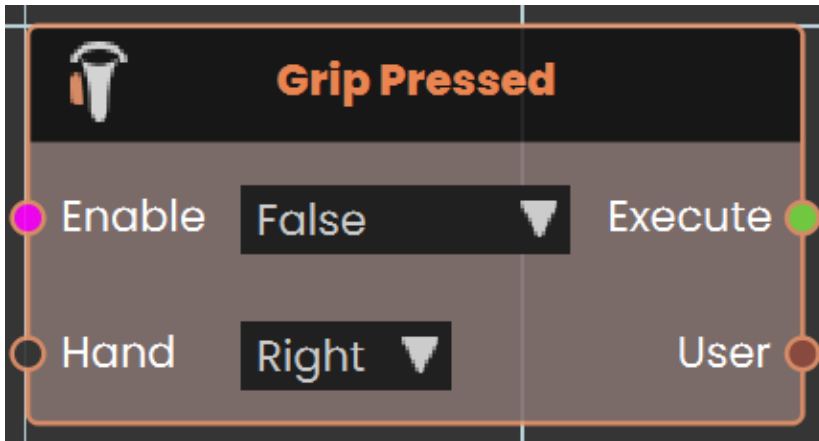


# Events

- Controller \ Grip
- SceneNode \ Triggered
- Voice Command
- AI Agent
- Variable
- On Message Received

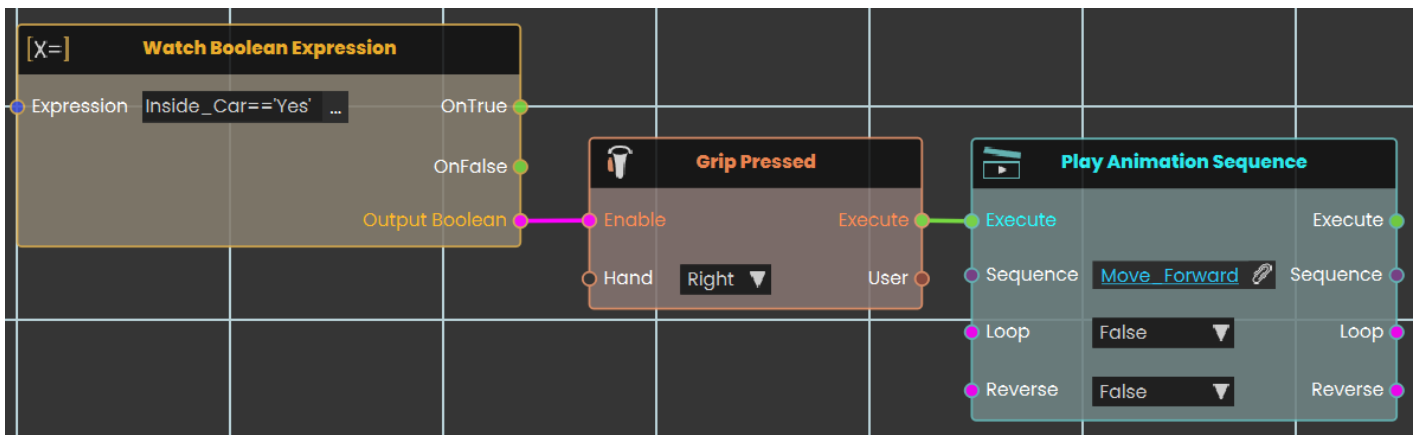
# Controller \ Grip

## Grip Pressed



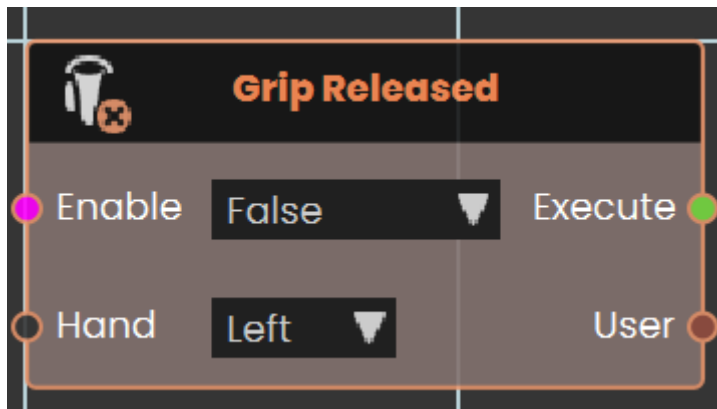
The **Grip Pressed node** enables the user to detect when the VR controller's grip is pressed. By connecting a response to this node, the specified action is executed each time the grip is pressed, allowing for interactive controls within the VR Experience.

