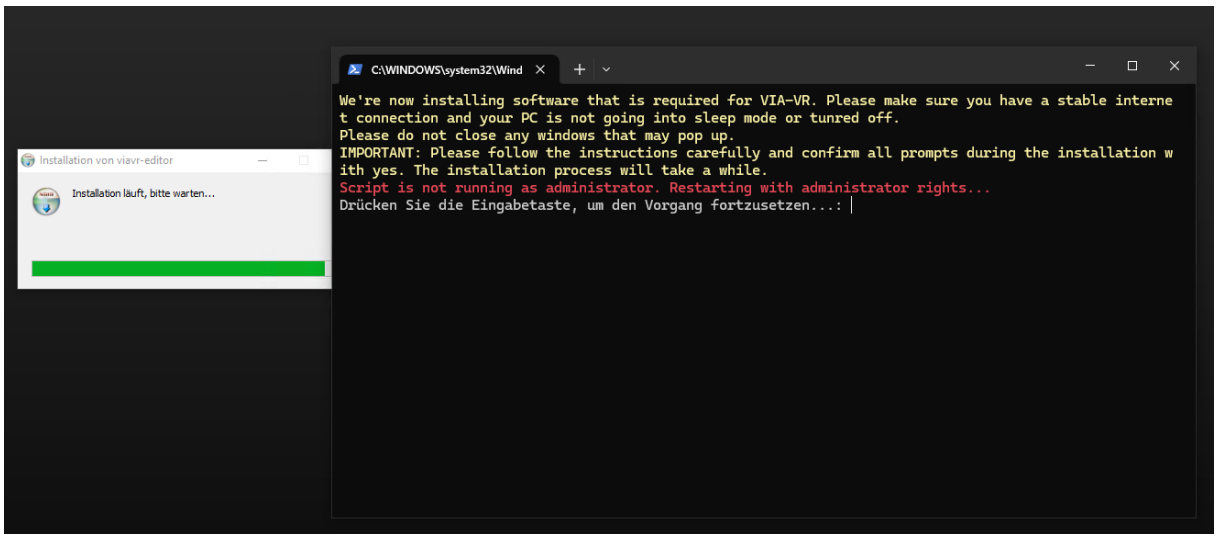
Internet Connection:

Make sure that your computer has a stable internet connection. Some components of the system may require Internet access during setup and operation.

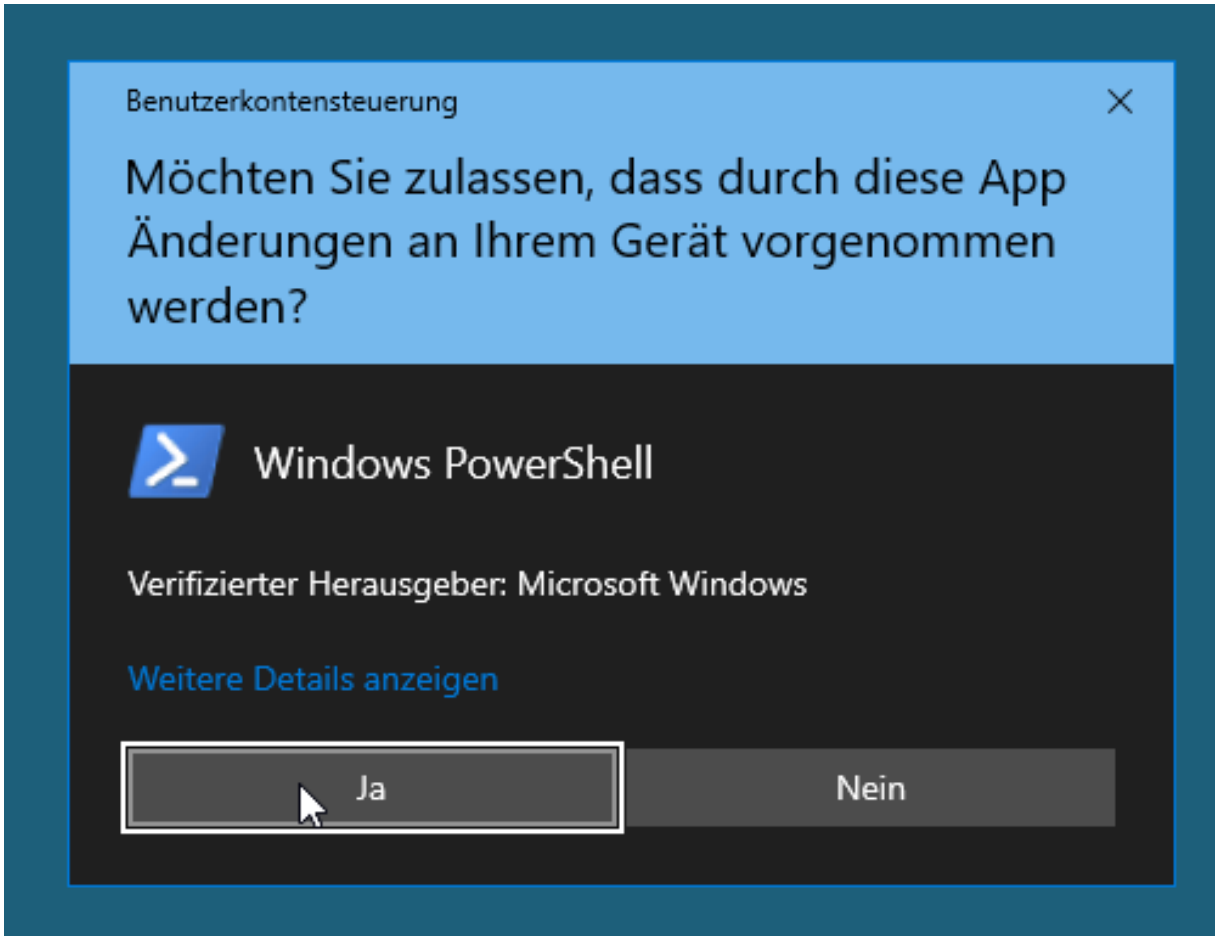
Start the InstallerÖ

The installation program will download and install all the required programs for you. You will need administrator rights on your PC for this during the installation. Please follow all instructions during the installation, confirm all dialogs with “Yes” and do not close any windows that open. Please be patient, some very large programs will be installed. This may take some time depending on your computer and your internet speed.

