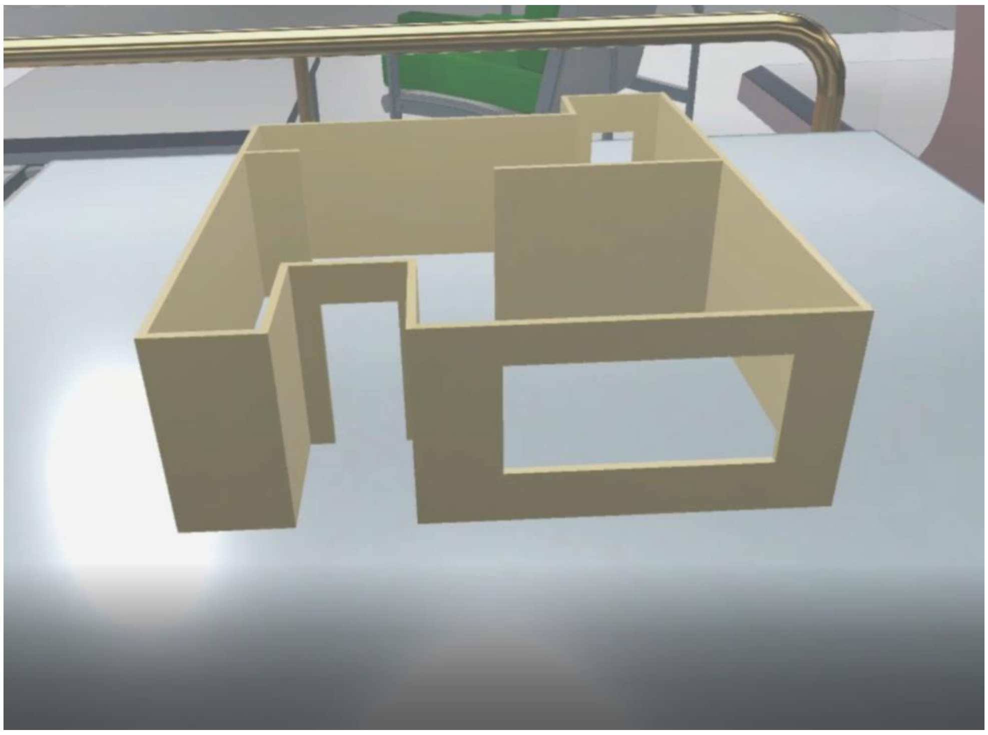


VR Floor Planner

Made by:	Netanel Lev	&	Dolev Ben Ami
Supervisors:	Boaz Sternfeld	&	Yaron Honen

September 2018

Table of contents

Abstract

The emergence of virtual reality (VR) has been one of the big stories of the past few years in the tech world. We believe that in the future, VR will become an integral part not just of presenting a project, but of the design process as well.

For many design-led industries, the biggest challenge is often convincing the client that the finished product will look just like (or better than) the 2D representation.

No matter how talented the designer, it can take a leap of faith and a vivid imagination from the client to get them on board with, and excited by, a design idea. Architecture is no different and that’s why VR for architecture and design could help transform this industry.

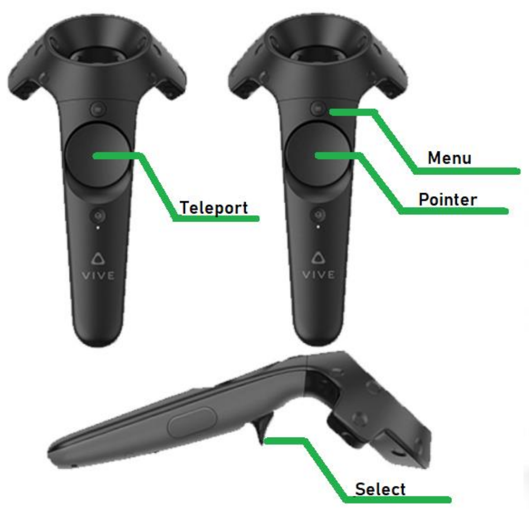
In this project, we look at the possibilities attached to this new technology, the benefits to designing in VR (for both the designer and the client), and how we expect the industry to grow and evolve as VR is accepted and implemented.