## Example



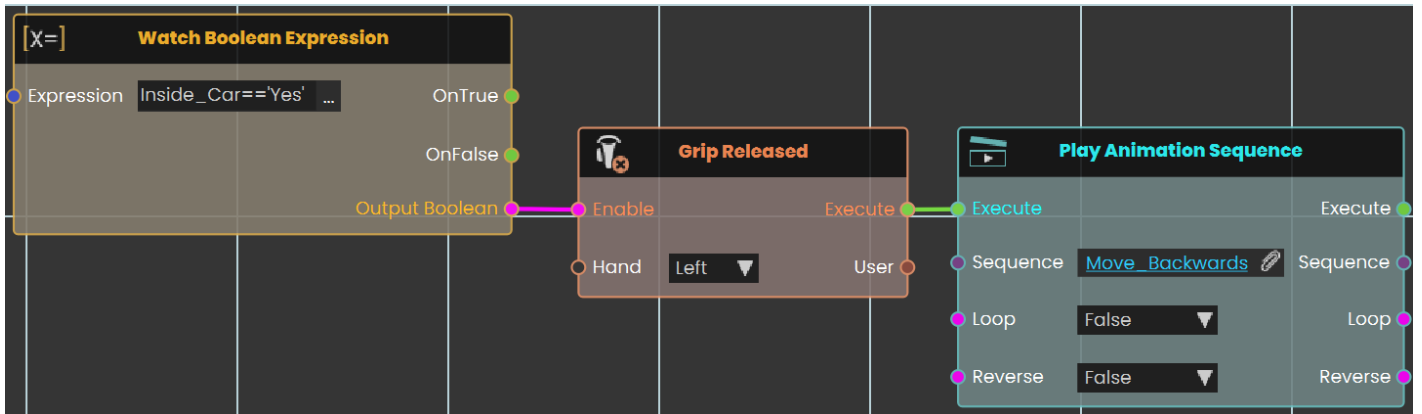
In this example, a **Grip Pressed node** is used to execute a response. Once the Grip press is triggered, the Grip Pressed node checks the boolean value. If the condition is met, the connected response executes, and the animation plays during the VR Experience.

## Grip Released



The **Grip Released node** enables the user to detect when the VR controller's grip is released. By connecting a response to this node, the specified action is executed each time the grip is released, allowing for interactive controls within the VR Experience.

# Example

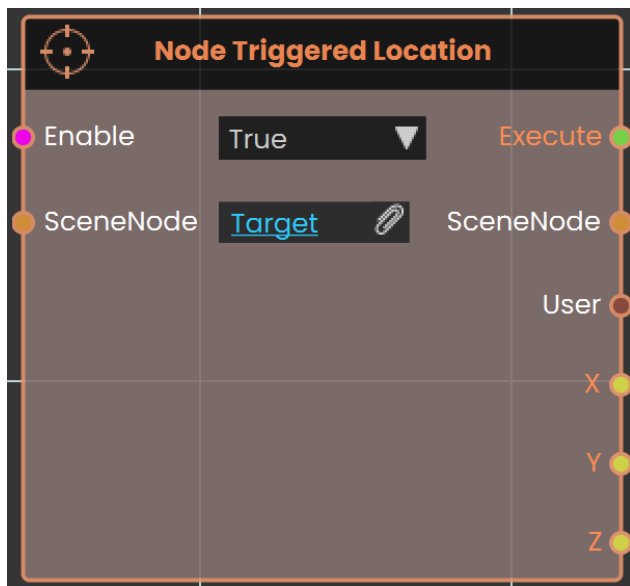


In this example, a **Grip Released node** is used to execute a response. Once the Grip released is triggered, the Grip Released node checks the boolean value. If the condition is met, the connected response executes, and the animation plays during the VR Experience.