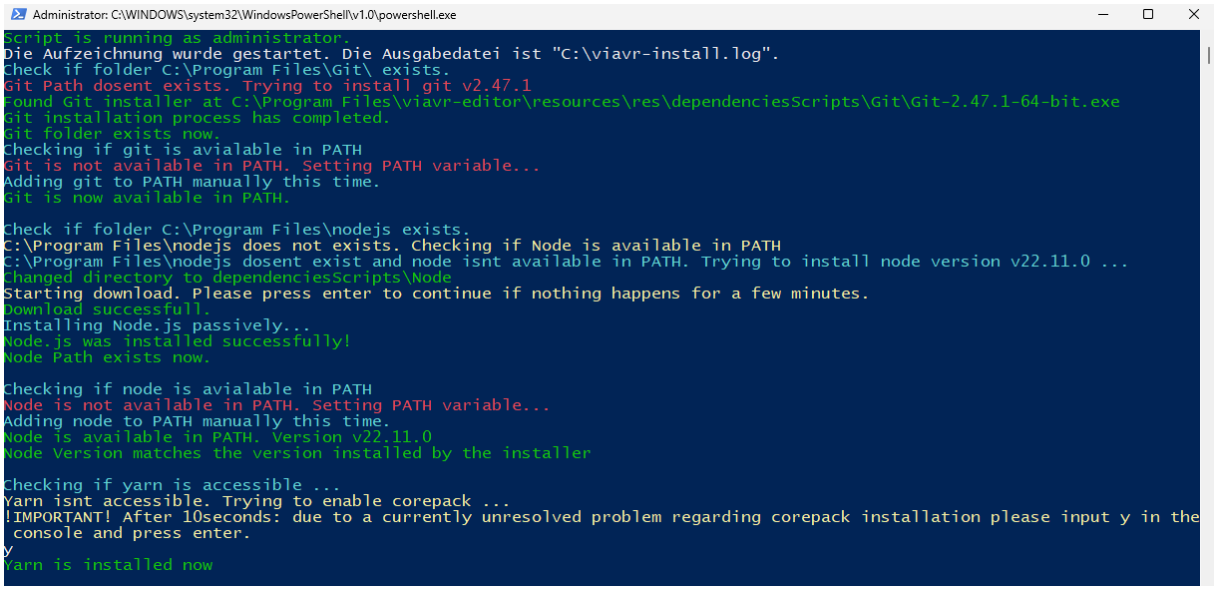
1. When prompted, press Enter to continue.



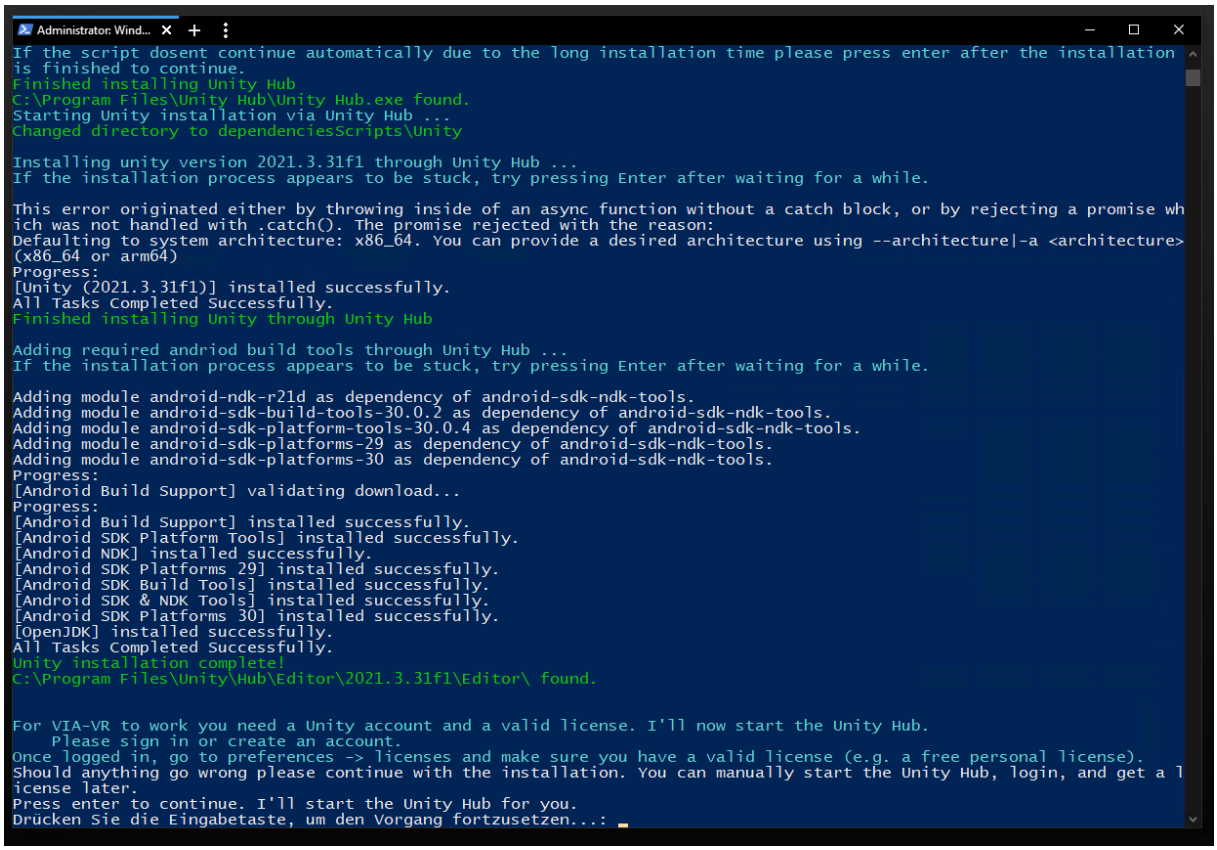
2. Bestätigen Sie die Anfrage nach Administrationsrechten.

