

UNIVERSITY OF TARTU  
Institute of Computer Science  
Computer Science Curriculum

Ott Saar

# **RoomMapperAR A Mobile Augmented Reality Room Mapper**

**Bachelor's Thesis (9 ECTS)**

Supervisor: Raimond-Hendrik Tunnel, MSc

Tartu 2019

# **RoomMapperAR A Mobile Augmented Reality Room Mapper**

## **Abstract:**

This thesis describes the development and testing of a room mapping application RoomMapperAR. The available room mapping applications are being analysed and the common features are implemented and improved upon. To determine the usability of the application numerous tests are conducted. The testing results will be used for further development until the application is ready to be published into the Google Play store as a free application.

## **Keywords:**

Computer graphics, Augmented Reality, Android, Unity, ARCore, UX design, mobile application, software development

**CERCS:** P170 Computer science, numerical analysis, systems, control

## **RoomMapperAR – mobiilne ruumikaardistamise rakendus kasutades liitreaalsust**

### **Lühikokkuvõte:**

Antud bakalaureusetöös kirjeldatakse ruumikaardistamis rakenduse arendust ja testimist. Olemasolevaid ruumikaardistamise rakendusi analüüsitakse ja levinumad tunnused implementeeritakse ja arendatakse edasi. Et veenduda rakenduse kasutatavuses viiakse läbi testid. Testimisetulemuste põhjal arendatakse rakendust edasi kuniks rakenduse valmimiseni, siis laetakse see tasuta rakendusena Google Play poodi.

### **Võtmesõnad:**

Arvutigraafika, Liitreaalsus, Android, Unity, ARCore, UX disain, mobiilirakendus, tarkvaraarendus.

**CERCS:** P170 Arvutiteadus, arvutusmeetodid, süsteemid, juhtimine (automaatjuhtimisteooria)

## Table of Contents

1	Introduction .....	5
2	Augmented Reality.....	6
2.1	Techniques.....	6
2.1.1	Markerless AR .....	6
2.1.2	Marker-based AR .....	8
2.1.3	Projection-based AR .....	9
2.1.4	Superimposition-based AR .....	9
3	Similar Applications.....	11
3.1	Magicplan (Android, IOS).....	11
3.2	RoomScan Pro (IOS).....	12
3.3	PLNAR (IOS).....	13
4	RoomMapperAR .....	15
4.1	Creating the Walls .....	15
4.2	Marking the Doors and Windows.....	16
4.3	Room Editing.....	19
4.4	Furniture Placement.....	20
4.5	Exporting the Object.....	21
5	Testing.....	22
5.1	Devices/Platform Testing .....	22
5.2	Usability .....	22
5.2.1	Room Creation .....	23
5.2.2	Editing Objects.....	25
5.2.3	Graphical User Interface .....	26
5.2.4	Comparison to MagicPlan.....	27
5.2.5	Additional Suggestions .....	28
5.3	Result Comparison .....	29
5.4	Future Improvements.....	31
6	Conclusion.....	32
7	References .....	33
	Appendix.....	34
I.	Glossary.....	34
II.	Accompanying Files.....	35
III.	Source Code.....	36

IV.	OBJ Example.....	37
V.	License.....	40

# 1 Introduction

Room mapping and floor plan creation are important parts of visualising and planning homes. Usually the creation of these room layouts require physical measuring tools and dedicated software where the found results can be marked down. However, with the growth and performance increase of mobile devices, the process of mapping rooms with only a smartphone is becoming more widespread. Using augmented reality technology, the user can measure the walls and mark them down without any need of physical measuring tools <sup>1</sup>.

The goal of this Bachelor's thesis is to create a room mapping application using augmented reality technology. Dedicated augmented reality room mapping applications can be found, but available applications lack any direct interactivity with the furniture, the wall height variable is usually not taken into account and 3D export of the result is usually limited. The created application should be helpful for simple home owners for creating a 3D-representation of their home. This would assist in designing the layout of their rooms and the placement of the furniture in it. The available tools in the application should make the designing process as easy as possible and not require any previous knowledge in room mapping. The application was tested on different devices and environments to ensure its compatibility and quality. The usability of the application was tested on people with no prior room mapping knowledge.

The first chapter of this thesis introduces the reader to the domain of augmented reality and what the technology is capable of at its current state. The second chapter describes the highest rated available room mapping applications on the market and brings out their strengths and weaknesses. The third chapter describes how augmented reality was implemented in the created RoomMapperAR application and the overall description of the added features. The fourth chapter gives an overview of the results in quality assurance and usability tests, problems found during testing and the fixes that were made.

---

<sup>1</sup> <https://www.popularmechanics.com/home/tools/a15886848/plnar-app-measure-map-rooms/>

## 2 Augmented Reality

Augmented reality (AR) is defined as a view of the physical world that is enhanced with information rendered by the computer [1]. This information can be text, audio, video or a 3D object that does not exist in the real world. Instead it is rendered after to give the illusion of its existence. Different from virtual reality (VR), where the whole world is virtual, augmented reality tries to combine both the virtual and the real world into one [2].

In most cases, AR systems need to have accurate information about the location and the orientation of the device. To implement an immersive AR experience, information in six degrees of freedom<sup>2</sup> is needed [2]. Six degrees of freedom consists of the X location, the Y location, the Z location, the yaw, the pitch and the roll.

However not all AR systems need accurate positional information. Depending on the purpose and the AR experience the application wants to make, different techniques can be used.

### 2.1 Techniques

There are four primary techniques used in the creation of an AR experience: (1) Markerless AR, (2) Marker-Based AR, (3) Projection-based AR and (4) Superimposition-based AR [3]. AR systems are not bound to only one technique, the use of a specific techniques depends on the purpose of the application and the hardware being used [2].

#### 2.1.1 Markerless AR

Markerless (aka position-based) AR uses device's sensor data to estimate the device's position [3]. Depending on the specific application, the methods and sensors used in determining the position can differ. The GPS sensor is used to determine the absolute world coordinates and is usually used in different GPS-based applications where the location of the device is crucial [2]. Image processing is often used to create feature points from the visual input. Feature points are visually distinctable points in the world, which the device uses to orient itself. This technique is used in cases where the physical position is not needed, and it is enough to know the position of the device itself in relation to the created feature point [1]. These feature points can be corners of the furniture or recognizable patterns on the ground. Feature points are most commonly used in the detection of the ground plane by processing the texture of the floor and creating a grid of

---

<sup>2</sup> [https://en.wikipedia.org/wiki/Six\\_degrees\\_of\\_freedom](https://en.wikipedia.org/wiki/Six_degrees_of_freedom)

from the found feature points. The common problem with feature point detection is that these points need to be well distinguishable. For example, detecting a unicolor wall is problematic since the differences in the texture are unnoticeable, thus no feature points can be detected <sup>3</sup>. The same principle applies to ground plane detection as well. The inertial (IMU) sensors like the accelerometer and the gyroscope are used to accurately detect the transformation changes of the device [2].

An example application using markerless AR is AR City <sup>4</sup>. AR City detects the area the device is in using GPS sensors. Since GPS alone is not that accurate <sup>5</sup>, computer vision is used to find the road or possible known points of interests in the camera feed. The visually detected ground plane helps to orient the device. The data from the IMU (Inertial Measurement Unit) sensors is used to compensate for the movement of the device and moves the shown augmented information accordingly (Figure 1).



Figure 1. AR City showing the route to the destination on the road

Markerless AR is a good technique to use, if the AR experience should only relay on the information that the device receives. For this reason, many mobile AR applications use this technique<sup>6</sup>, since for implementation only the device's sensors are needed. However, problems rise when using the image processing in an environment that does not have any distinguishable points.

---

<sup>3</sup> <https://blog.griddynamics.com/how-arkit-and-arcore-recognize-vertical-planes/>

<sup>4</sup> <https://www.blippar.com/blog/2017/11/06/welcome-ar-city-future-maps-and-navigation>

<sup>5</sup> <https://blog.solidsignal.com/tutorials/just-how-accurate-is-gps-exactly/>

<sup>6</sup> <https://www.makeuseof.com/tag/best-augmented-reality-apps-android/>

### 2.1.2 Marker-based AR

Marker-based AR is an AR realization technique, where the device recognizes a predefined symbol or picture [3]. The symbol, also called a *marker*, is taken as a reference and a virtual object is placed in relation to it. A

good marker object is visually easily identified. It is said that a pattern with four corners/points is the most easily and accurately recognisable shape in a changing environment [1]. For this reason, most of the found markers are black squares containing a code for identification (Figure 2) [4].



Figure 2. Examples of possible marker objects<sup>7</sup>

In the year 2019 marker-based AR was used to promote “Eesti Laul”, Estonia’s song competition and its songs<sup>8</sup>. It used “Tere” milk packages as the reference object and placed the artists in front of them (Figure 3).



