TODO

* Implement getting build datetimestamp for composing build versio number automatically.
* Implement 'buddy space'
* Implement 'budy space definition' mode
* Show camera feed
* Show Pickray
* Let user define reference system:
* pick 3 points/markers in corners of the play area using pick ray
* Implement avatar te represent users
* Implement connecting to buddies, broadcasting of position (irt buddy shared reference) to buddies, drawing buddies on their current location.
* Implement picking doors to open/close them
* Implement toggling building lights on/off
* 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
* Implement toggling to guardian edit mode On/Off(default) using LTC (Left Thumb Click)
* Guardian is visible
* Controls:
* LT left and right: rotate model
* RT up, down, left and right: translate model (relative to viewer)
* Implement FPS counter
* HUD control (Simple white text to bottom of view.)
* Toggle ON/OFF how?
* Fix HUD menu not working
* Fix FadeOut/FadeIn (flickers + depth-clipping)
* Make PickRay prefab
* Instantiate prefab under right hand
* enable both L and R pickrays.
* Implement option to toggle furniture visibility
* Refine Project Demo2:
* Relocate model so that origin is at center of building
* Refine Project KS046:
* Add bumpmap gevelsteen texture
* Relocate model so that origin is at center of building
* Properly divide model into layers:
* Omgeving
* Dak
* Gelijkvloers
* Verdieping
* 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.
* Implement a way to toggle furniture visibility
* Improve code related to ButtonMappingUI, in order to mitigate code verbosity/duplication.
* o Implement ‘Command’ class?
* § CommandMoveXY
* § CommandMoveXYZ
* § CommandResetViewerLocation (click r thumbstick)
* § CommandActivateNextPOI
* § CommandActivatePrevPOI
* § CommandActivateNextProject
* § CommandActivatePrevProject
* § CommandToggleImmersionMode
* § CommandTranslateModel
* § CommandRotateModel
* § CommandResetModelLocation (click l thumbstick)
* · Fix sensitivity for l thumbstick when manipulating model
* · Fix sensitivity for thumbsticks for highlighting button UI. (set deabZone to 0.5?)
* P046
* POI fixes
* Terras POI: too low
* POI rechtergevel: verplaats naar hoekje van terras, en fix hoogte.
* POI leefruimte: plaats in midden van leefruimte.
* Create POI:
* “VooraanRechts”
* WC?
* SKP fixes
* Texture/material names
* Add WC pot
* Keuken:
* Add cooking plates
* Add dampkap
* Aanrecht
* Badkamer:
* Add kraantjes, douchekop
* Berging:
* Add shelf
* Fix height of zetel zwart 1P in leefruimte: te hoog.
* Fix orientatie van kastdeur in slaapkamer.
* Implement aplication states:
* Previewing While previewing a project. Input is active. Model is visible
* Loading: While (un)loading a project to preview. 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 toggling lights on/off
* Implement custom(prettier) skyboxes
* Tweak unity GFX settings
* Reflection
* Ambient occlusion
* ???
* Implement Edit mode:
* If active, user can define, edit, save POI for a given project.
* Will we do this? Not usefull without making POI transferable between builds which requires additional development… Only do when we have a workflow that needs it! For now prepare projects in Unity editor, which is fine.
* 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…)
* Implement OVR driven input.

DONE

* Implemented Unity Input (buttons and kb).
* Fix texture load from SKP files.
* Implement loading projects from separate scenes asynchronously
* 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
* 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.