**TODO**

Bugfixes

* Only show button mapping ui when controller height is below certain treshold irt center eye anchor.
* Enable back-face culling.
* Fix FadeOut/FadeIn (flickers + depth-clipping)
* Maquette mode: Model should rotate around center of the building.
* Prevent awkward teleportations: When teleporting, make sure the user ends up in the POI (IE offset trackingspace, so that not center of tracking space is == POI, but location of head.) this will prevent the sporadic ‘awkward teleporting’, eg when after a teleport, the user ends up on top of a stair, inside a wall, outside the gevel in thin air, etc…)
* Project KS046:
* Model
* L0 Keuken dampkap texture missing
* L1 dubbel bed put puniform color on pillows
* L0 Wasplaats: Add laundry bin
* L0 Technische ruimte: Add: boiler, ketel, rek
* L1 Slaapkamer 2: Add: Buro stoel.
* L1 Slaapkamer 1, 3: Add: Nachttafeltje
* L0 Leefruimte: Zetel 1P lijkt te klein geschaald...
* POI
* L1 WC: in muur?

Design Defects

* Improve code related to controllerstate and ButtonMappingUI, in order to mitigate code verbosity/duplication.
* Implement ButtonState concept (Default/Down/Pressed/Up)
* Implement ‘Command’ class?
* § CommandMoveXY
* § CommandMoveXYZ
* § CommandResetViewerLocation (click r thumbstick)
* § CommandActivateNextPOI
* § CommandActivatePrevPOI
* § CommandActivateNextProject
* § CommandActivatePrevProject
* § CommandToggleImmersionMode
* § CommandTranslateModel
* § CommandRotateModel
* § CommandResetModelLocation (click l thumbstick)

Features

Buddy space

* Implement application stat 'budy space definition'
* Show camera feed
* Show Pickray
* Let user define 'shared buddy reference system':
* pick 3 points/markers (on ground level plane?) in different corners of the play area using pick ray
* Implement connecting to buddies using TCP-IP
* Implement broadcasting position (irt buddy shared reference system) to buddies.
* Implement avatar te represent users
* Implement drawing buddies on their current location using avatar.

Doors

* Implement Model Doors management, implement gathering doors from model.
* Implement picking doors to open/close them

Building lights

* Implement toggling building lights on/off

Environment

* Implement setting sun position: hour-of-day , sun azimuth
* Implement setting animation speed for time (sun position)
* Implement weather system
* (Fog, clouds, rain, snow...)
* Implement setting weather

FPS Counter

* Implement FPS counter
* HUD control (Simple white text to bottom of view.)
* Toggle ON/OFF how?

Pick ray

* Make PickRay prefab
* Instantiate prefab under right hand
* enable both L and R pickrays.
* Disable L pickray, use R pickray only
* ImmersionModeMauette: Use R index trigger to toggle model layer visibility.

Update documentation

* How to prepare a sketchup model for ArchiVR
* How to initialize an ArchiVR project
* Import SKP file as asset.
* Make sure a folder with the needed textures is present nevt to the model asset. Note: This is necessary for now because of a bug in Unity SKP imprter, that fails to find the textures embedded in the SKP file itself.
* Set 'Generate Lightmapping' UV's on newly imported SKP model asset.
* Set static flag to 'Contribute GI' where necessary (entire model except 'bovenkant muur' objects)
* Add box/mesh colliders to use for maquette-mode layer picking.

Furniture

* Implement option to toggle global furniture visibility.
* Implement a way to toggle visibility of furniture by picking?
* --> How to unhide???
* Implement custom(prettier) skyboxes

GFX settings

* Implement a menu to display and edit GFX settings
* Quality
* Reflection ?
* Ambient occlusion ?
* ... ?

Edit Mode

In Edit mode, user can edit the active project

* POI: create, edit, delete
* Furniture: create, edit, delete
* ...?

Will we even do this? Not usefull without making editable stuff transferable between builds which requires additional development… Only do when we have a workflow that needs it! For the time being, all project editing will be don in Unity editor, which suffices for now.

DONE

* Implement OVR driven input.
* Implemented Unity Input (buttons and kb).
* Fix texture load from SKP files. (No spaces in texture names allowed, material name must == texture name.)
* Implement loading projects from separate scenes asynchronously
* Implement aplication states:
* Default: While previewing the active project. Input is active. Model is visible
* Teleporting: While teleporting to another Project and / or POI. All input is blocked (in order to prevent requests to activate Project or POI),....
* First fade out
* Then show title of new project while async loading it
* Then fade in again
* Implement immersion modes: Walkthrough and Maquette
* Set menuMode default to ‘None’
* Fine-tune size and font for Controller-anchored menus.
* Walkthrough mode: Implement showing the name of active POI in RIght COntroller Menu.
* Implement showing projec name in Left controller Menu
* Implement fading (out, in) transition when teleporting (improves user confort).
* Implement pickray.
* Maquette Mode: Implement hiding model layers using picking.
* Unhide all model layers when exiting immersion mode ‘Maquette’
* Editor mode: Implement showing keys next to button action label
* Implement PickRay class
* Generate some bumpmap gevelsteen textures from DAAS, and add them to resources repo.
* Maquette immersion mode: Implement exclusive rotate/translate locking mechanism: While rotating, no translation can be performed and vice-versa.
* ButtomMappingUI: Implement deadzone for thumbsticks.
* Implement dynamic pickray color:
* If no hit: Use textured cylinder with fading out (fully opaque -> fully transparent) uniform white color
* If hit: Change to red color.
* Implement getting build datetimestamp for composing build version number automatically.
* Fix HUD menu not working
* ImmersionModeWalkthrough: Implement a way to align model with tracking space:
* LT left and right: rotate model around center eye position.
* RT up, down, left and right: translate model (relative to viewer)