Figure 3. Artist in front of the „Tere“ milk package<sup>9</sup>

This technique is often used in interactive books and item showcases where the shown symbol is known to the device [3]. The tracking of the objects is usually not a problem as long as the marker is seen by the device. However, without the detectable marker the made application is useless.

<sup>7</sup> <https://www.prototechsolutions.com/augmented-reality-development/>

<sup>8</sup> <https://digi.geenius.ee/rubriik/uudis/liitreaalsus-pani-eestis-inimesi-34-rohkem-piima-ostma/>

<sup>9</sup> <https://www.am.ee/node/6803>



### 2.1.3 Projection-based AR

Projection-based AR uses artificial lighting, which is projected onto the real-world objects, changing the shown information depending on user input [3].

An example of this is ARSandbox <sup>10</sup>. ARSandbox is a hands-on science exhibit that teaches about different geographic, geologic, and hydrologic concepts such as reading a topography map, forming of watersheds and the meaning of contour lines. Using depth sensors, the application reads in the height of the sand in different parts. The received information is processed and turned into a heightmap which gets projected onto the sand using an overhead video projector (Figure 4).

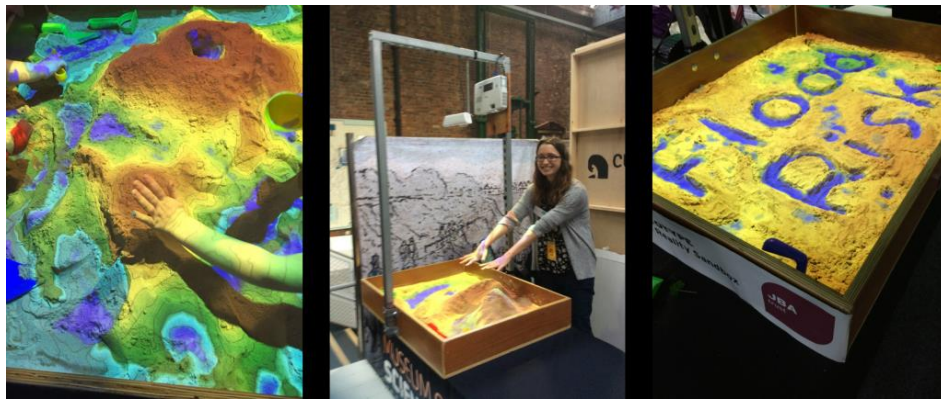


Figure 4. Picture of ARSandBox and the simple setup of it<sup>11</sup>

Different from the previously described techniques, this one is not that mainstream. Usually the implementation of the technique requires special sensors for the input and a way to project the information, making it not suitable for mobile use.

### 2.1.4 Superimposition-based AR

Superimposition-based AR either partially or fully replaces a physical world object with a virtual one [3]. In this kind of augmentation, the recognition of the object is vital, because the replacement of the object cannot take place unless the object from the original view is recognised. This way of augmenting reality is not widely used, however there are some studies to use it in the medical field. Some surgeries have been performed where the

---

<sup>10</sup> <https://arsandbox.ucdavis.edu/>

<sup>11</sup> <https://www.jbatrust.org/how-we-help/physical-models/ar-sandbox/>

surgeon is using pre-taken X-ray images and real-time AR view without receiving any X-ray exposure <sup>12</sup> (Figure 5).

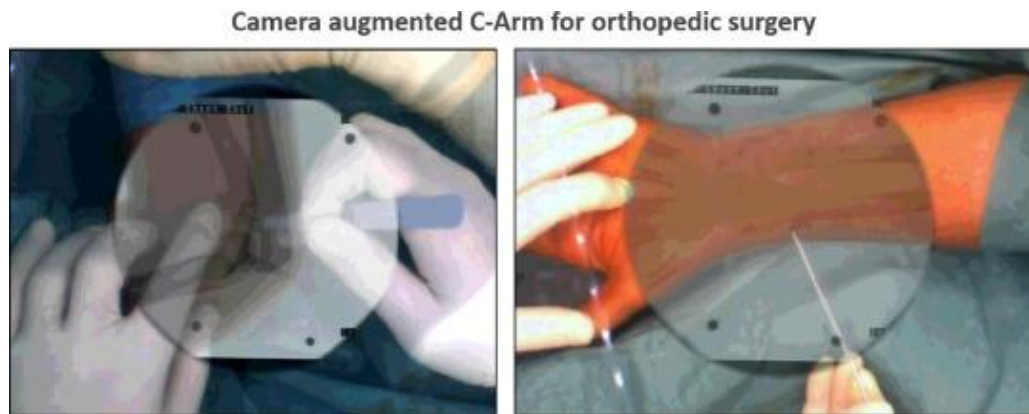


Figure 5. Surgeon using AR combined with previously taken x-ray image

As of now this technique is quite in the early development, meaning only a few specialized cases can be seen.

Out of all the described techniques, markerless AR requires the least amount of time for setup. For this reason, most of the room mapping applications use it. Same reason makes it good for simple users, since it does not require any previous augmented reality experience to use.

---

<sup>12</sup> <https://www.sciencedirect.com/science/article/pii/S1361841516301013>

### 3 Similar Applications

To determine what the common features of the room mapping applications are, it is necessary to analyse available room mapping applications. From all the possible applications, three have been taken into review. Two of the chosen applications were highest rated applications by Apartment Therapy <sup>13</sup> and the third one was found during the research for applications that could export mapped rooms as 3D objects.

Most of the applications require ARCore or ARKit to use. ARCore (Android) and ARKit (IOS) are SDKs which handle the augmented reality for the applications.

#### 3.1 Magicplan (Android, IOS)

Magicplan is a room mapping application for Android and IOS, it is made by the company Sensopia and it was released in the year 2011 <sup>14</sup>. At first the application was only usable on IOS devices, however by the year 2013 a working version for Android devices was released.

For room mapping, Magicplan uses camera input and creates a ground plane from the detected feature points. This detected ground plane is used as a reference for all the further measurements. The user is required to mark down the corners of the walls from which the room is created. The made walls are placed into the virtual world's room (Figure 6).

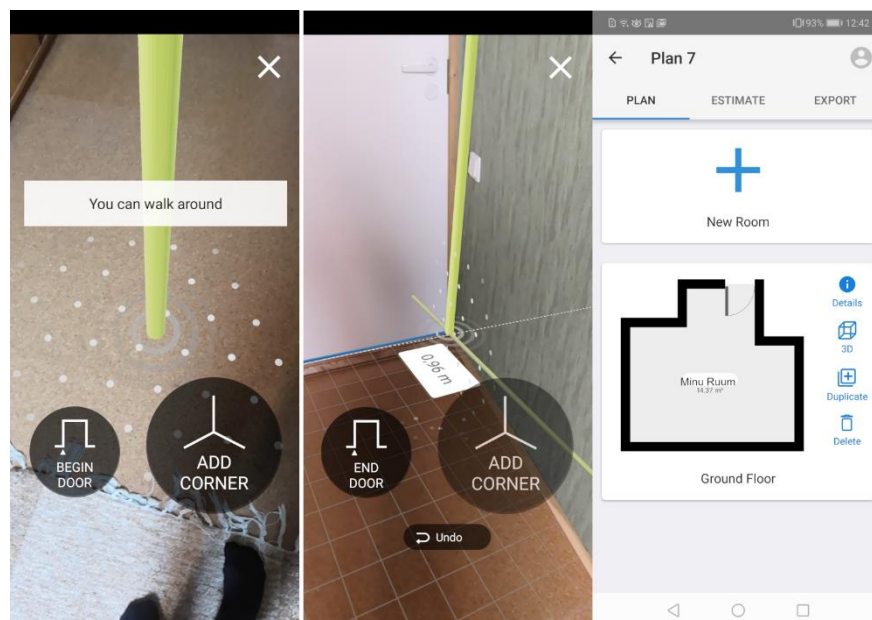


Figure 6. MagicPlan. Detection of the floor plane, marking corners and the final result

<sup>13</sup> <https://www.apartmenttherapy.com/the-7-best-apps-for-room-design-amp-room-layout-244213>

<sup>14</sup> [http://www.sensopia.com/webmail/pressrelease\\_eng.html](http://www.sensopia.com/webmail/pressrelease_eng.html)

Magicplan's advantage over the other analysed solutions is that the compatibility with ARCore or ARKit is not required. This means that with an older device, that does not support these SDKs, the room can be mapped. However, the measurements might not be that accurate, since the accurate wall calculations depend on the ground plane.

Magicplan's disadvantages are that the heights of the walls are not taken into consideration and the result is not exportable as a 3D object for further improvements. However, the application does let the user to export the measured floor-plan in PDF format.