Currently there are many 2D editing applications designed for architects. Many of them are not easy to use. It was important for us that the application will be user friendly.

The results far exceeded our expectations. We received very positive feedback from friends and staff that used the application. We conclude that incorporating VR in the architecture industry seems very promising.

System

The app was developed for the HTC VIVE platform, which consists of a headset, controllers and 2 base stations. They need to be configured properly (see links below) before the app can be used.



The project was developed using:

1. Unity 2018 – the game engine that runs our app.
2. C# - primary programming language of the project.
3. Visual studio 2017 – IDE used for developing C# scripts.
4. SteamVR client – has to be installed for the HTC VIVE to operate.

For more information please refer to:

https://unity.com/

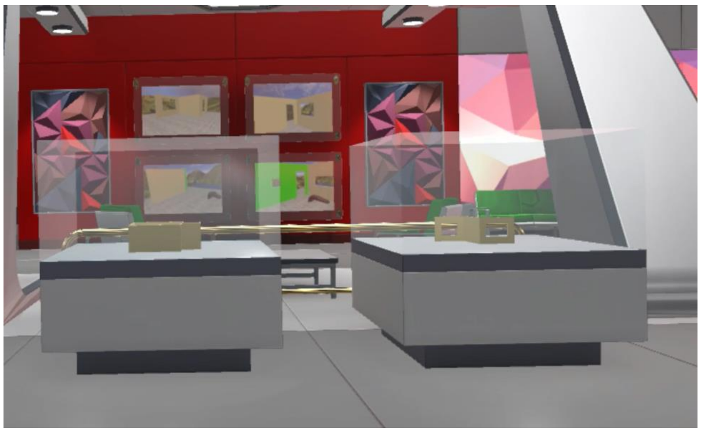
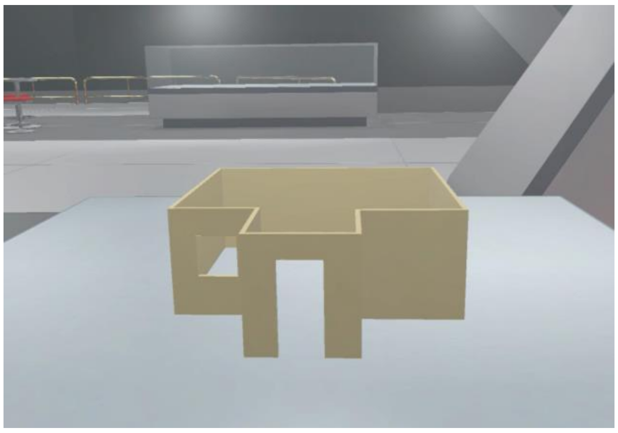
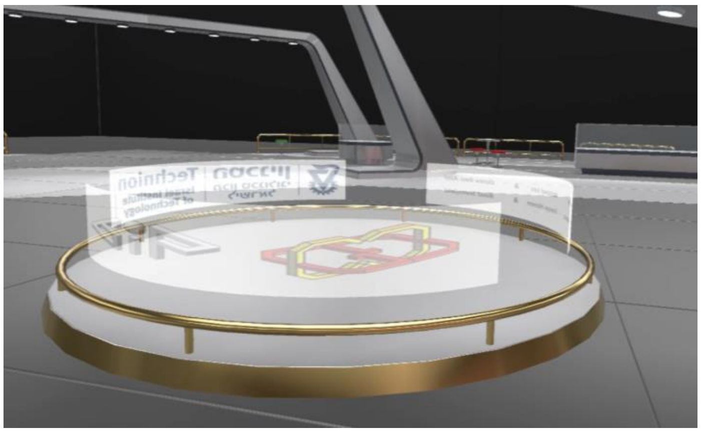
https://www.vive.com/us/