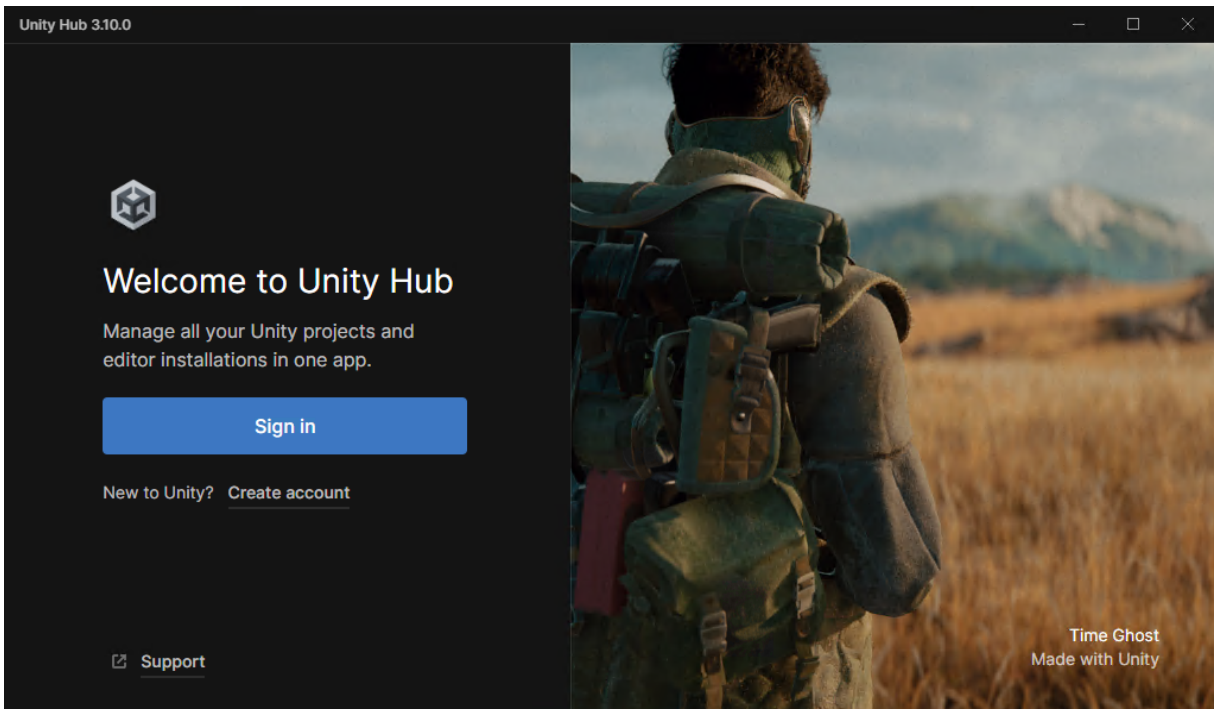


3. When prompted, press the y key so that the letter y appears in the console window. Confirm with Enter.



4. To use VIA-VR you need a Unity account and a valid license. When you are prompted to do so, press Enter. The Unity Hub program will then start. Here you must log in or create an account. Then select Preferences -> Licenses in Unity Hub and make sure that you have an active license. A free personal license is sufficient. After you have checked your license, you can close the Unity Hub program and continue with the installation. To do this, press Enter again.





5. When you are prompted to do so, press Enter.

```
Installing dependencies for Reticulum.
Starting checks for Reticulum
Cygwin is installed at 'C:\cygwin64'.
Cygwin command 'df' is available.
PostgreSQL is installed at 'C:\Program Files\PostgreSQL\16\bin'.
PostgreSQL is running on port 5432.
Elixir is installed and available in the PATH.
Folder '_build' exists in 'viavr-reticulum\'.
Folder 'deps' exists in 'viavr-reticulum\'.
Folder 'priv' exists in 'viavr-reticulum\'.
Trying to create postgres database required for Reticulum...
Could not find Hex, which is needed to build dependency :ecto_boot_migration
Shall I install Hex? (if running non-interactively, use "mix local.hex --force") [Yn] _
```

If something has gone wrong:

- If you suspect that the installation program is stuck, press Enter repeatedly to continue.
- If something goes wrong when logging into the Unity Hub, continue with the installation of the remaining software. You can then exit the Unity Hub and restart your PC if necessary. Then start the Unity Hub program yourself and try to log in again.
- If a software component has not been installed correctly, restart the installation program. Missing software components should be recognized and installed.

🔧Network configuration for the Supervisor Monitor

If you want to use the Supervisor Monitor, you must ensure that your computer and your VR device can communicate with each other. The easiest way to do this is to set up a hotspot using a smartphone to connect your computer and VR device. Your smartphone does not need an internet connection, so no mobile data is used. As an alternative to the mobile hotspot, you can ask your IT manager to enable port 3001 TCP in the local network. Make sure that your computer and VR device are connected to the same router (e.g. connected to the same WLAN) or are connected to the same network of your institution.