### 3.2 RoomScan Pro (IOS)

RoomScan Pro is a room mapping application made for IOS devices by the company Locometic and was released in the year 2014<sup>15</sup>. This application gives the user a possibility to map rooms using 3 different methods: (1) similar to the previous application using the camera input, (2) by touching the walls with your phone and (3) a possibility to completely draw the floor plans in the application. Since the last one does not use any augmented reality components, it is not described in this thesis.

Room mapping by touching the walls is a unique way to get the floor plans of the room. This solution relies heavily on the IMU sensors of the device and does all the calculations based on the travelled distance between the walls. The user must touch every wall holding the phone vertically against the surface of the wall until it is registered, after that the user proceeds to another wall. During movement the distance between the walls is calculated and the intersection of the walls get marked as corners (Figure 7).

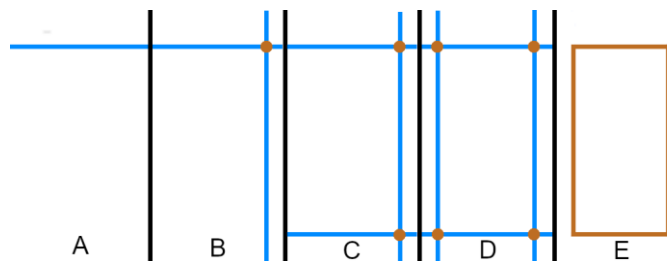


Figure 7. A to D is adding of the walls, E is the final result

The detection of the walls via camera input is a combination of the previously described wall touching method and the usual ground plane detection method. RoomScan Pro

<sup>15</sup> <https://www.cnet.com/news/roomscan-app-creates-floor-plans-just-by-tapping-walls/>

detects the ground plane and then requires the user to mark the line of the wall instead of the corner as in Magicplan (Figure 8).

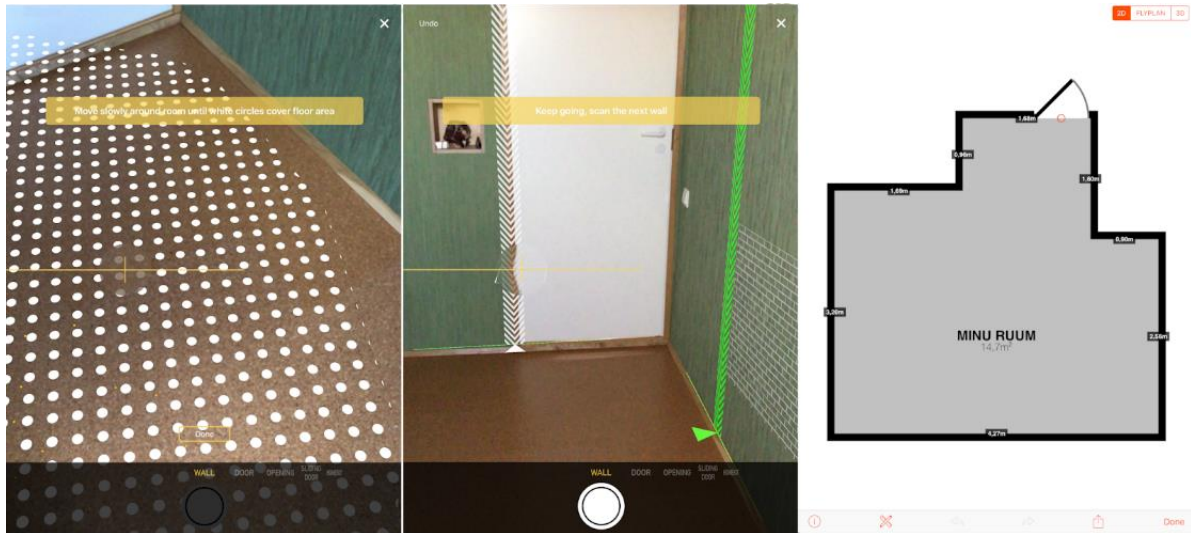


Figure 8. RoomScan Pro. Detection of the floor, marked walls and the result

The same method of calculating intersecting points is used as in previously described wall touching method, making the overall room mapping faster.

RoomScan Pro's disadvantage is that the height of the walls is not taken into consideration and the 3D-model of the created room layout cannot be used outside of the application. Although the room layout is not exportable as a 3D object, it is still possible to view the layout by exporting the result to PDF format.

### 3.3 PLNAR (IOS)

PLNAR is an application made for IOS devices by Smart Picture Technologies and its first release was at the year 2017 for IOS 11<sup>16</sup>. The application maps the room using similar techniques as the previously described application Magicplan, i.e. by using the detected floor plane and by user placed corners of the walls. However, PLNAR uses an additional height variable in the wall calculation process, making it more accurate to the room that is being mapped. The measurement of the wall height is made after all the corners for the

<sup>16</sup> <https://vrscout.com/news/plnar-ar-floor-plans/>



walls are placed (Figure 9).

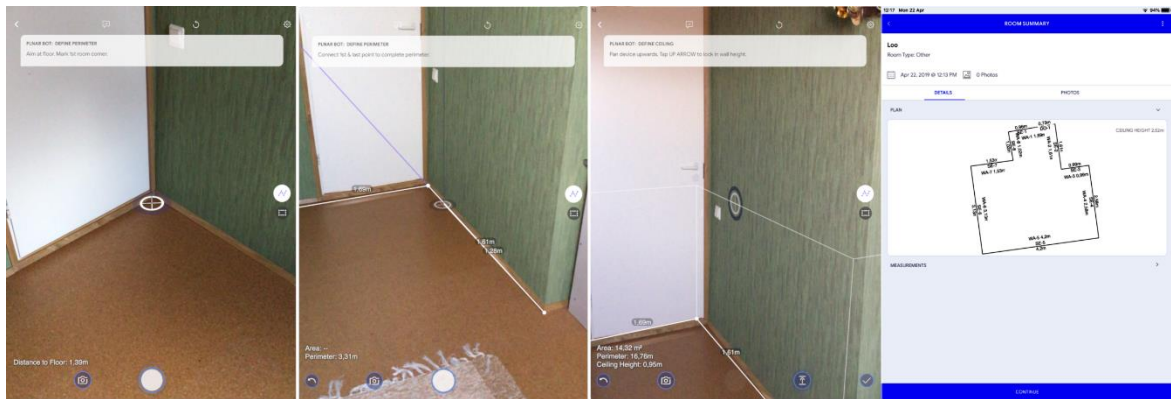


Figure 9. PLNAR. From left to right: detection of the floor, marked corners, measuring the height and the final result

This makes it good for the walls that are all the same height. On special cases where all the walls are not the same height, it does not perform that well. PLNAR has the capability to export the room model into different 2D and 3D formats<sup>17</sup>. Although the application is primarily aimed at the house selling and advertising businesses, it is still useable by simple home owners, who want to create layouts of their homes. The down-side to the application is its fee-based project creation system, meaning for every made new project certain fee must be paid<sup>18</sup>.

From all the tested applications, none provided the perfect solution. There either was no accurate wall height changing or the model could not be exported as a 3D object. When creating the RoomMapperAR these were the main features that were focused on. All the analysed applications used markerless AR and relied on the floor plane, since it did not require any additional preparation prior to mapping. So, it was decided to use markerless AR in created application as well.

<sup>17</sup><http://help.plnar.co/plnar-portal-faqs/how-to-access-download-2d-3d-files-from-plnar-projects>

<sup>18</sup> <https://portal.smartpicture3d.com/pricing-plans.html>

## 4 RoomMapperAR

RoomMapperAR was created in the Unity game engine with the Google's ARCore SDK. Unity was used for the making of the user interface and for managing the virtual objects in the virtual world. ARCore was used to detect the floor planes and handle the device's IMU sensors [5]. Although there are better SDKs in the terms of device support and with more improved features<sup>19</sup>, ARCore was chosen since its free to use and publish with.

The application follows the shown diagram of Figure 10 when moving between different modes and scenes.

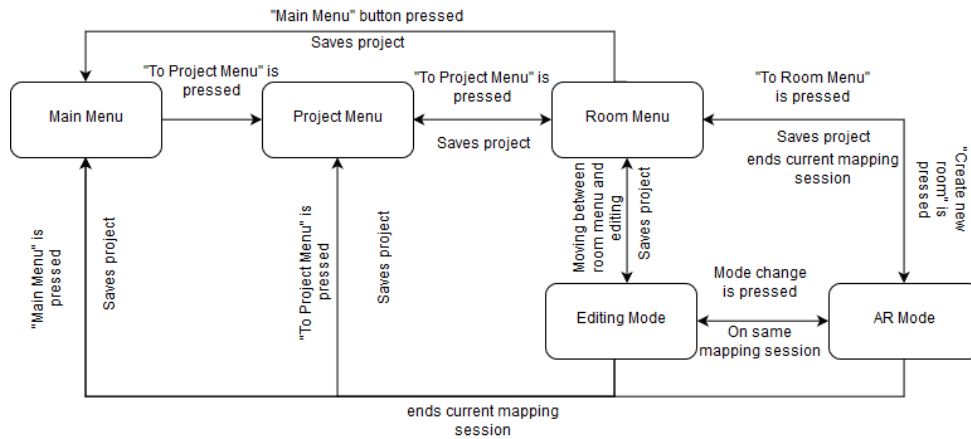


Figure 10. Movement between different modes and scenes

The main features of RoomMapperAR are the wall creation with a changeable height variable, semi-automatic door and window detection, and a possibility to export measured room as a 3D file.

