# 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.

Input

|  |  |  |
| --- | --- | --- |
| 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). The tracking space is located around the model. The model location can be manipulated as follows:

* Rotate around ‘up’ vector
* Translate up/down

The maquette can be reset to it’s default position (no rotation, y=1m 20cm) by clicking the right controller thumbstick.

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 natura lto 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.

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.

# Flythrough

In any immersion mode, the location of the tracking space 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