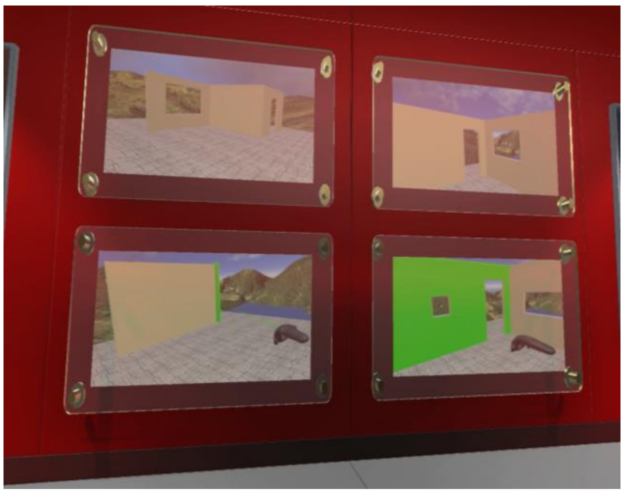
Application overview

The application consists of 2 scenes - start and edit.

The start scene

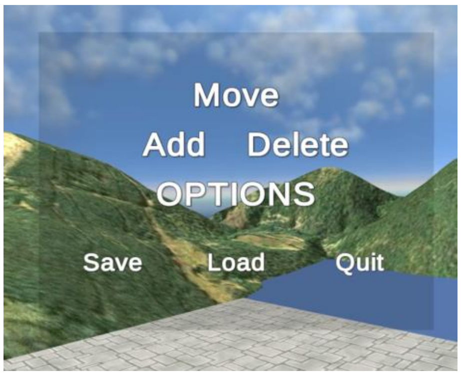
The start scene is a show room that displays all the saved models as miniatures on stands. This show room was inspired by an architect's office, which allows the clients to experience the product. Clicking on a miniature will switch to edit mode and load this model.

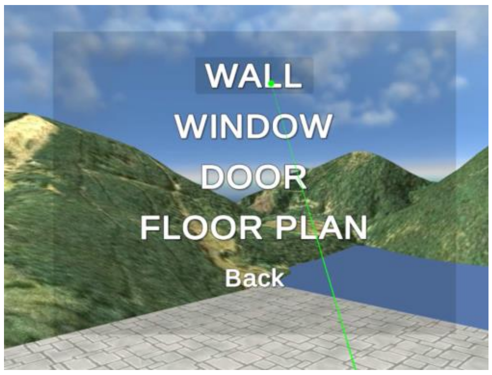
המעבדה לעיבוד גיאומטרי של תמונות
Geometric Image Processing Laboratory
Center for Graphics and Geometric Computing
VR Floor Planner
Made by: Netanel Lev & Dolev Ben Ami
Supervisors: Yaron Honen & Boaz Sternfeld


The edit scene

The user is located inside a full scale model. He can move around and edit the model: Create, delete and adjust walls, windows and doors.

