Spike

AR Mapping

This spike aims to give an overview of how AR Mapping is handled in the project.

Definitions

A-space

An imaginary 3D space where a model is rendered during “standard” rendering; not in AR. The axes defining the directions of this space is relative.

B-space

The 3D space used to render a model in AR and is constructed when ARCore starts tracking. Therefore, this space’s directions are described by the real world; when the rendering activity is started, the x-axis is to the right of the phone (relative to the real world), the y-axis is always straight up (relative to the real world), and the z-axis is the normal-vector of the x- and y-axis.

Trackables

A class made for containing the data about trackable world planes, plane-intersections/lines, and line-intersections/corners. The class receives information about surfaces such as walls, floors, and roofs from ARCora and calculates the trackable information mentioned above. The Trackables describes the B-space.

Space-geometry

Data describing the geometry of a room. This data is provided by Rendra via their API, and the space-geometry of a specific room can be found by using the camera position when in that room. Most usefully, the space-geometry contains a room’s vertices, which are the same as the Trackables line-intersections/corners.

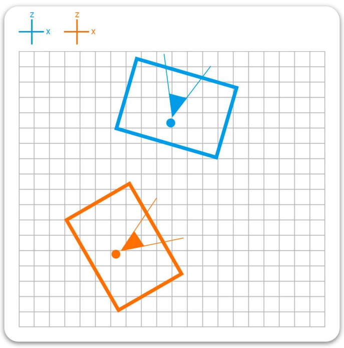
Description

AR Mapping is the process of mapping a model from A-space to B-space to overlay it correctly over the real world. This requires the app to find a transformation with a rotation and translation that can be applied to the model’s root-node.

Implementation Overview

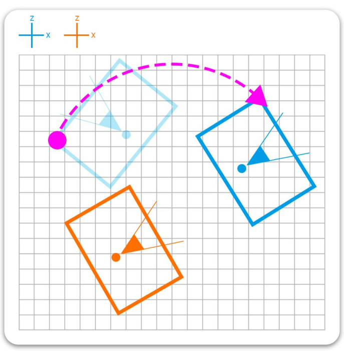
We can apply A-space to any space, as it is relative. it can, therefore, be described as the same space as B-space. This solution works because models are defined by the A-space.

A-Space B-Space



1 | Initial State

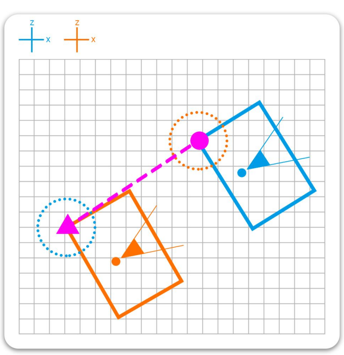
Initial example state



2 | Finding Rotation Using Rotational Difference

Rotate the A-space model by the rotation between the directional vectors of the planes the cameras are looking at.

The planes' directional vectors should be used because of the cameras' deviating rotation caused by imprecise user interaction.



3 | Finding Translation

Find the corresponding point in the model and space-geometry, and find the vector from P(A-space) to P(B-Space).

Implementation Details

1 | Initial State
Initial example state

2 | Finding Rotation Using Rotational Difference
Rotate the A-space model by the rotation between the directional vectors of the planes the cameras are looking at.

The planes' directional vectors should be used because of the cameras' deviating rotation caused by imprecise user interaction.

3 | Finding Translation
Find the corresponding point in the model and space-geometry, and find the vector from P(A-space) to P(B-Space).

This could easily be achieved by using the planes' definition in step 2, which is defined by a point and a directional vector,