---

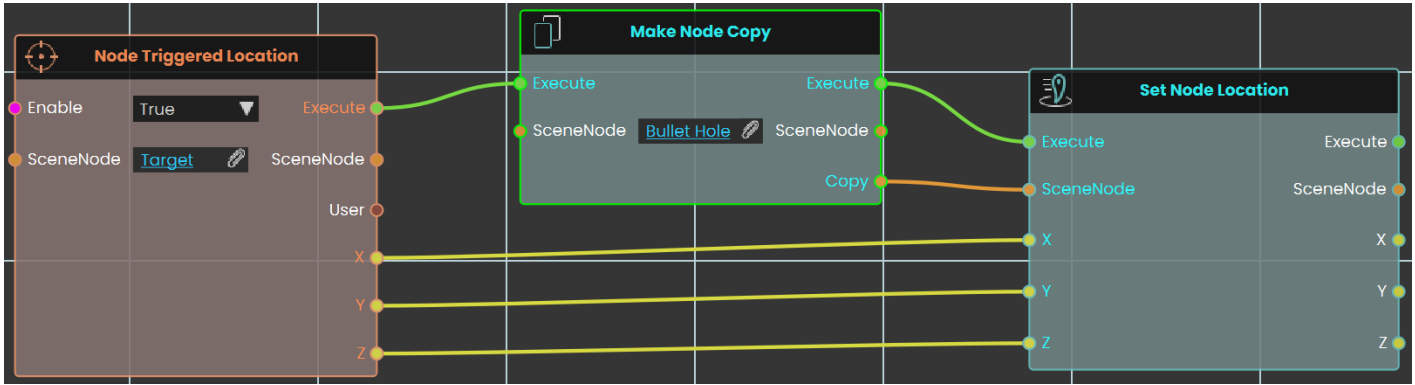
# SceneNode \ Triggered

## Node Triggered Location



The **Node Triggered Location** is used to capture the specific location or position of a triggered event within the VR environment. It enables tracking where a particular interaction occurred, which can be useful for dynamic object placement, analytics, or event-triggered responses based on spatial coordinates.

## Example



In this example, the **Node Triggered Location** is used to capture the location where the user triggers the target. When the target is hit, the node records the exact spot, and a bullet hole is placed using the **Set Node Location**. The **Make Node Copy** is used to enable the user to shoot more than one bullet, creating a new bullet hole at each triggered location. This setup allows for repeated interactions, with each shot creating a new bullet hole in the correct spot.