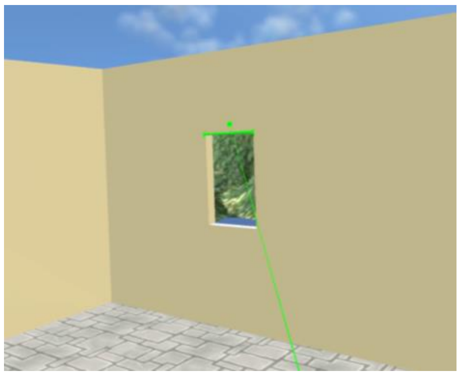


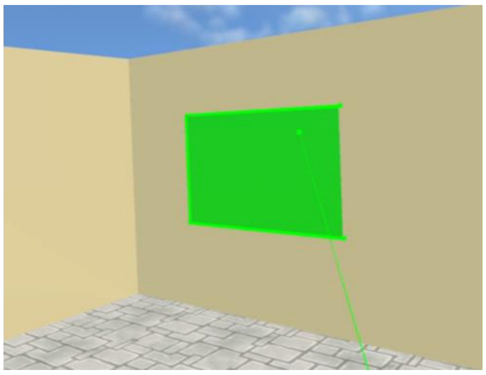
Main menu



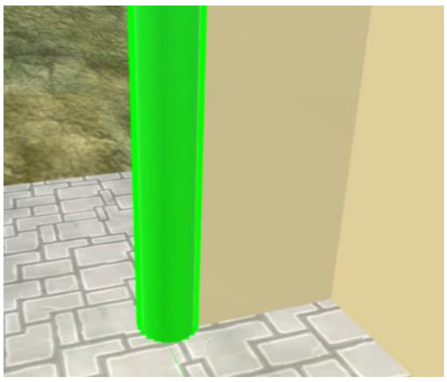
Add menu

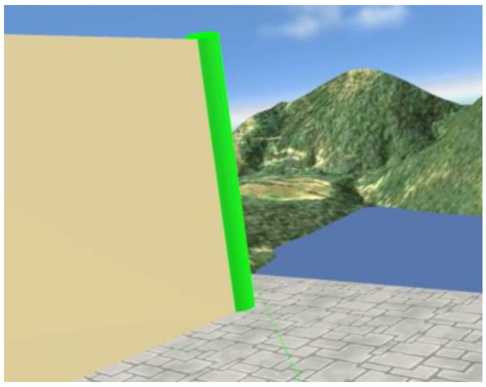
Drag and resize operations are united into a mode called “move”. When the user points at a section it is highlighted.


Resize by grabbing edge


Drag by grabbing middle

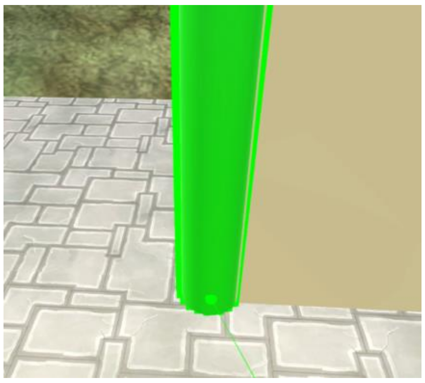
A wall edge is highlighted during interaction (resizing or creating a new wall from that edge).