### 4.1 Creating the Walls

Before making the walls, an initial floor plane is detected via the ARCore's built-in floor detection. ARCore's floor detection is using feature points to detect the boundaries of the floor<sup>20</sup>. This makes it markerless AR, meaning that if there are recognizable points, the device can track its position relative to the points. The detected floor plane is used for the initialisation of the *infinite virtual floor plane*, which is used for all further measurements. After floor detection, an invisible marker object is placed on the virtual floor using ray casting. The visible cursor object follows that invisible marker. Cursor is used in the marking down process as a reference for the corners of the walls and for measuring the height.

<sup>19</sup> <https://thinkmobiles.com/blog/best-ar-sdk-review/>

<sup>20</sup> [https://developers.google.com/ar/discover/concepts#motion\\_tracking](https://developers.google.com/ar/discover/concepts#motion_tracking)

The initial wall is made using 3 points that the user needs to mark down: first point is placed as the starting point of the current wall, second for the height and the third for marking the end of the wall (Figure 11).

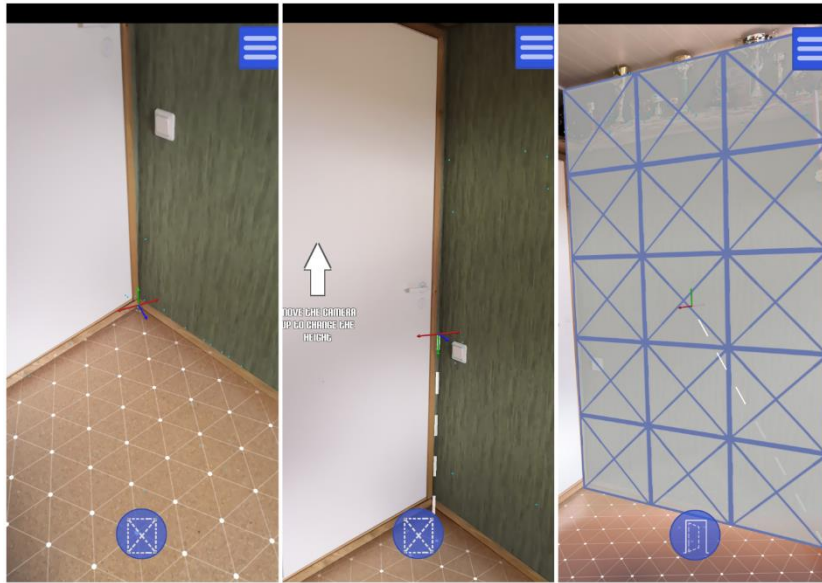


Figure 11. Marking of the first corner (left), measuring the height of the wall (middle) and the finished wall with the marked third corner (right).

After the first wall is placed, all the other walls are made using only one corner and the ending points of the last made wall, keeping the height of the new wall same to the previous. If the wall happens to be of a different height on some spots, the height variable can be changed for the specific corner or both corners. If height is changed for the last corner, all the new marked corners will be using the new measured height for the wall creation. The creation of the walls is available while the user is working on the newly created room. Once the user goes to the room menu (leaves the current room creation) the placing of the walls ends. After that the editing of the placed objects is still possible, but the creation of new walls into the selected room is not.

## 4.2 Marking the Doors and Windows

The doors and windows can be marked on the created walls by either of the two methods: (1) by using computer vision to detect the door and window frames or (2) by marking the outline of the object in the application. The latter is an alternative way to mark the doors and windows if the computer vision detection is not able to do it due to a complex scenario or poor lightning conditions. The manual outline marking is like the wall creation, only difference being that the marker moves on the created wall instead of the floor plane and the outline requires 2 points to make (Figure 15A).





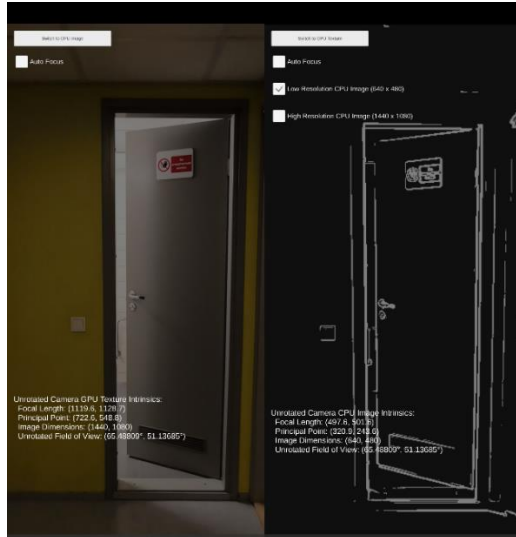


Figure 13. Before and after the Sobel edge detection

The result image with detected edges is used with the Hough Transform algorithm to find all the lines in the image.

The Hough Transform works by taking the x and y coordinates of a point and using them to calculate  $\rho$ , which is the distance from the centre of the coordinate system to the potential line. This gives us a perpendicular line to the potential line in the polar coordinate system [7]. The  $\rho$  is calculated using this formula:

$$\rho = |x \cdot \cos(\theta) + y \cdot \sin(\theta)|.$$

By assigning different  $\theta$  values in a range  $[-90,90)$  we will find different possible values for  $\rho$ , thus finding all the possible lines going through the selected x and y coordinates. To determine if the line is real or not, different  $\rho$  and  $\theta$  pair values get accumulated. The most collected  $\rho$  and  $\theta$  pairs are actual lines (Figure 14).

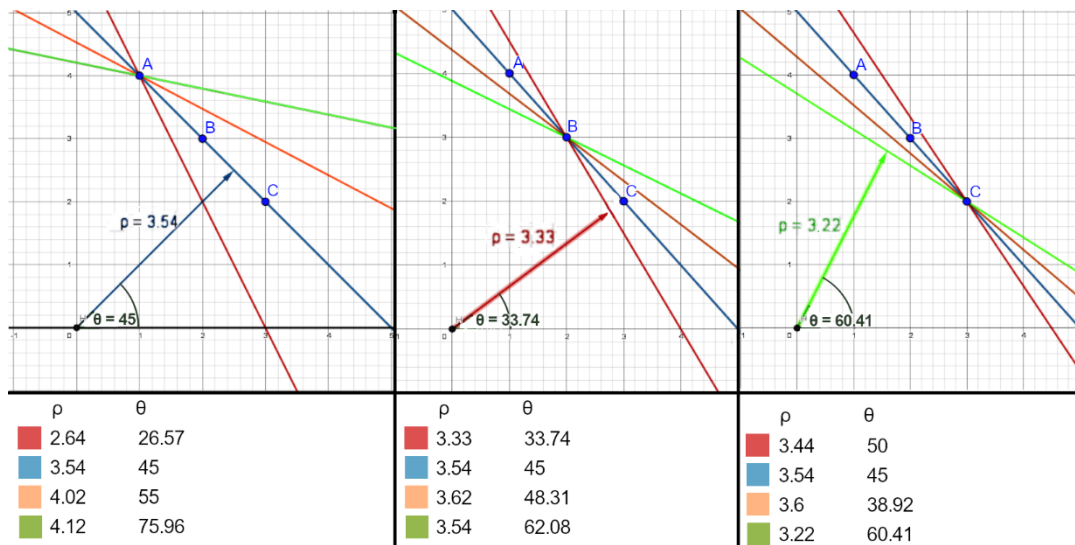


Figure 14. Visualisation of Hough Transform. A,B,C are the points of a line.

Because the goal of the detection is to find doors and windows, the range is shortened to calculate in a range  $[-90, -80] \cup [-10, 10] \cup [80, 90)$ , since most of the door and window borders / edges are either parallel or perpendicular to the floor. All the detected lines are then compared and the found parallel lines are put together in a list. All the pairs of parallel lines are compared to other parallel lines to determine the ones that are perpendicular. Both in finding the parallel and perpendicular lines a  $\pm 2^\circ$  error is allowed, since the pixels might not be perfectly on the line. The result is a list of rectangular shapes found from the screen space [7].