- ■ ■
- ■
-

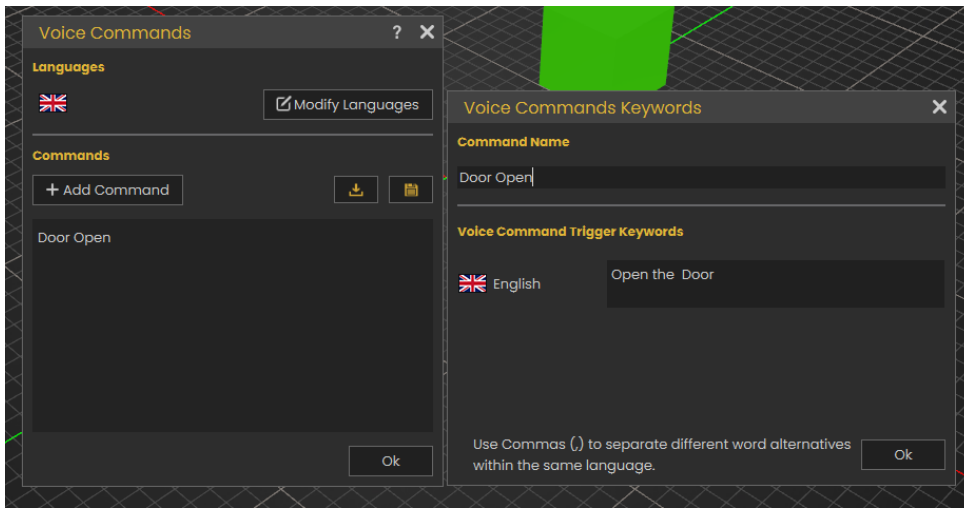
# Voice Command

## Voice Command Recognized

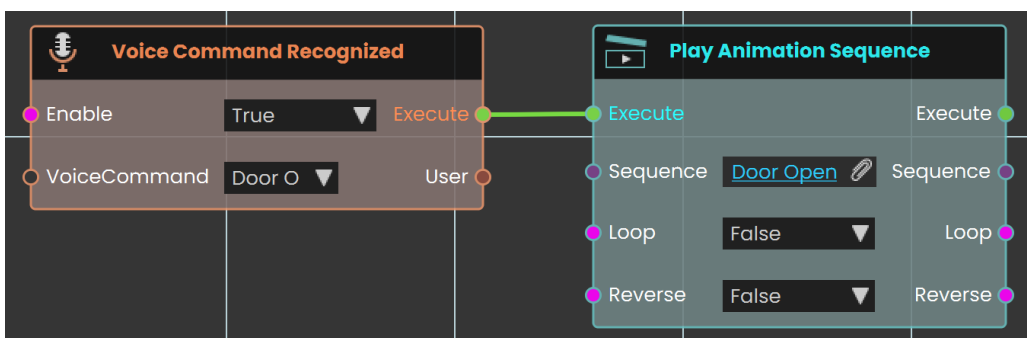


The **Voice Command Recognized event** is used to activate a response when the user says a specific command. This event listens for predefined voice commands and triggers the associated actions or responses within the VR environment upon recognizing the command.

## Example



In this example, a new voice command named "Door Open" is created by accessing the Voice Command section from the Interaction menu and adding the command in the Voice Commands window.



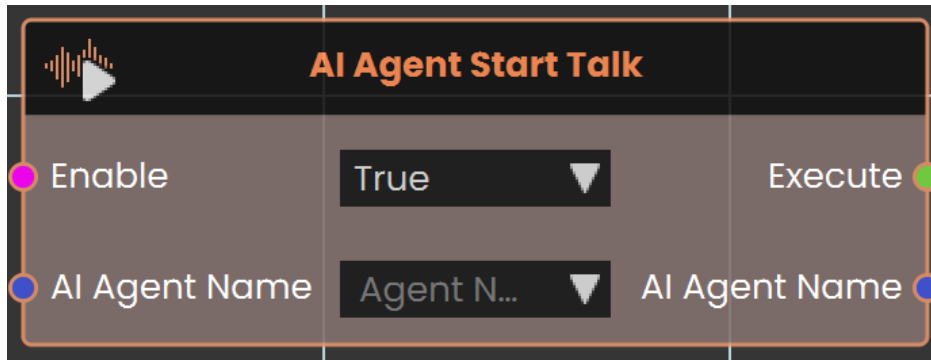
The **Voice Command Recognized event** is used to link the newly created "Door Open" command. When the user says "Open the door", the door open sequence is triggered and plays.





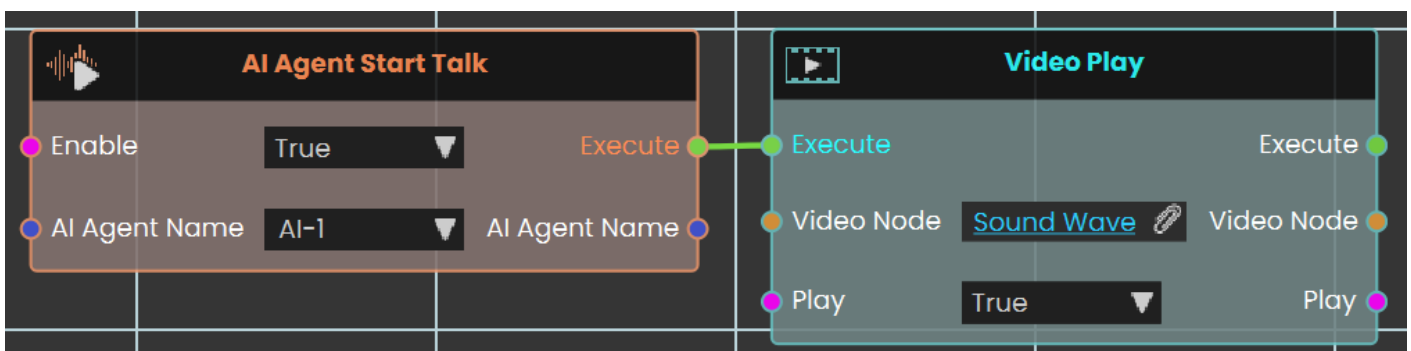
# AI Agent

## AI Agent Start Talk



The **AI Agent Start Talk** event activates when the AI agent begins speaking. This event allows users to manage and synchronize actions with the AI's speech, ensuring that specific elements or behaviors respond dynamically as the AI starts talking.