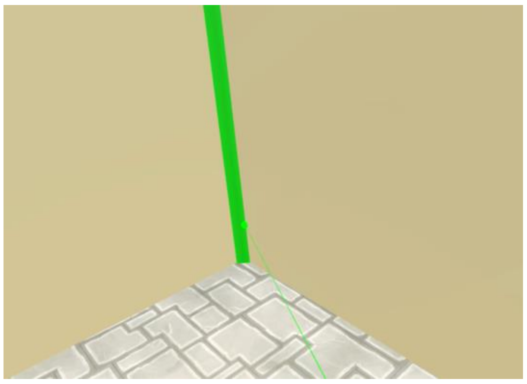




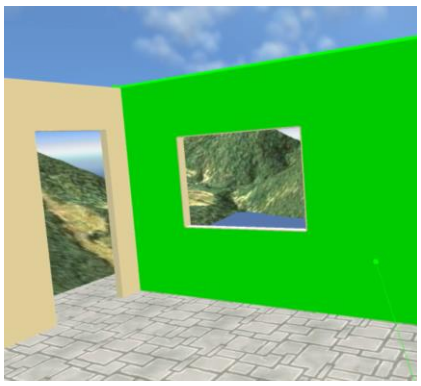
Auto wall attachment

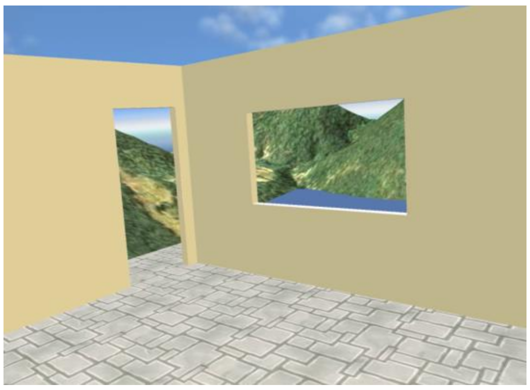
This is a collection of special features that help the user to create a flawless floor plan, without wall gaps or overlaps. When grabbing a wall edge and placing it near another wall edge, it will be automatically attached to the other edge, to form a perfect corner.


Grabbing a wall's edge


Attached to another edge

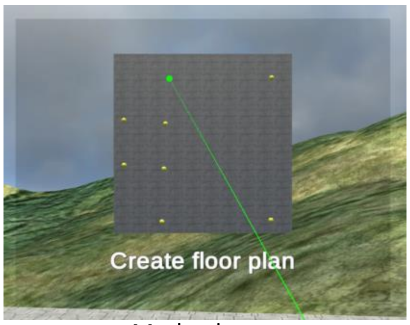
When moving an entire wall and it's edge touches another wall, it will be attached to the axis of the stationary wall. This prevents gaps or overlays between them.


Moving entire wall

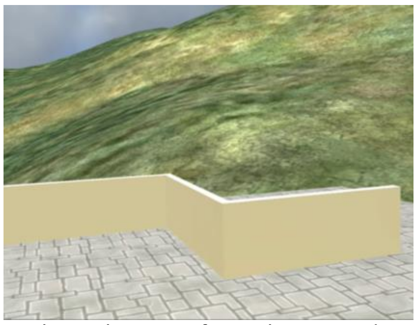

Attached to another wall

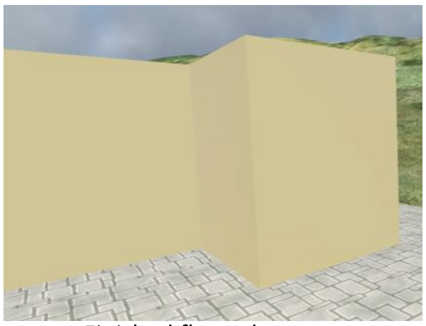
2D Floor Plan

The user can choose to create the initial skeleton using a 2D interface. The user marks the corners, and then the floor plan rises from the ground.

