# General

Archi-VR is an application to preview architectural projects in <virtual Reality (VR) on the Oculus Quest device. The application can be built with a set of Archi-VR projects.

# Archi-VR project

An Archi-VR project represents a snapshot of a physical architectural design for a physical construction project.

Single project build  
 For a single construction project, several project scan be prepared and included in a single Archi-VR build fort that construction project.

Original/Before  
Cleanup/Teardown  
Construction Phase RuwBouw  
Construction Phase Technieken  
…  
Construction Phase Final

This is called a ‘single project build’

## Portfolio build

Likewise, several Archi-VR projects for different construction project scan be added to a single Archi-VR build.

Project 001  
Project 002

This is called a ‘portfolio build’

Archi-VR Trainstation

* Projects
  + Original State
  + Proposal A
  + Proposal B
  + Proposal C

The Archi-VR application has the concept of an ‘active project’ this is the project currently being previewed. The user can toggle between the active project by cycling the projects included in the build.

When switching to a new project, and the project contains a POI with the same name as the active POI, then that POI is automatically activated.

Input

When run on an Oculus Quest, all input is perfomed via the Oculus Quest’s Touch Controllers:

Afbeelding met tekst

Automatisch gegenereerde beschrijving

When running in VR, the controllers are visualised on their real-world location. UI like menu’s and flyouts can be attached to the controllers. When running in the editor, the controllers can be visualused optionally, located at a fixed offset in the current view. This is an aid for debugging the controller-attached UI.

|  |  |  |
| --- | --- | --- |
| Command | Touch | Keyboard |
| Next Project | A Button | Up arrow |
| Prev Project | B Button | Down arrow |

# Startup

Application starts up in Walkthrough in the following way:

* immersion mode ‘Walkthrough’ activated
* first Project (if available) activated
* first POI (if available) activated

# Immersion mode

Immersion mode can be toggled using the left index trigger.  
  
Immersion modes:

* Walkthrough mode
* Maquette mode

The application is said to ‘run in the active immersion mode’. The active immersion mode can be toggled between ‘walkthrough’ and ‘maquette’ mode.

Input

|  |  |  |
| --- | --- | --- |
| Command | Touch | Keyboard |
| Toggle immersion mode | Left index trigger | I |

## Walkthrough mode

The project is represented in real scale in Virtual Reality. The user can toggle between Points-Of-Interest included in the project.

Input

|  |  |  |
| --- | --- | --- |
| Command | Touch | Keyboard |
| Next POI | X Button | Left arrow |
| Prev POI | Y Button | Right arrow |

When switching to a new POI, the tracking space is ofsetted to counter-act the offset from the tracking center position. This assures that the user always lands on the intended POI position, thereby avoiding ‘awkward’ teleportations into walls, thin air, or above staircases. Downside is that the volume in which the userr can walk freely, can be malaligned tot he POI – leaving too much room for movement at one side, too little room of movement on the opposing side.

## Maquette mode

In maquette mode, the model is rescaled to 1/25 scale (the usual scale for physical architectural maquettes). When entering this mode, the tracking space is located at it’s default location (around the model. The model is scaled 1/25, and resides at a height of 1m 20 cm, with no rotation.

In maquette mode, the ‘maquette preview context’ scene is visible. It is a real-scale context in which the maquette is located, in order to make the process of previewing a model feel more natura lto the user(It does not feel natural to float in thin air without something to stand on). On top of that, some info is displayed in the preview context. The preview context consists of the following:

* A platform/room on which the model is centered.
* A 3D text couting ‘KS-architect’
* A 3D tekst couting the name of the project being previewed.

### Toggle model layer visibility

The user can toggle different parts of the model visible/invisible as follows:

* Touch the part with the left controller, so that it (or it’s outline/bbox) becomes highlighted.
* ‘Click’ the part with the left controller index button.

### Manipulate model location

The model location can be manipulated using the as follows:

* Left Thumb horizontal: Rotate model around up vector.
* Left Thumb vertical: Translate model along up vector.
* Left Thumb Click: Reset model to its default location (no rotation, y=1m 20cm)

# Movement

In any immersion mode, the position of the tracking space (and hence viewpoint) can be manipulated as follows:

* Move forward: Translate along ‘forward’ viewing direction.
* Move backward: Translate along ‘backward’ viewing direction.
* Move left: Translate along ‘left’ viewing vector
* Move right: Translate along ‘right’ viewing vector
* Move up: Translate along ‘up’ viewing vector
* Move up: Translate along ‘down’ viewing vector

In the editor, the inputs are as follows:

* Z -> Forward
* S ->Backward
* Q -> Left
* D ->Right
* R -> Up
* D ->Down

The view can be rotated by pressing LShift while moving the mouse.

In VR, the controls are mapped to he Right controller as follows:

* Thumb Up -> Forward
* Thumb Down ->Backward
* Thumb Left -> Left
* Thumb Right ->Right
* Index Trigger -> Up
* Hand Trigger ->Down

# How to prepare Sketchup for Unity import

Below the rules fors ketchup resources, in order to be importable as Unity assets:

* Material names:
  + Texture name == Material name
  + No spaces
  + -> Use Pascal Casing? (eg. VloerTegel512x512, Boom2D256x512) forxture/material names
  + Use only JPG? (or is png also fine?)
* Face orientation
  + Should be consistent and correct. (no reversed faces)
  + Tip: Use Edit/Face Style/Monochrome to check face orientation.

# How to prepare Sketchup for Archi-VR import

* + Use outliner to divide model into the entities that should be visiblility-togglable in Maquette mode. Eg:
    - Model
      * Context (Street, garden,…)
      * Outside (Gevel)
      * Roof
      * Basement2
      * Basement 1
      * Groundfloor
      * Floor1
      * Floor2
      * …

# How to create an ArchiVR project

Hierarchy:

* + - Project
      * Model
      * POI
      * Lighting
  + Copy-paste-rename an existing ArchiVR project
  + Open it
  + Remove the content of the ‘Model’ node
  + (create a new folder under assets/Projects/ProjectName to store the necessary assets fort he ArchiVR project in.
  + Import the assets
  + Drag in models under the ‘Model’ node.
  + Adjust the POI.
    - Add/Edit/Remove POI to match the project.
    - Tip: Create additional POI by copy/paste/rename existing POI. (start from an original on the same level)
    - Relocate the a POI by selecting it (not its children!), and dragging it to the correct position/orientation.
  + Adjust the lighting
    - Add/Edit/Remove lights to match the project.
    - Tip: Create additional Light by copy/paste/rename existing lights. (start from an original on the same level)