The problem with this is that other rectangular shapes such as cupboards get detected as well. This problem is solved by asking the user for confirmation if the found rectangle is indeed a door frame or a window (Figure 15B).

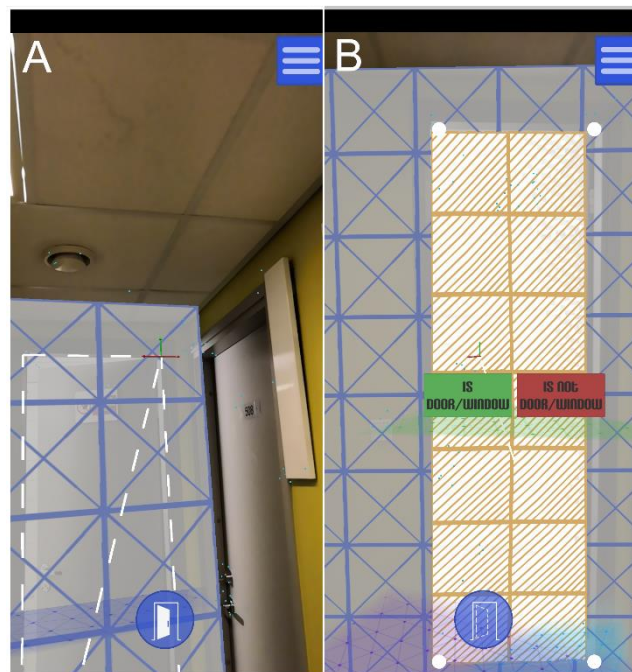


Figure 15 A – Manual door placement, B – Automatic door detection

If it is confirmed by the user, the object is made and taken into consideration when exporting the 3D-object.

### 4.3 Room Editing

For editing the made measurements or to delete the created walls, the user can select an object in *AR Mode* or *Editing Mode*. Upon selecting a wall, a toolbar with all the changeable parameters appears on top of the screen. If the corner editing is selected, all the vertices of the selected wall are displayed as selectable spheres. The wall dimensions can

then be changed by moving the directional arrows near the spheres. The directional arrows are used to move the vertex along one axis at a time. This assures that the changes made in one axis do not affect the other axes (Figure 16). Other parameters, such as position, rotation and scale, are changed in an interactive approach. For example, when the movement parameter is toggled, the dragging of the object causes it to move to the dragged point. On the toggle of the rotation parameter, additional buttons appear for toggling the precise rotation.

If the *Editing Mode* is accessed during the same measuring session, the collected information about the environment is kept. Meaning the user can return to the *AR Mode* and all the placed walls will be on their correct positions. The information about the detected feature points and position calculations are kept running in the background while working in the *Editing Mode*. Although it takes away processing power from the device, it is useful as it makes fixing misplaced walls and other room objects easier and more accurate.

Editing a room after the end of the measuring session is possible, but the *AR Mode* is not accessible since all the information about the feature points and the device's position is lost. The creation of a feature point uses different variables like the position during the creation, illumination of the detectable surface, angle of the device in the relation to the mentioned surface and many more, making it difficult to recreate, so the information about it is not saved. Manual placement of the made room in *AR Mode* might be possible, but in the current state of the application this feature was not implemented.

#### 4.4 Furniture Placement

To use the application as a room planner, a feature for placing common furniture objects is also added. The objects can be placed in *AR Mode*, making it easier to see the placement of the object in the room according to other objects, or in *Editing Mode*, where the user gets the overview of the created room. Furniture objects parameters such as position, rotation, scale and colour are changeable. This allows the user to refurnish the room for the purpose of having the visual representation of possible changes in the room after furnishing.



Figure 16 Selected wall with visible corners. Darker toolbar parameter shows what parameter changing is toggled

A similar control scheme with the wall is used to manipulate furniture - toolbar at the top of the screen allows to choose the parameter being changed. However, instead of the corner editing parameter, furniture objects have colour changing parameter.

#### 4.5 Exporting the Object

The application uses Wavefront's OBJ format to export the room as a 3D-object. This format was suitable for exporting because of the following reasons:

- it does not require any extra SDKs,
- it is easily writeable in text format
- most of the common 3D applications support it <sup>21</sup>.

OBJ format takes 4 types of parameters: (1) Position of each vertex, (2) UV position of each texture coordinate, (3) vertex normals, (4) the faces that make each polygon <sup>22</sup>.

To export the selected room and all the associated meshes, the walls and furniture objects were made as children of the room object. This way it is possible to get all the meshes using only a reference to the parent object. From the meshes we get all the needed information which will be added into the output string (Appendix IV). The result (Figure 17) will be saved into the *ExportedObjects* folder on the user's device.

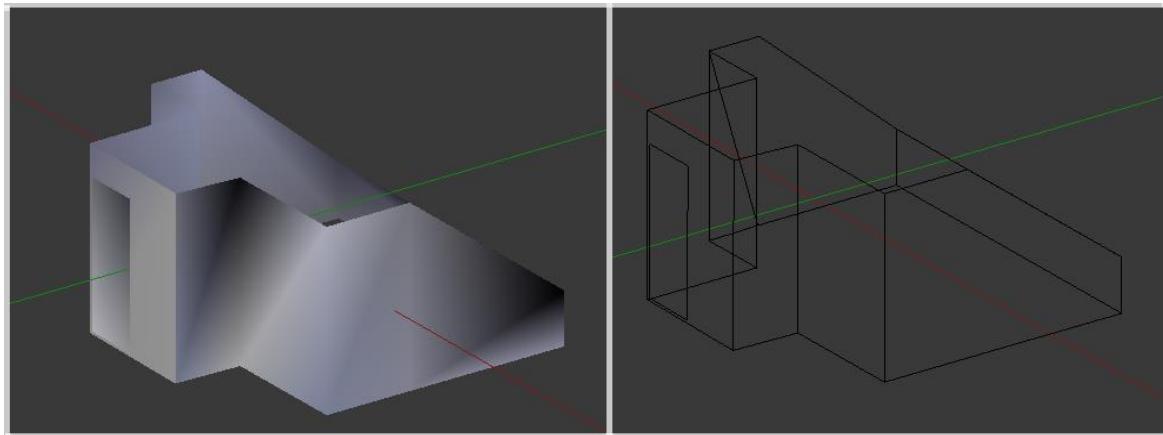


Figure 17. 3D representation of the measured room.  
Normal view (left) and wireframe view (right) of the object

The initial idea was to make a new folder inside the root folder of the Android device, however, this was not achieved since the Unity does not have so many built in permission options. So as of now the file is saved in the data folder of the Application ("storage/emulated/0/Android/data/com.ottsaar.roommapperar/files/ExportedObjects").

<sup>21</sup> <https://fileinfo.com/extension/obj>

<sup>22</sup> [https://en.wikipedia.org/wiki/Wavefront\\_.obj\\_file](https://en.wikipedia.org/wiki/Wavefront_.obj_file)

## 5 Testing

In any software development it is necessary to test the application in the real-world scenarios and on real people. While developing the application, some aspects that are natural for the developer, might not be so clear for the potential users [8]. That is why the usability testing is important and why it was conducted for this application as well. Since the application will be used on different devices, it was tested on different brands of smartphones.

### 5.1 Devices/Platform Testing

The application was tested on 5 different devices:

- Huawei Honor 10
- Nokia 7 plus
- Xperia XZ1 Compact
- Samsung Galaxy 7 Edge
- Samsung Note.

The application was also tried on Xiaomi Mi A2 lite, but because the device does not support ARCore, the application did not even install on the device. The known problem with ARCore, as with most AR SDK-s, is that not all the available devices support it, due to lack of computational power or not having the needed sensors<sup>23</sup>.

All the main features worked as intended, the floor plane was successfully detected on all of the devices, wall creation and the exporting of the 3D object worked. The only problems were seen on the automatic detection of the door and window objects. This is possibly due to the devices having different performance capabilities, so every device performed differently. Overall it seems that the algorithms used in the door and window detection are not that efficient, causing lag spikes on weaker devices and on more complex scenes.

### 5.2 Usability

Since the application is aimed at simple home owners, it was necessary to test how understandable the wall creation process was, how added features performed and what could be improved. For this 5 people tested RoomMapperAR. One tester had previous experience with room design. None of the participants had used any room mapping

---

<sup>23</sup> <https://developers.google.com/ar/discover/supported-devices>

applications before, which was good for the testing, since it showed how well the application managed to explain itself and how people with no previous experience would use it.

The testers were given a simple task list of things they had to do (Appendix V). They were asked to speak loudly about different actions they took and if any problems occurred to try and solve them themselves. On every testing session the author had to help the testers in certain parts, meaning the application needs to explain itself a bit more. The test environment for room mapping was *Axinom* (207) room in the Ülikooli 17, Tartu. The room was chosen, because it was well lit and spacious enough for moving around.