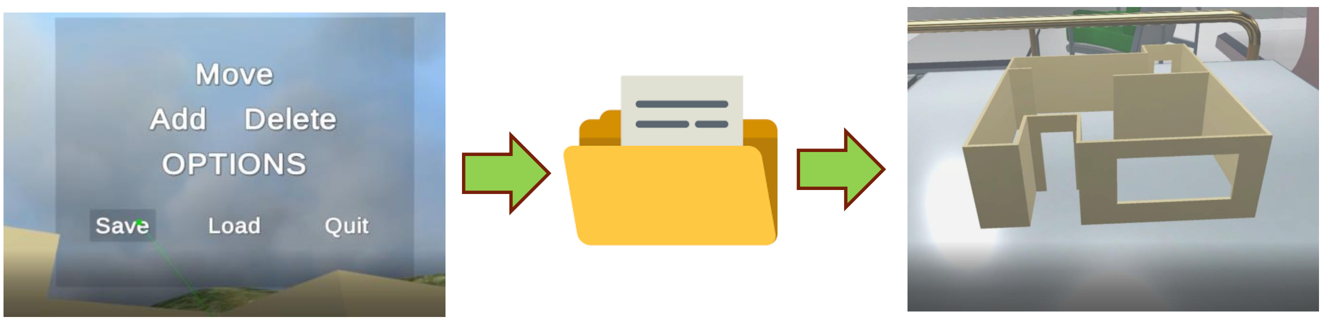


Mark edges


Floor plan rises from the ground


Finished floor plan

Save & load mechanism



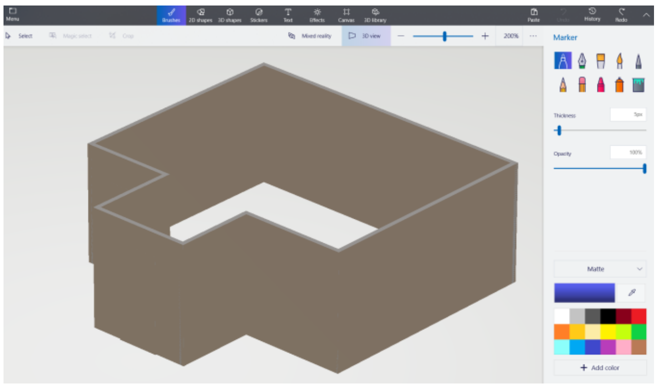
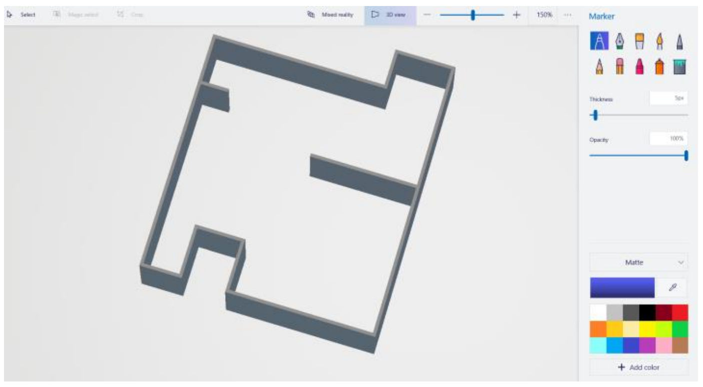
The models are saved in binary files using serialization.
To present the miniatures (in start scene) all the models are loaded in a small scale.

Export to obj

The user can export his model to obj file. This enables the user to

- Share his design with others
- 3D print his design
- Continue to edit in other programs

These models were exported to obj and then opened in 3D paint:

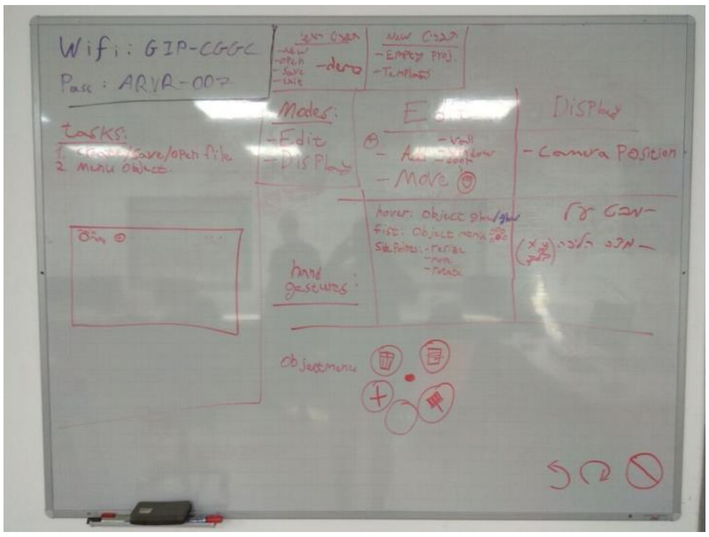



Development process

Design

We created a specific description of:

- Features
- Scenes
- Menus
- Controller operation
- Modules



Development methodology

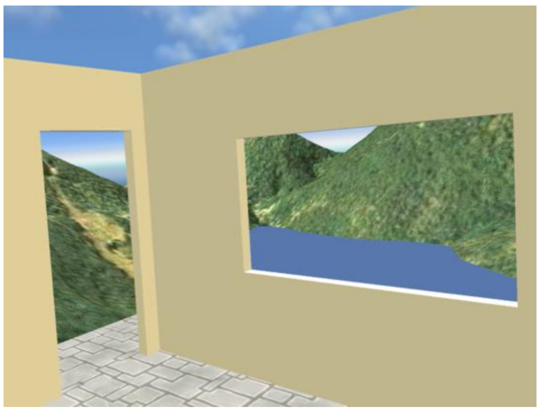
The development process was dynamic, feature driven and very efficient. We were inspired by Agile software development. It was comprised of very short and intense iterations (about 2 full working days). Each iteration had specific goals.