1.3 How to Start

Double-click the `viavr-editor.exe` to start the application.

1.3.1 🔧Initial Configuration

- When you start the application for the first time, you need to check some settings - Go to File > Preferences
- Make sure that the path to the Unity executable file (Unity.exe) is correct - Leave the other settings at their default values
- If a firewall authorization is requested at the first start, allow it

1.3.2 Welcome Screen

- We recommend that you work through the tutorial if you are using the application for the first time.
- Otherwise, either create a new project or open an existing .via file
- A project is saved in a .via file. Once a .via file has been loaded or created, you will need to restart the editor if you want to work on a different project.
- We recommend that you check our templates, i.e. ready-made, customizable VIA VR Experiences, under `[application installation folder]/res/templates`
  - in particular the templates created to demonstrate how pyrophobia exposure therapy could be supported and what an experience to promote gait rehabilitation could look like

2 Optimize Tab

2.1 Introduction

Somethimes, 3D objects (.gltf files) downloaded from the web are visually very complex. Too high complexity can lead to reduced performance of the VR application. The objects tab lets you convert complex .gltf files into optimized .glb files to enhance performance without significant loss of visual detail. We call this process mesh preprocessing.

2.2 Optimization Process

2.2.1 File Preparation

1. **File Requirement**: The component only supports the optimization of .gltf file formats. Ensure your 3D object files are in this format before proceeding.
2. **File Limitation**: Currently, optimization is limited to one file at a time to ensure quality and performance.

2.2.2 Running the Preprocessor

1. **Selecting the File**: Use the `FileDrop` area to select or drag and drop your .gltf file. The interface indicates successful file selection.
2. **Configuring Advanced Settings**: Before optimization, you can configure advanced settings by clicking on the `AdvancedSettings` button. Adjust settings according to your optimization preferences.
3. **Initiating Optimization**: With your file selected and settings configured, click the `Optimize` button to start the optimization process. The button will be disabled until a file is selected or if the system is currently processing another file.

2.2.3 Optimization Feedback

- **Processing Indicator**: Upon initiating the optimization, a loading indicator (spinner) appears, signaling that the process is underway.
- **Success Notification**: A success notification is displayed upon successful optimization, indicating that the file has been processed.
- **Error Notification**: If the file could not be processed or if an unsupported file format is selected, an error notification will appear.

2.2.4 Previewing Optimized Files

- **Preview Availability**: Upon successful optimization, a preview of the optimized 3D object (.glb format) is available in the `Preview` section.
- **Preview Conditions**: The preview is only available if the optimization process completes successfully and an optimized file is generated.

2.3 Saving Optimized Files

Optimized files are automatically named with the original file name appended with `_optimized.glb` and are saved in the same directory as the original files. This naming convention helps in easily identifying optimized versions of your 3D objects.

2.4 Troubleshooting

- **File Format Issues**: Ensure the selected file is in .gltf format. The component does not support other 3D object file formats for optimization.
- **No Preview Displayed**: If the preview does not display post-optimization, ensure that the optimization process completed successfully and check the file directory for the optimized file.
- **Performance Delays**: Large files or complex objects may extend processing times. Please be patient during the optimization process.

3 Objects Tab

3.1 Overview

The Spoke Editor is a platform designed for the creation and customization of virtual spaces. It offers a comprehensive suite of tools for users to design immersive 3D environments. This manual provides detailed instructions on utilizing the Spoke Editor's features for professional and effective virtual space creation.

3.2 Accessing the Spoke Editor

The editor is available under the "Objects" tab.

3.3 User Interface Components

The Spoke Editor interface is divided into several key areas, each dedicated to specific aspects of scene creation:

- **Assets Panel**: On the bottom left, this panel provides access to a library of 3D models, images, and other assets that can be included in your scene.
- **Hierarchy Panel**: Situated on the right, it displays the structure of objects within the scene, facilitating easy selection and organization.
- **Properties Panel**: Directly below the Hierarchy Panel, this section allows for the adjustment of properties for the currently selected object.
- **Viewport Panel**: It is the main window through which users interact with and visualize their scene. It offers a real-time 3D view of the virtual space, allowing for direct manipulation of objects and immediate feedback on changes.

3.4 Spawn Point

You must have exactly one "Spawn Point" object in the scene. This is where the VR user will be initially positioned in the virtual environment. So choose the position and orientation accordingly.

3.5 Detailed Feature Usage

3.5.1 Adding and Modifying Objects

1. **Add Objects**: From the Assets Panel, drag and drop 3D models or other assets into your scene.
2. **Select Objects**: Click on an object in the scene or in the Hierarchy Panel to select it.
3. **Modify Properties**: With an object selected, use the Properties Panel to adjust its attributes, such as position, scale, rotation, and specific model properties.

3.5.2 Object Selection

- Selecting objects within the viewport is straightforward; a simple click on an object will select it. For multiple selections, hold the shift key while clicking on additional objects. This feature is crucial for efficiently editing or transforming several objects simultaneously.

3.5.3 Camera Movement

- **Orbiting**: By left-clicking and dragging within the viewport, users can orbit around the scene, providing a comprehensive view of the environment from various angles.
- **Fly Mode**: Activated by holding the right mouse button, fly mode offers an immersive exploration experience. Users can look around the scene using the mouse and navigate using the WASD keys. This mode is particularly useful for adjusting the scene from specific vantage points or exploring the layout from a first-person perspective.
- **Focusing on Objects**: Pressing the F key allows users to quickly focus the camera on selected objects, ensuring that adjustments and inspections can be made efficiently.