### 5.2.1 Room Creation

The testers were asked to rate different aspects of the room creation, to pin down the specific problems that occurred during the room mapping in AR Mode. The viewed aspects were wall creation, door/window detection and room editing overall.

#### 5.2.1.1 Wall Creation

The test results showed that the creation of the walls and the feature to change the height of the wall were overall useful, but due to unclear controls and lack of information it was not very comfortable to use. When asked to rate the usefulness of the height changing from 1 (“completely useless”) to 6 (“very useful”), the answers were more to the positive side (Figure 18).

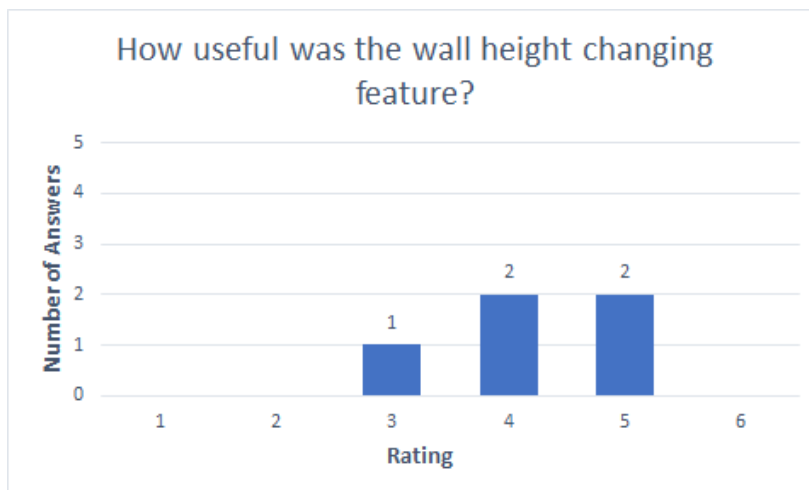


Figure 18. The usefulness of the height variable changing

In the follow up question it was explained that the idea itself was good, that it makes the measured rooms more accurate, but accessing the wall height changing was not so clear.

At first the testers did not even notice the ability to change the height despite the tip pointing it out.

#### **5.2.1.2 Door and Window Placement**

All the participants preferred the manual marking of the door and window objects (Figure 19).

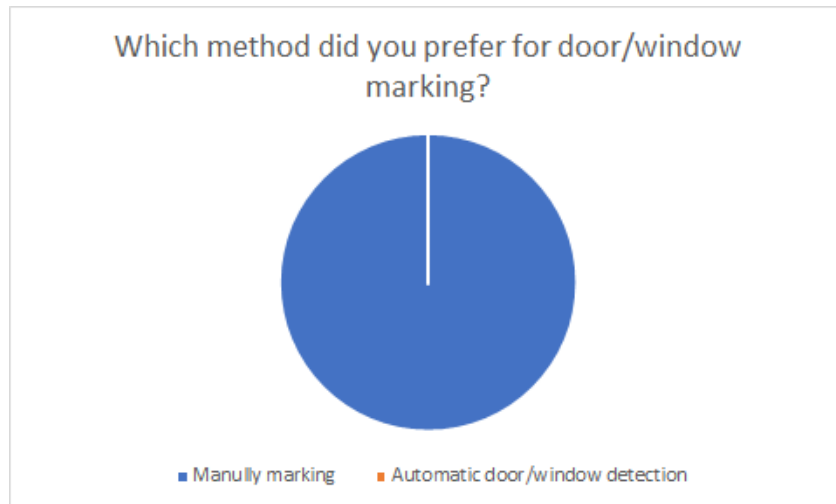


Figure 19. The testers preferences for door marking.

The explanations given showed that the current state of the automatic detection was not very user-friendly, in a sense that it sometimes caused performance spikes in the overall wall marking process, making it quite uncomfortable.

It was concluded from the testing that the used algorithms were not efficient enough in the mobile environment for rectangle detection and the implementation should be further worked on.

#### **5.2.1.3 Tracking of the Measured Walls**

The tests showed that the application did lose the tracked walls more than once (Figure 20). The testers complained that the misaligned walls on the loss of tracking were disorienting. Usually showing the starting point of the room creation readjusted the already made room.



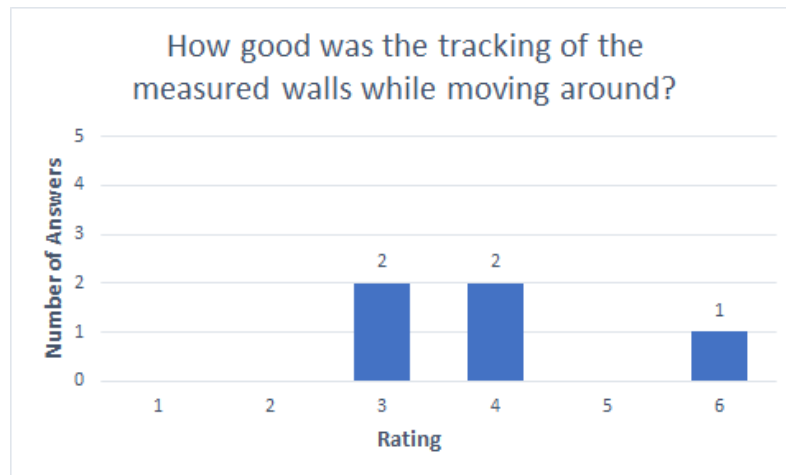


Figure 20. Ratings for the performance of room tracking.  
1 (“lost it all the time”) to 6 (“not noticeable problems”).

One of the testers rated the wall tracking with a score 6 out of 6, meaning there were not any noticeable problems. When asked what the tester did differently in his room mapping session, he said that he took more time in finding the floor plane. This might have caused the ARCore’s floor plane detection to build bigger initial floor, meaning more recognisable points to fall back to when the device’s position gets lost.

### 5.2.2 Editing Objects

Since the editing of the objects in RoomMapperAR is quite different from available room mapping applications it was necessary to ask how well it performed.

The overall ratings for the usability of the object placement and editing showed that the editing of the objects need a lot of polishing and reworking with the average of 3.8 out of 6 (Figure 21).

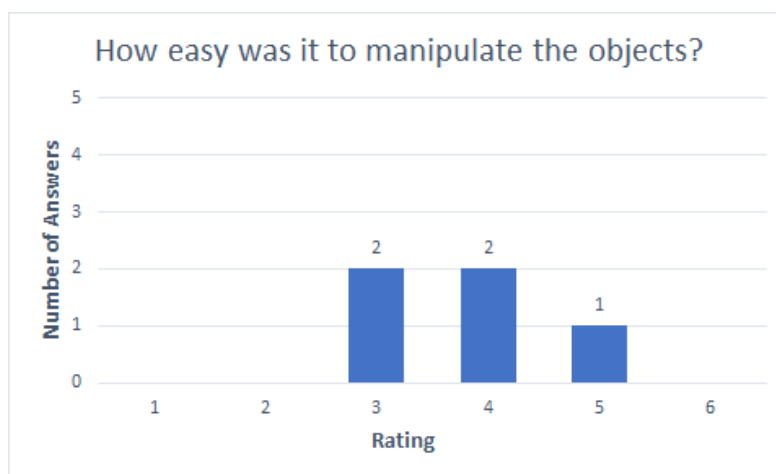


Figure 21. Usability of the object placement.  
1 (“very difficult”) to 6 (“very easy”)

It was suggested by the testers, that the directional axes for manipulating objects would be better and more accurate than the interactive approach that was taken. It was also suggested that toggleable precision that was used in the rotation tool to be added to other tools as well.

If asked which mode for editing objects was easier for the testers the results were mixed (Figure 22).

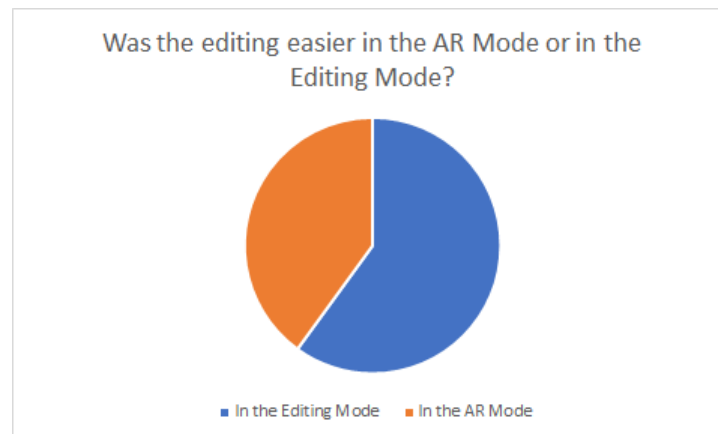


Figure 22. Diagram showing the preferences of the testers when editing

Two testers who favoured the *AR Mode*, explained that editing and placing the objects in the room felt more natural, since the user was technically still in the room. The tester could scale the objects better and see the results immediately. The testers who preferred *Editing Mode* said that the overview of the room was good for aligning objects and for seeing the results from the other perspective.