This allowed the development process to be adaptive and iterative. When the needs of the project changed, we reacted fast and adjusted accordingly.

Quality assurance has to be done manually, as there is no automated testing infrastructure. This may cause some challenges in debugging a graphic applications.
We anticipated that and thanks to the development methodology we caught bugs very early, and were able to fix them with relative ease.

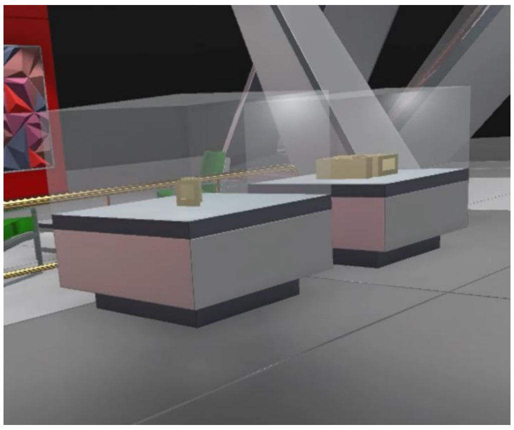
Windows & doors

The most straight forward approach to adding a window or a door, is to destroy the wall and create a new wall which is made of smaller pieces. This gets complicated if you allow more than one object per wall. We solved this by using a special shader for doors, windows and walls. It was a simple and fast solution the problem.



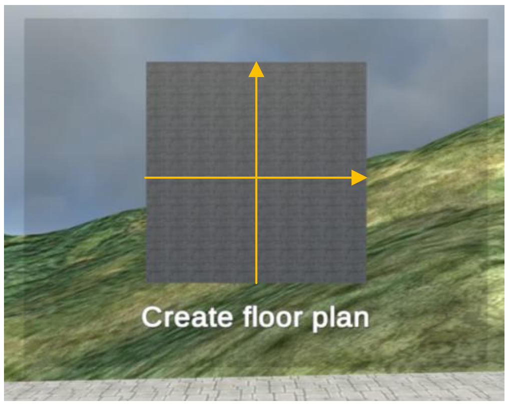
Save & Load mechanism

The models are saved in binary files using serialization. The save algorithm iterates over all game objects and uses a serializable class to save relevant information (position, rotation, scale…).
To present the miniatures (in start scene) all the models are loaded in a small scale.



2D Floor Plan algorithm

A coordinate system was placed on the menu. Each marked corner had a an (x,z) values on the menu coordinate system, which was translated to the model coordinate system. The algorithm auto aligns the walls, so the corners will be perfect and the walls will have no gaps between them, even if the user makes a mistake when marking corners.



Conclusions

The results far exceeded our expectations. We received very positive feedback from friends and staff that used the application. Incorporating VR in the architecture industry seems very promising.

At the beginning of the project, we intended to use the Manus VR gloves to interact with the app. As we began implementation, we discovered that the gloves don't support some of the most important gestures we need, such as pinching fingers to grab a wall edge.
This made us redesign the entire user interface and how to interact with objects. The gloves need to be improved to unleash their full potential.

Designing in VR is intuitive and feels natural, and the possibilities are unlimited. The user base is growing very fast. At this point of time, there aren't many tools for designing and developing in VR. This creates a rare opportunity for commercial use.

Special thanks

We want to thank Boaz Sternfeld, Yaron Honen, GIP and CGGC laboratories for the help and resources that were invested to make this project come true.