TODO

* Implement showing keys next to button action label
* Implement option to toggle furniture visibility
* 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.
* Refine Project Demo2:
* Relocate model so that origin is at center of building
* Generate some bumpmap gevelsteen textures from DAAS, and add them to resources repo
* 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
* 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.
* Maquette mode: Implement showing/hiding parts of the model
* Implement help menu (HUD menu, attached either to front of head or controller(s)) explaining controls.
* Walkthrough mode: Implement showing the name of active POI
* World-space Tekst GameObject
* lokated in fixed location in front of user, or
* attached to front of head?
* 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.
* Unhide all model layers when exiting immersion mode ‘Maquette’
* Implement pickray class