3.5.4 Transform Gizmo

- With objects selected, the transform gizmo becomes visible, offering intuitive control over the position, rotation, and scale of the selected items. This gizmo is integral for precise manipulations, enhancing the user's ability to tailor the scene to their specific requirements.

3.5.5 Transformation Tools: Precision Manipulation of Virtual Assets

In the realm of 3D environment design, transformation tools are indispensable for the meticulous placement, orientation, and scaling of objects. These tools are the backbone of spatial customization, allowing designers to achieve the exact look and feel desired for a virtual space.

- **Translation Tool**: Facilitates the precise relocation of objects along the X (left-right), Y (up-down), and Z (forward-backward) axes. This tool is essential for positioning objects within the environment accurately. Additionally, the “grab and place” feature enables intuitive, freeform movement of selections, enhancing the user experience.
- **Rotation Tool**: Empowers designers to adjust the orientation of objects by rotating them around their central axis. This functionality is critical for aligning objects with the envisioned layout of the space.
- **Scaling Tool**: Allows for the adjustment of an object’s dimensions, enabling users to uniformly or non-uniformly scale objects to fit the design requirements of the environment.

3.5.6 Grid Tools: Enhancing Alignment and Placement Accuracy

The grid system offers a structural reference that significantly aids in the alignment and precise placement of objects, ensuring consistency and order within the design.

- **Transform Space Toggle**: A crucial feature that allows for the adjustment of the transformation gizmo orientation, providing the option to manipulate objects based on the world’s axes or the object’s local axes. This distinction becomes particularly relevant when dealing with rotated objects, facilitating more intuitive movement and alignment.
- **Pivot Point Adjustment**: Addresses challenges associated with the placement and rotation of objects whose model’s pivot points may not align with the desired pivot location. By adjusting the pivot calculation, users can achieve more accurate positioning and rotation based on the object’s center, bottom, or a specific selection.
- **Snap Mode Activation**: Enhances placement precision by enabling objects to align or snap to specific positions or rotations based on the grid. This feature is instrumental in achieving exact alignment and uniformity in object placement.
  - **Snap Mode Controls**: Define the granularity of the snapping mechanism, allowing for meticulous control over the transformation tool’s application.
- **Grid Visibility and Elevation Controls**: Offer the ability to toggle the grid’s visibility and adjust its elevation. These features are invaluable for vertical construction projects, providing a reference plane at various heights.

3.5.7 Lighting

- **Add Lighting**: Enhance your scene with realistic lighting. Adjust light properties like color, intensity, and shadows in the Properties Panel for dynamic effects.

3.6 Saving Projects

- **Saving**: Regularly save your project by clicking the “Save” button in the Toolbar to avoid data loss (hotkey: ctrl + s).

3.7 Conclusion

The Spoke Editor offers a robust platform for creating detailed and interactive 3D environments. For further exploration of Spoke’s capabilities, please refer to the complete documentation available at the Mozilla Hubs Documentation.

4 Behavior Tab

4.1 Meta Data Editor

4.1.1 Introduction

The MetaDataEditor is a user interface component designed for tagging scene objects within a virtual environment platform, such as VIA-VR. It allows users to associate metadata tags with objects to facilitate their identification and manipulation by other parts of the system or packages.

4.1.2 Requirements

Before using the MetaDataEditor, ensure you have the following:

- An active instance of the application where the MetaDataEditor component is integrated.
- Scene objects already loaded or created within the application.

4.1.3 Opening the Meta Data Editor

The MetaDataEditor is accessible within the settings panel of the application. To open it:

1. Navigate to the settings panel.
2. Locate and expand the `MetaData` accordion section.

4.1.4 Using the MetaDataEditor

Selecting a Scene Object

1. Click on the dropdown menu labeled as “Select an object…”.
2. Choose an object from the list of scene objects. These objects are loaded dynamically from your scene.

Tagging Objects

1. With an object selected, click on the `Select tags...` button to open the tag selection menu.
2. The menu displays a list of available tags that can be assigned to the selected object.
3. Click on a tag to select it. You can select multiple tags.
4. To deselect a tag, click on it again in the selection menu.

Adding Tags to the Selected Object

1. Once you have selected one or more tags, click on the Add Tags button to apply them to the selected object.
2. If no tags are selected and you attempt to add tags, an error message will be displayed prompting you to select at least one tag.

Removing Tags

1. To remove an existing tag from an object, locate the object's entry in the displayed list under the MetaData section.
2. Each tagged object will have its tags displayed, along with a close icon next to each tag.
3. Click on the close icon next to a tag to remove it from the object.

Saving Changes Changes are automatically saved when you add or remove tags from an object. There is no need for a separate save action.

4.1.5 Additional Features

- **Tooltip Help**: Hover over the help icon (a question mark) next to the Meta Data title to view a tooltip providing more information about object tagging.
- **Dynamic Tagging Feedback**: The interface dynamically updates to reflect the available tags for selection based on the current context and selections, ensuring you can only assign relevant and non-duplicate tags to each object.

4.1.6 Troubleshooting

- **Objects or Tags Not Appearing**: Ensure the scene objects are correctly loaded into the application and that the MetaDataEditor component is properly initialized.
- **Changes Not Saving**: Confirm that you have network connectivity and that the backend services (if applicable) are running correctly.

For further assistance, refer to the application's support documentation or contact the support team.

4.2 Variable Editor

4.2.1 Introduction

The VariableEditor is a user interface component designed to manage variables within a software application. Variables are essential for storing data such as numbers, text, or boolean values (Yes/No values) that can be used for actions and events throughout the application.

4.2.2 Requirements

Before interacting with the VariableEditor, ensure the following prerequisites are met:

- The software application where the VariableEditor component is integrated is properly installed and running.
- You have necessary permissions or roles required to modify variables within the application.