### 5.2.3 Graphical User Interface

First impressions mean a lot. Even if the application is useful, bad controls and bad design choices can throw potential users off and, in many cases, make the application useless <sup>24</sup>. The testers were asked to rate the overall visual style of the GUI (graphical user interface) and the comprehension of it (Figure 23).

<sup>24</sup> <http://adscendblog.com/app-making-great-first-impression/>

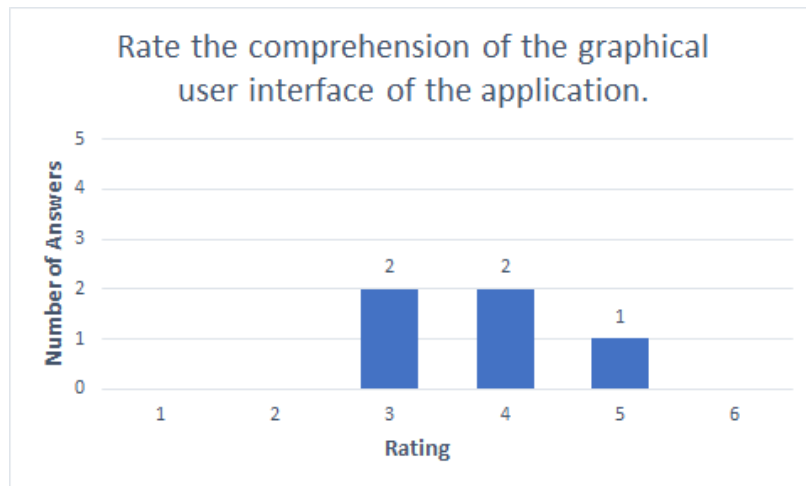


Figure 23. Comprehension of the user interface.  
1 (“awful”) to 6 (“perfect”)

The feedback from the questions showed that the chosen visual style itself was not bad, but as mentioned by the testers before, the way the controls worked was too complex to understand. It was recommended to give more information about the different buttons and overall make them a bit bigger.

#### 5.2.4 Comparison to MagicPlan

After testing RoomMapperAR the testers were asked to map the room with the existing Android application MagicPlan. Applications were compared to each other to determine if the RoomMapperAR would be a good alternative to the already existing application.

The overall rating of the MagicPlan was better with an average of 4.4 out of 6, compared to the RoomMapperAR average of 3.6 out of 6 (Figure 24).

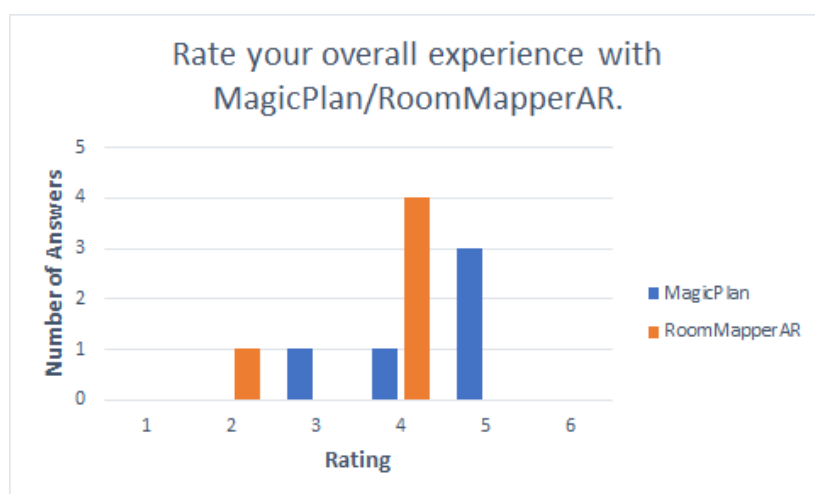


Figure 24. Overall experience for both of the applications.  
1 (“very bad”) to 6 (“very good”)

The testers liked the polished feel of the Magicplan and its simple controls with one button for starting the door and one for adding the corner of the wall.

When comparing the room creation, the testers favoured the Magicplan (Figure 25).

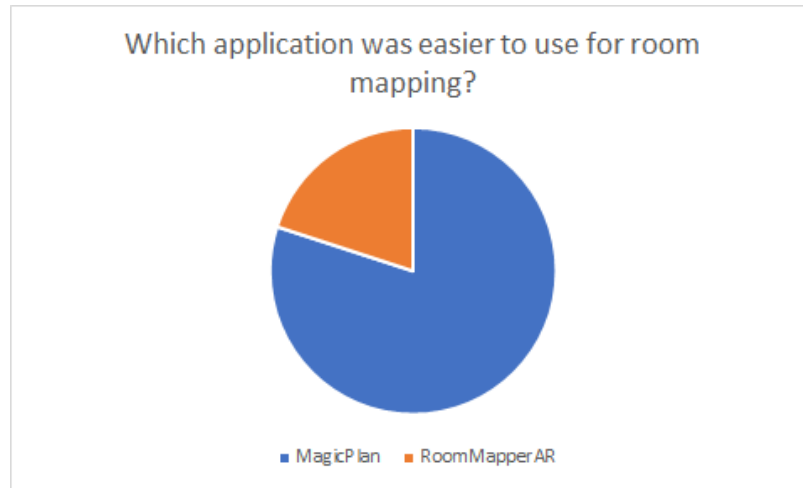


Figure 25. Preferred application for room mapping

One tester who preferred RoomMapperAR over Magicplan explained that the visual representation of the made walls made it easier. Other testers liked the simple controls of the Magicplan and the overall smoothness of the application. Overall it was said that both applications had flaws when placing corners.

When comparing furniture placement feature of the applications, testers preferred the axis-based furniture placement of the Magicplan over RoomMapperAR's more interactive approach.

Two of the testers who liked the RoomMapperAR's approach said that the manipulation in the 3D gave them better visual overview. Other testers said that the 2D view of the room felt better, it was enough for accurate object manipulation. When using the axis-based manipulation they felt more in control of the changes that were happening.

#### 5.2.5 Additional Suggestions

The following additional suggestions were made by the testers:

- The snapping tool should be toggleable
- Additional 2D view to have overview of the created room
- Undo button
- The "tips" given by the application should be always visible in some form
- Simpler controls for height variable changing

The testers have made some good suggestions regarding the application and the way the application is supposed to work. These suggestions will be added into the application in some form.

### 5.3 Result Comparison

To determine if the RoomMapperAR, despite the difficult controls, could deliver similar results as the reviewed applications, all of them were put into a test. The task was to map a room as close to the original as possible. Since RoomMapperAR does not have a 2D view of the room in the application, a top-down orthographic projection<sup>25</sup> of the exported room was made in Blender 3D. The results were compared side-by-side (Figure 26) and by overlaying the pictures on top of the original room layout to see how close they were to the original (Figure 27).

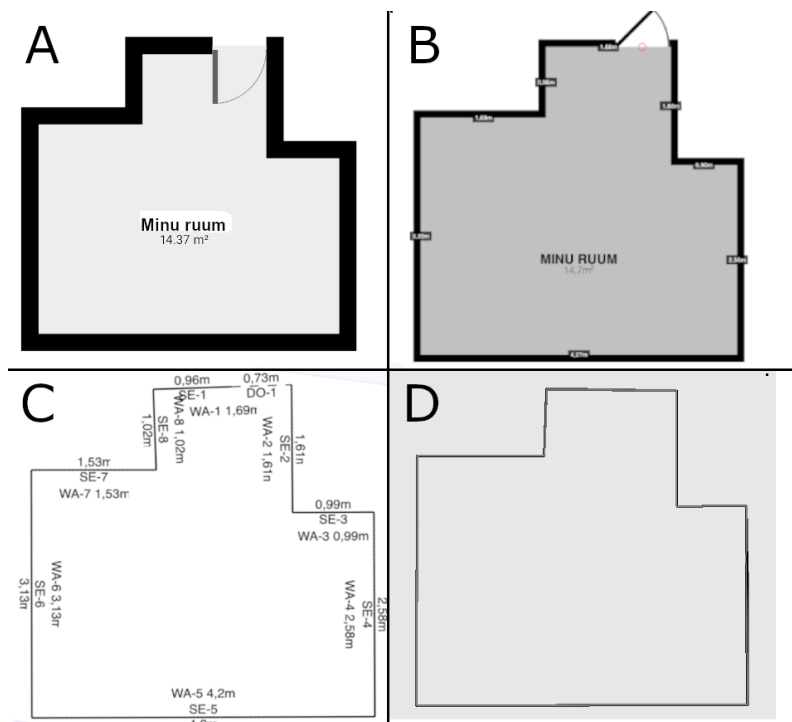


Figure 26. Side-by-side comparison of the results. A – Magicplan, B – RoomScan Pro, C – PLNAR, D – RoomMapperAR

The comparison shows that all the applications managed to capture the overall layout of the room correctly. Magicplan and RoomScan Pro had the best snapping and alignment of

<sup>25</sup> [https://en.wikipedia.org/wiki/Orthographic\\_projection](https://en.wikipedia.org/wiki/Orthographic_projection)

the corners, since all the walls were straight. PLNAR and RoomMapperAR had some crooked walls near the end of the measuring session.

