Guide for the Future

for World Builder VR

This guide contains description and content about possible directions the app can pursue. None of these are of course set in stone, as the direction is ultimately YOUR decision. These are just some ideas to help you get started and NONE of these should be taken as ultimatums.

Obsah

[1. Model Importer 3](#_Toc90751015)

[1. Current Architecture 3](#_Toc90751016)

[2. What’s Needed 3](#_Toc90751017)

[2. Stage Selection/Import 4](#_Toc90751018)

[1. Current Architecture 4](#_Toc90751019)

[2. What’s Needed 4](#_Toc90751020)

[3. Model Creator 5](#_Toc90751021)

[1. Current Architecture 5](#_Toc90751022)

[2. What’s needed 5](#_Toc90751023)

[4. Interactivity Tool 6](#_Toc90751024)

[1. Current Architecture 6](#_Toc90751025)

[2. What’s Needed 6](#_Toc90751026)

[5. VR Keyboard 7](#_Toc90751027)

[3. Current Architecture 7](#_Toc90751028)

[4. What’s needed 7](#_Toc90751029)

# Model Importer

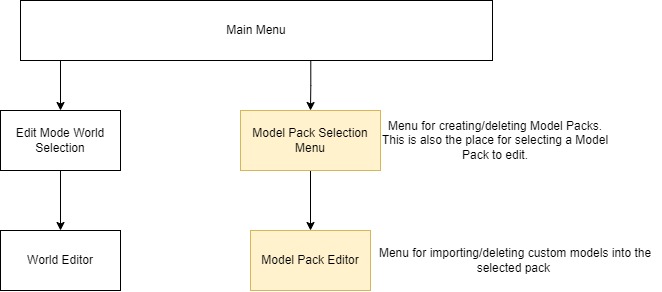
Right now, if you want to add more models into the app, it must be done by the developer in the Unity Engine itself. There is no way for the user to add any of their own content. The idea here is to add a separate Model Importer Editor into the Main Menu of the app.

## Current Architecture

Currently all the game’s models are stored in one model pack, where both the models and the pack are of type ScriptableObjects. The app loads model packs at startup via the ModelPackLibraryLoader.

## What’s Needed

To implement this feature, there is a need for a menu where the user can create/delete their own model packs. These packs can be than opened into a separate menu, where the user will be able to import external model files (.obj, .fbx, etc.) into the currently edited pack. Besides just importing models into this app, they should also be adjusted to work with the app itself by giving them the same structure as the already existing Model Prefabs.



To achieve more logical in-app structure, the models and model packs should be reworked from the current ScriptableObjects structure into the same structure, that’s used by WorldAssets.

# Stage Selection/Import

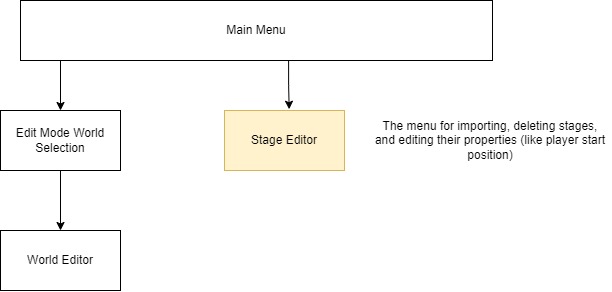
The same thing that was described for models in the last section also applies for stages. Stages are just World Models, the places you place the models into. The app would greatly benefit from having the user import their own stages and build on them.

## Current Architecture

Currently the app has only 1 stage (default house). Stages are not stored in packs like models are, but they use the same architecture as models (ScriptableObjects). This again is not good for in-app importing and needs the dev and Unity Engine to be done.

## What’s Needed

StageAssets need to switch architectures from using the ScriptableObjects to the same one as the WorldAssets utilize. A system for importing 3D models also needs to be in-place. Then the app needs a special menu for importing, deleting stages, and editing their properties.



# Model Creator

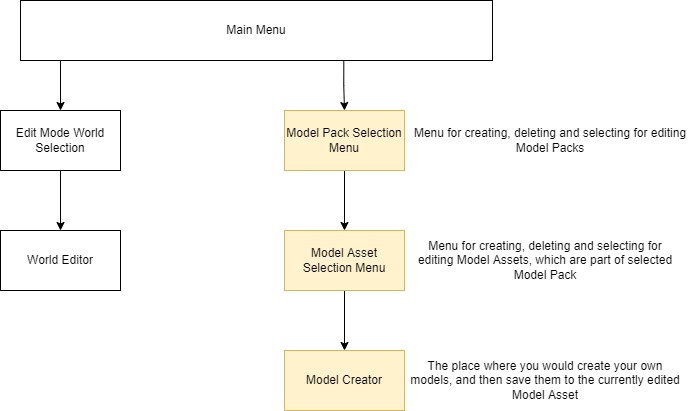
Modeling in VR has a lot of benefits, as well as, having everything you need in one app. It would be very beneficial to model all your models and stages in the app itself and not have to deal with any importing at all!

## Current Architecture

Currently there is no architecture to support this feature. The only think is what would be the end point of the model creation, which is the ModelAsset containing the model GameObject.

## What’s needed

Firstly, it would be necessary to transfer the Model Architecture from using ScriptableObjects to using the same architecture as WorldAssets. Next would be the menus for setting things up. A menu for creating and deleting model packs. Also the menu where you would select a model pack to edit. Another menu, where you would create, delete, and select to edit ModelAssets. And lastly the Model Creator Editor, where you would be able to create your own models.



# Interactivity Tool

Right now, all the models are all, but interactive. The only thing you can do with them is to place them and remove them. Adding some way to interact with them would greatly increase the fun of the app (and possibilities). What makes the most sense is an Interactivity Tool. (There can be times the user WANTS to have a flying, non-interactive object.)

## Current Architecture

Currently, the app has a Tool System in-place, so the Interactivity tool would make to be added as one of the selectable tools.

## What’s Needed

Creating a new kind of tool, that can select place objects and upon selection, something like a properties panel shows up next to it. In this panel the user could select, if they want the object to have things like gravity, being pickable, etc.

# VR Keyboard

In the app, there is currently no way to input any text of any kind and thus navigation between player created worlds is not best. The solution to this would be to add a keyboard the user could use while in VR. (Special focus on it being usable in VR.)

## Current Architecture

Currently the app doesn’t support anything for this feature, besides setting names, descriptions, and authors to WorldAssets.

## What’s needed

The VR keyboard system, that would then be called, whenever an asset/app would require a string input. As this a separate system, not much tampering with core features would be needed.