4.2.3 Accessing the VariableEditor

The VariableEditor can be found within the settings panel of the application. To access it:

1. Navigate to the settings panel of your application.
2. Look for the `Variables` section and click on it to expand the VariableEditor interface.

4.2.4 Adding a Variable

1. Click on the `Add Variable` button located at the bottom of the VariableEditor section. This action will create a new row in the table for variable entry.
2. Enter the desired variable name in the `Name` column text input.
3. Select the variable type (Number, Text, or Yes/No Value) from the dropdown in the `Type` column. The type determines the kind of data the variable will store.
4. Based on the selected type, enter the default value for the variable in the `Default Value` column:
    - For `Number` or `Text` types, directly input the default value.
    - For the `Yes/No Value` type, select either “Yes” or “No” from the dropdown.

4.2.5 Editing a Variable

1. To edit a variable’s name, type, or default value, locate the variable row in the table.

2. Modify the variable's name by directly editing the text in the `Name` column's text input.
3. Change the variable type by selecting a different option from the dropdown in the `Type` column. Note that changing the type may reset the default value.
4. Update the default value by editing the input in the `Default Value` column, corresponding to the selected variable type.

4.2.6 Deleting a Variable

1. Find the variable you wish to delete in the table.
2. Click on the `Delete` button located in the row of the variable you intend to remove. The variable will be immediately deleted from the list.

4.2.7 Saving Changes

Changes to variables (including additions, edits, and deletions) are automatically saved as you make them. There is no separate save action required.

4.2.8 Additional Features

- **Tooltip Information**: Hover over the help icon (a question mark) next to the `Variables` title to view a tooltip providing more information about variables and their use cases.

4.2.9 Troubleshooting

- **Variables Not Saving**: Ensure you have a stable connection to the backend services, and there are no errors in the console related to network requests.
- **Unable to Edit/Delete Variables**: Confirm that you have the necessary permissions to modify the variables. If the issue persists, contact your system administrator.

For further assistance or inquiries regarding the VariableEditor, refer to the software application's help documentation or contact the support team.

4.3 Events Editor

4.3.1 Introduction

The EventsEditor is a component designed to manage and configure events within the VIA-VR platform. Events are triggers that initiate actions based on specific conditions, enhancing interactivity and

functionality in virtual environments. This manual provides instructions on how to interact with the EventsEditor to add, configure, and manage these events.

4.3.2 Requirements

Before using the EventsEditor, ensure the following conditions are met:

- The VIA-VR platform or a compatible application with EventsEditor integration is running.
- You have the necessary permissions to modify events within the application.

4.3.3 Accessing the EventsEditor

The EventsEditor is accessible within the settings or configuration panel of the VIA-VR platform. To open it:

1. Navigate to the settings or configuration panel.
2. Locate and expand the `Events` section to access the EventsEditor interface.

4.3.4 Adding an Event

1. Click on the `Select event...` button to open the event selection menu.
2. Browse through the list of available events and select the one you wish to add. The events list is dynamically loaded based on the enabled packages within your VIA-VR platform.
3. Once an event is selected, its name will be displayed on the button.
4. Click the `Add Event` button to officially add the selected event to your scene or project. If no event is selected, a prompt will appear, reminding you to select an event.

4.3.5 Configuring Events

After adding an event, you can configure its parameters and associated actions:

1. Each event you add will appear as a separate entry in the EventsEditor.
2. Select an event to expand its configuration options, where you can:
    - Adjust event-specific parameters, if applicable.
    - Define the action sequence that should be triggered by the event.
3. Use the provided UI components to add, remove, or modify the actions and parameters of the event as required.

4.3.6 Configuring Event Parameters

After adding an event to your scene or project, you can configure its parameters to fine-tune its behavior:

1. **Accessing Parameter Configuration**: Click on an event you've added to expand its details. If the event has configurable parameters, they will be listed under the "Parameters" section.
2. **Setting Parameter Values**:
    - **General Parameters**: For parameters expecting a text input (e.g., numeric values, strings), directly type the desired value into the text field next to the parameter's name.
    - **Object Selection**: For parameters named `gameObject` or `other`, indicating the selection of a scene object, use the dropdown menu to select one of the available scene objects. These objects are dynamically loaded based on the current scene configuration.
    - **Tag Selection**: For parameters named `tag`, which require specifying a tag, use the dropdown menu to select from predefined tags such as "Avatar", "Floor", "Teleport Anchor", "Collectable", etc.
3. **Updating Parameters**: After setting a parameter value, the system automatically updates the event configuration. These updates are immediately reflected in the EventsEditor and saved.

4.3.7 Managing Action Sequences

Events in VIA-VR can trigger sequences of actions. The `ActionSequence` component within each event allows for the creation and management of these sequences:

1. **Accessing Action Sequence Configuration**: Within the expanded details of an event, the action sequence section is available at the bottom.
2. **Adding Actions to the Sequence**: Use the provided interface to add actions to the sequence. This typically involves selecting actions from a predefined list and configuring their parameters.
3. **Ordering Actions**: Actions within the sequence execute in the order they appear. You can adjust the order as needed to achieve the desired behavior. The EventsEditor supports drag-and-drop functionality for reordering components within an action sequence. Simply click and hold a component, then drag it to its new position and release. This allows you to fine-tune the order of operations for actions and conditions.
4. **Removing Actions**: Each component can be removed from the sequence by clicking the `Delete` or `Close` button (represented by a cross icon) associated with that component. This allows you to easily manage the components within your action sequence.

5. **Saving Action Sequences**: Changes to action sequences are automatically saved upon modification, ensuring that your event behavior is updated in real time.