## Example

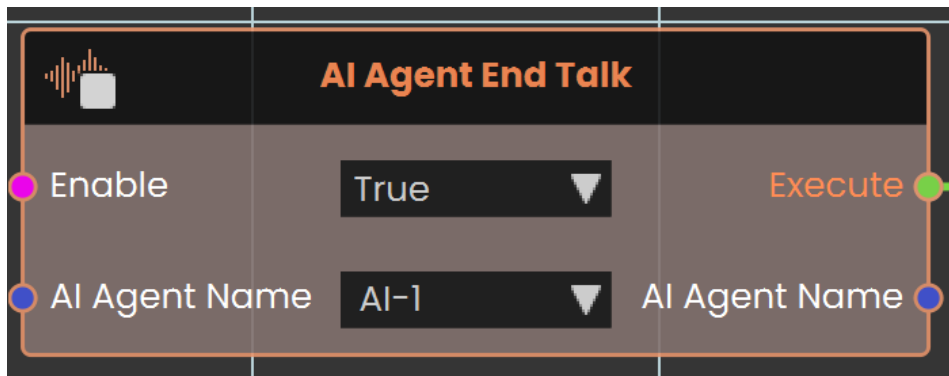


In this example, the **AI Agent Start Talk** event is used to play a "sound wave" video when the AI begins speaking. As soon as the AI starts talking, the "sound wave" video is triggered, visually representing the AI's speech in real-time.



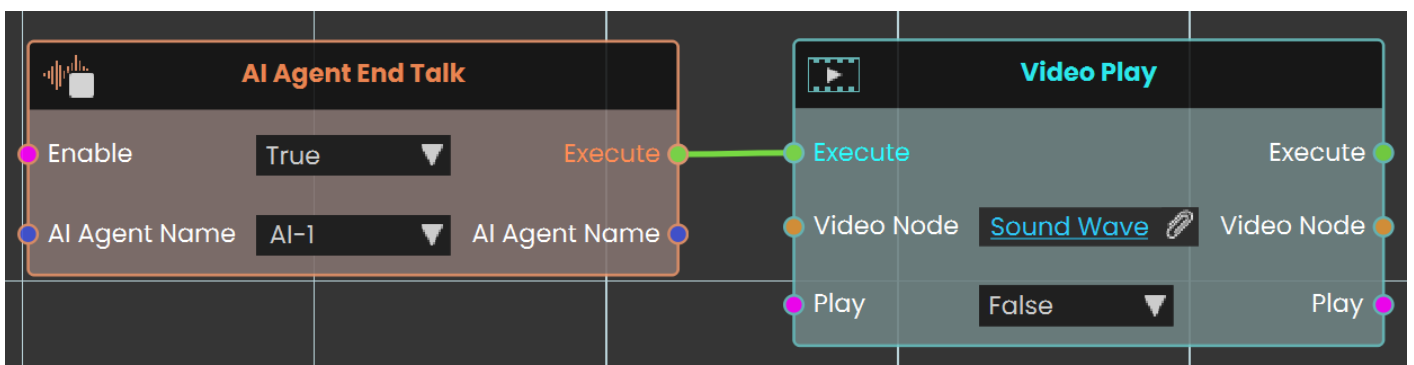
Tutorial is available on SimLab VR Discord server

## AI Agent End Talk



The **AI Agent End Talk** event activates when the AI agent finishes speaking. This event helps users control what happens after the AI stops talking, allowing for smooth transitions and responsive interactions in the VR Experience.

## Example



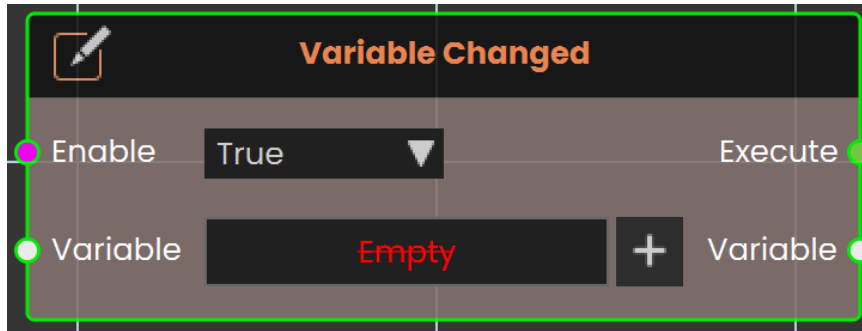
In this example, the **AI End Talk** event is used to stop the "sound wave" video when the AI finishes speaking. Once the AI stops talking, the video is deactivated, indicating the end of the speech.