4.3.8 Configuring Action Sequences

Action sequences allow for the definition of a series of actions that are executed when an event is triggered. These actions can be straightforward commands or complex conditional operations.

Adding Components to an Action Sequence

1. **Action Components**: To add an action to the sequence, click the `Add Action` button. This will introduce a new action component where you can define the action's specifics, such as the action type and parameters.
2. **If-Else Condition Components**: To incorporate conditional logic, click the `Add If-Else Condition` button (available up to a depth of 3 to prevent excessive complexity). This allows you to specify conditions under which certain actions are performed. You can define the condition based on variables, comparison operators, and then specify actions for both the true (then) and false (else) outcomes.

Configuring Components

- Each component added to the sequence can be configured individually.
- For action components, specify the desired action and any necessary parameters.
- For if-else components, define the variable, comparison operator, comparison value, and actions to execute for both outcomes of the condition.

4.3.9 Configuring Action Components

Action components define specific actions to be executed as part of an event's action sequence.

Selecting an Action

1. To add an action, click the `Add Action` button within the ActionSequence component.
2. Use the `Select action` menu to choose an action from the available options. These options are dynamically populated based on the actions defined within the VIA-VR platform.
3. Once selected, the chosen action's name will display on the button, indicating the active selection.

Configuring Action Parameters

- After selecting an action, configure its parameters according to the needs of your event. This may involve setting values for predefined parameters, selecting scene objects, or choosing tags.
- For object-related parameters, use the dropdown to select one of the available scene objects.
- For tag parameters, select the appropriate tag from another dropdown menu.
- Directly input values for other parameter types using the provided text input fields.

4.3.10 Configuring If-Else Condition Components

If-else condition components introduce conditional logic into the action sequence, allowing for different actions based on specified conditions.

Defining the Condition

1. **Variable Selection**: Use the `Select variable` menu to choose a variable that the condition will evaluate. This menu is populated with variables defined within the VIA-VR platform.
2. **Operator Selection**: Choose the comparison operator (e.g., equals, does not equal, is greater than, is less than) that determines how the variable's value will be compared against the comparison value.
3. **Comparison Value**: Set the value to compare the variable against. The input method for the comparison value depends on the variable's type.

Specifying Then and Else Actions

- **Then Actions**: Define the action sequence to execute if the condition evaluates to true. Use it to add, configure, and order the actions.
- **Else Actions**: Similarly, define the action sequence for the scenario where the condition evaluates to false.

Advanced Sequencing and Nesting

- Actions and if-else conditions can be nested within each other to create complex logic structures. The EventsEditor supports nesting up to a depth of 3 to ensure manageability and performance.
- Utilize drag-and-drop functionality to reorder components within the action sequence, tailoring the event's behavior to precise requirements.
  - **Drag Handle**: Use the provided drag handle to click and drag components to rearrange the sequence order.

  - **Visual Feedback**: As you drag a component, the interface provides feedback on where the component can be placed within the sequence.

4.3.11 Event Description

For events that include a description, this information is displayed within the event's expanded details section, providing additional context or instructions regarding the event's purpose and usage.

4.3.12 Removing an Event

1. To remove an event, locate the event entry you wish to delete.
2. Click the `Close` or `Delete` icon (represented by a cross icon) next to the event's name to remove it from your configuration.

4.3.13 Saving Changes

- All changes made within the EventsEditor are automatically saved upon modification. There is no need for a manual save action.
- Ensure you properly configure each event and its associated actions to prevent unintended behavior in your VIA-VR environment.

4.3.14 Additional Features

- **Tooltip Help**: Hover over the help icon (a question mark) next to the `Events` title to view a tooltip providing more information about events and their functionality.

4.3.15 Troubleshooting

- **Events Not Loading**: Ensure that your VIA-VR platform or application is correctly connected to its backend services and that all required packages are properly installed.
- **Actions Not Executing**: If actions within an event do not execute as expected, check the order and configuration of the action sequence to ensure that all actions are correctly set up.
- **Issues with Selecting Variables or Actions**: Verify that all required backend services are operational and that the VIA-VR platform is up to date with the latest configurations and definitions for actions and variables.

- **Problems with Conditional Logic**: Ensure that conditions are correctly defined with valid variables, operators, and comparison values. Incorrectly configured conditions may lead to unexpected behavior.

5 Articy Tab

You can use articy:draft 3 to write engaging dialogs that are displayed next to non-player characters in Virtual Reality.

- First, press the “Open Articy Editor” and create dialogs
  - “Entities” in articy:draft correspons to avatars in the VIA-VR application. They are synchronized
  - Please refer to the articy:draft documentation on how to create dialogs
  - We currently support branching with max. 2 options
  - Keep in mind to keep individual dialog fragments short, otherwise there might be display issues in VR
  - Don’t forget to save the articy project before closing articy:draft (ctrl + s)
- Use the dropdowns in the Articy tab to map a dialog to an avatar

6 Character Tab

> You need access to the University of Dortmunds (TUD) character pipeline to create a virtual character of a real person.

Use the character editor to create scanned characters via the University of Dortmund character pipeline. (Non-player) character in this contect refers to a virtual person within the VR experience. We sometimes also refer to them as avatars.

To add a new character, follow these steps:

- Enter the name for your new character in the textbox at the bottom
- Click on the “Add character” button
- Go to the objects tab, and place a group object in the scene. This will be used as placeholder. In VR, the non-player character will appear at this position. Save the project (ctrl + s) before switching back to the character tab
- Assign the character to the placeholder object using the dropdown
- Select “show QR code” to see the code that you have to scan with the TUD’s character scan app
- Please consult documentation provided by TUD for more details
- Once the character is done processing, use the download button to download the character

7 Export Tab

7.1 Introduction

This manual provides guidance on using the Export interface.