Tutorial is available on SimLab VR Discord server

# Variable

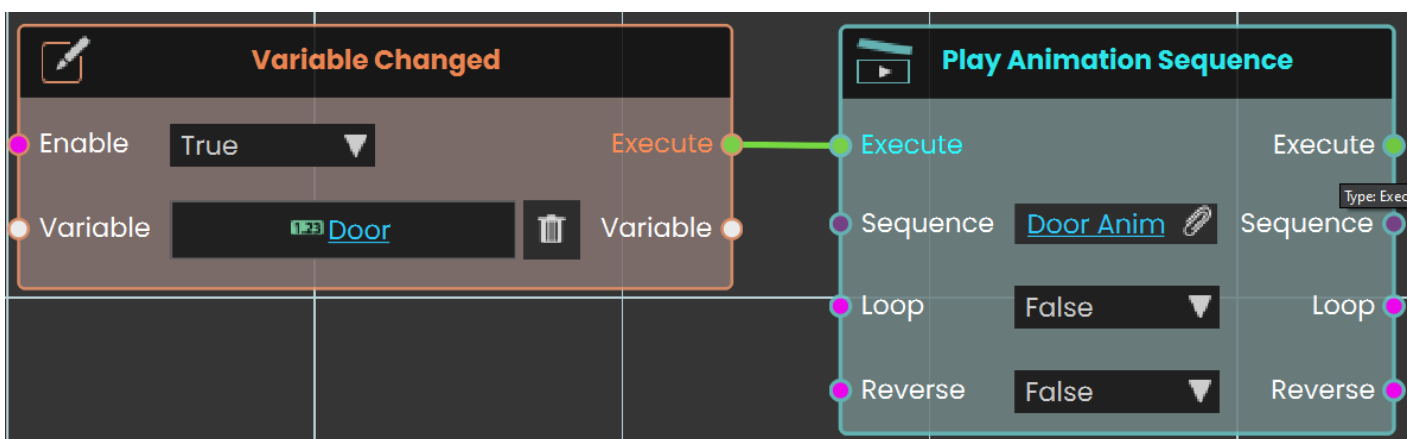
## Variable Changed



The **Variable Changed** event

activates when the value of a specified variable is modified. This event helps users detect real-time changes and trigger actions accordingly, enabling dynamic and responsive interactions within the VR Experience.

## Example



In this example, the **Variable Changed** event is set to monitor changes in the variable **Door**. When the value of this variable is modified, the event triggers the **Play Animation Sequence** node, which activates the **Door Anim** animation sequence.

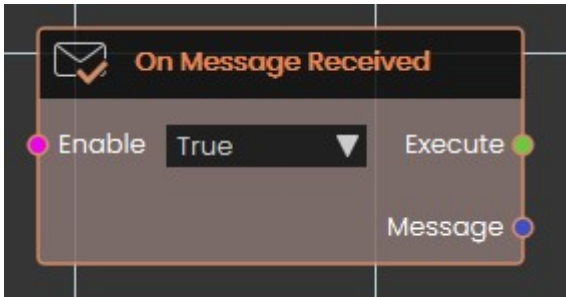


# On Message Received

SimLab Composer 14 introduces WebSocket support, enabling seamless integration with external systems, allowing it to work with external hardware, co-simulation engines, and websites, unlocking numerous possibilities.

Use the node **On Message Received** to trigger an event when a message is received.

The **On Message Received** node triggers an event when an unhandled message is received from the external connection. This node allows you to access the incoming message and take appropriate action based on its content. It is useful for responding to external data or commands and processing messages within the VR Experience.



## Input Ports:

- Enable

## Output Ports:

- Execute
- Message

The WebSocket nodes (Open External Connection, Send Message, and Receive Message) are exclusively available in the Ultimate Edition.