7.2 Accessing the Export Functionality

The final project can be generated within the “Export” tab. The export can be invoked when you are ready to compile and generate your virtual interactive experience.

7.3 Key Components

7.3.1 Packages Management

- **Fetching Packages**: The interface fetches and displays available Unity packages for inclusion in your VIA experience.
- **Selecting Packages**:
  - Toggle the selection of packages as needed. Mandatory packages are pre-selected and cannot be deselected.
  - For selected packages with additional configuration requirements, the interface provides access to advanced configurations.

7.3.2 Supervisor Montior

- If you want to use the supervisor montior, check the corresponding checkbox
- If you additionally want to create a floor map of the virtual environment, check the corresponding checkbox and click the “create floor map” button
- If you want to add triggers to the floor map, install the corresponding trigger packe (e.g. fire triggers) before creating the floor map
- To create a floor map: first draw the floor map in the embedded draw.io editor
  - You can press the “more shapes” button to access “floorplans” elements
  - You can also past images from your clipboard into the editor
  - We strongly recommend you to first go the objects tab and move the camera to a top-down perspective. Make sure to turn the camera so that the blue arrow (z-axis) points to the top and the red arrow (x-axis) points to the left. You can take a screenshot and paste this in the floor map editor for easier editing.

  - Once you are satisfied with the floor map in step 1, press “save & exit”
  - Define triggers in step 2: drag triggers from the top left to the desired place in the floor map. Click on the new trigger to select a corresponding scene object, the trigger, and the default value of the trigger
  - Make sure to press “save” once you are done

- If you use a floor map, make sure to designate the part of the virtual environment that the VR user might be in. To this end, go to the objects tab and move the camera to a top-down perspective. Make sure to turn the camera so that the blue arrow (z-axis) points to the top and the red arrow (x-axis) points to the left. Now add two group elements to the scene. Name one “top right” and one “bottom left”. Move these objects to their corresponding place in the VR environment. Make sure to place these objects precisely, so that the VR user’s position can be acurately displayed in the floor map. Now save the project (ctrl + s) and go to the behavior tab. Tag “top right” and “bottom left” with their corresponding tags.

7.3.3 Build Process

- **Initiating Build**:
  - Click on “Generate Experience” to start the build process.
  - You can choose to save your current project and continue with the build, or proceed without saving.
- **Build Status**: The interface displays a loading indicator during the build process, with informative messages providing status updates.

7.4 Spinner and Info Boxes

- **Fetching Information**: Spinner indicators and information boxes are displayed while the interface fetches package details or during the build process.

7.5 Practical Use

1. **Configure Packages**: Review available packages, adjust selections as necessary, and configure settings for selected packages.
2. **Generate Experience**: Initiate the build process by clicking “Generate Experience.” Choose to save changes if prompted.
3. **Monitor Progress**: Watch for spinner indicators and toaster notifications for status updates.

7.6 Installation Steps

7.6.1 Requirements

- A Pico VR device
- A computer
- A USB cable compatible with your Pico VR device
- The APK file you wish to install

7.6.2 Step 1: Connect the Pico VR Device to Your Computer

1. Ensure your Pico VR device is powered on.
2. Connect the device to your computer using the USB cable.
3. On your computer, open the file explorer. Your Pico VR device should be recognized as an external storage device.

7.6.3 Step 2: Transfer the APK File to the Pico VR Device

1. Locate the APK file on your computer.
2. Copy the APK file.
3. Navigate to your Pico VR device's storage via the file explorer.
4. Paste the APK file into a suitable location on the device's internal storage.

7.6.4 Step 3: Install the APK File Using Pico VR Device

1. Disconnect the Pico VR device from the computer.
2. Put on your Pico VR headset and navigate to the File Manager application.
3. In the File Manager, locate the APK file you transferred. You may need to select the option to view all APK files.
4. Select the APK file to initiate the installation process. Follow any on-screen instructions to complete the installation.

7.7 Launching the Application

Once the application is installed:

1. Navigate to the Library tab on your Pico VR device.

2. Locate the newly installed application in the list of available applications.
3. Select the application to launch it.

8 Supervisor Monitor

The supervisor monitor is a separate application that can be used by another person during a VR session. Using the supervisor monitor, the supervisor is - among other things - able to see what the VR user sees.

8.1 How to Connect

- You can user either run the installer to install the supervisor monitor on your system, or use the portable version
- Double-click `viavr-supervisor-monitor.exe` to start the supervisor monitor
- Now start the VR app on your VR device
- The VR user now has to enter the IP address (e.g. 192.168.0.1) displayed on the supervisor monitor
  - To enter the IP address, the VR user has to point on the number buttons and select a button by pressing the trigger button
  - Once the IP address is entered, the VR user has to press the enter button to confirm
- Once the connection is establised, the supervisor sees a panel layout. This layout is customizable, i.e. each panel is resizable, can be closed and reopend or spectated in fullscreen mode

8.2 Panels

- Stream: let's the supervisor see and hear what the VR users experiences. Use the volume slider to change the volume or mute the stream
- Protocol: let's the supervisor take notes. Certain actions are automatically added to the protocol. When the supervisor quits the supervisor monitor, he is asked whether he wants to save the protocol
- Floor map: requires the floor map feature to be enabled. If a floor map was configured, the supervisor monitor sees a top-down perspective of the virtual environment, the VR users position and orientation, an objects of interest (so called "triggers")
  - If triggers exist in the floor map, the user can press on the corresponding red icon to select the state of the trigger. The state will be automatically updated in the virtual environment. Example "fire exposure therapy": the virtual environment contains a fireplace which is indicated as a red icon in the floor map. The supervisor can turn on the fire place and select the size of the fire while the VR user is experiencing the VR application.