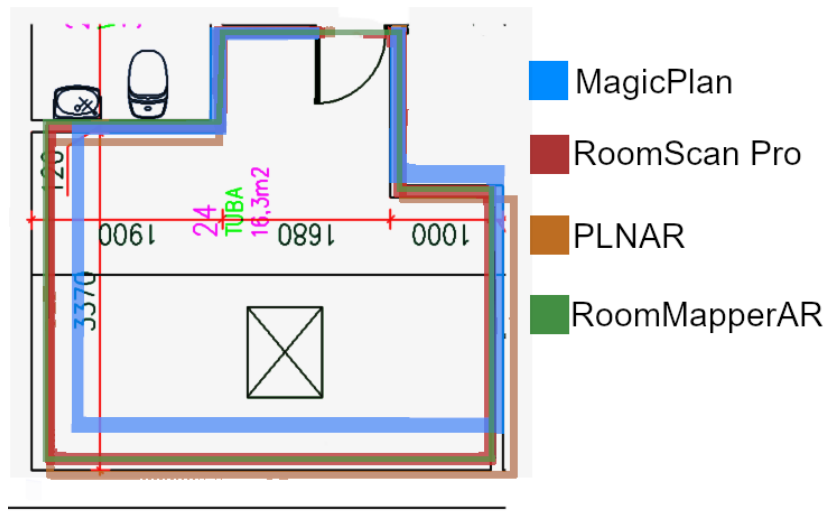


Figure 27. All the results overlayed on top of the original layout

The overlay images were aligned using the door frame as a reference, since this was the start point of the mapping process for all of the applications. The comparison shows that the RoomScan Pro and RoomMapperAR were closest to the original room layout, both showing really similar result. Magicplan had an excellent alignment of the door, but during the movement to other corners the calculations went wrong.

Since only PLNAR had the ability to export the measured room as a 3D object, it was used in comparison with the export result of RoomMapperAR (Figure 28).

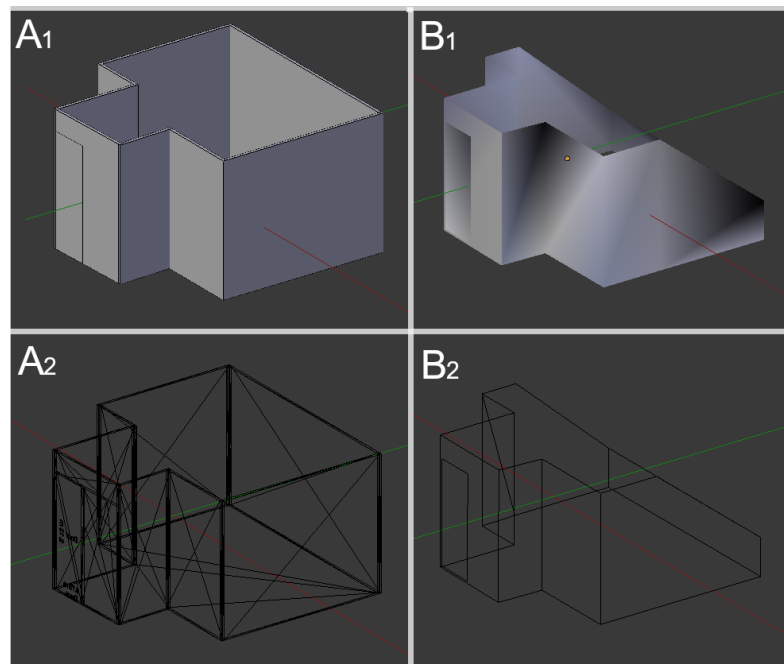


Figure 28. Side-by-side view of the 3D objects. A – PLNAR, B – RoomMapperAR.

While RoomMapperAR captured the height changes of the room correctly, the result of the PLNAR's export is more polished, since the walls have thickness to them.

#### **5.4 Future Improvements**

All the suggestions made by the testers will be implemented in some way, since most of them were of things that were overlooked at first. Things that felt good and easy while developing the application, were not so useable by other people. The toggling of the height changing by holding the placement button was not so intuitive after all and caused lots of confusion. Hidden additional tools seemed like a good idea since the view was not cluttered with unnecessary information, they were only shown when the height needed to be changed. This however was causing more confusion than an extra button for height changing would have caused.

The saving system of the project will be improved as well, since it is not very comfortable to navigate in the Android files and search for the application folder. Since this is a common problem with Unity Android applications, hopefully with the release of a new Unity version, this will be improved as well.

## 6 Conclusion

During the work for the thesis a room mapping application named RoomMapperAR was made. Available room mapping applications were analysed to determine the common features of a room mapping application, which were followed when creating RoomMapperAR.

The main features of the made application are wall creation with changeable height variable, automatic door and window detection and exporting of the made room as a 3D object. For the automatic door and window detection, Hough Transform algorithm was used. Basic features for room editing and furniture placement were added to make the application useable as a room designer. Objects in the room and the placed walls can be manipulated in two different modes: in *ARMode*, which shows the object through the camera view in the real world and in *Editing Mode*, that is good for more precise object manipulation.

RoomMapperAR was tested on different devices and the result of the mapping process was compared to the results made by available applications. It was concluded that the created layout was similar to the ones created by the reviewed applications and added height variable made the exported room truer to the original layout. To determine the usability of the RoomMapperAR usability tests were conducted. There were total of five testers, who did not have any previous experience with room mapping applications. The testers were asked to rate the main features of the application, comprehension of the user interface and the overall flow. After testing the RoomMapperAR, the testers were asked to map the same room with available room mapping application Magicplan. It was concluded that the made application needs further improvements for it to be useable by users with no previous experience.

I plan to continue working on this application. All the suggestions made by the testers will be implemented, the overall flow of the application will be reworked and tested, before releasing RoomMapperAR to Google Play Store. I plan to release it as a free-to-use application, so the people could use it as a way to export their rooms as 3D objects.



## 7 References

- [1] Furht B. Handbook of Augmented Reality; New York: Springer, 2011.
- [2] Craig B.A. Understanding Augmented Reality, 1st ed.; San Francisco: Morgan Kaufmann, 2013.
- [3] The Ultimate Guide to Understanding Augmented Reality (AR) Technology. [Online] Available: <https://www.realitytechnologies.com/augmented-reality/>. [Accessed Februar 2019].
- [4] Pagani A., Henriques J., Stricker, D. Sensors for Location-based Augmented Reality the Example of Galileo and Egnos.; ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences. Vol XLI-B1, 2016. pp 1173-1177.
- [5] Linowes J., Babilinski K. Augmented Reality for Developers. Packt Publishing, 2017.
- [6] Gonzalez C.R., Woods E.R. Digital Image Processing 3rd ed.; New Jersey: Pearson, 2006.
- [7] Gonzalez C.R., Woods E.R., Eddins.L.S. Digital image processing using Matlab 2nd ed.; United States of America: Gatesmark, 2009.
- [8] Nielsen J. Usability 101: Introduction to Usability. [Online]. Available: <https://www.nngroup.com/articles/usability-101-introduction-to-usability/>. [Accessed April 2019].

## Appendix

### I. Glossary

<b>ARCore</b>	Google's platform for building AR experiences. <sup>26</sup>
<b>ARKit</b>	Apple's platform for building AR experiences. This only works on IOS devices. <sup>27</sup>
<b>Game engine</b>	A game engine is a software-development environment designed for building video games. Developers are using game engines for constructing games for different available platforms. <sup>28</sup>
<b>IMU(Inertial Measurement Unit) sensor</b>	An inertial measurement unit (IMU) sensor is a sensor that measure an acceleration and angular rate of the device. Accelerometers, gyroscopes and magnetometers are the most common IMU sensors. <sup>29</sup>
<b>Orthographic projection</b>	Orthographic projection (sometimes orthogonal projection) is a way to represent three-dimensional objects in two dimensions. <sup>30</sup>
<b>Polar coordinates</b>	The polar coordinate system is a two-dimensional coordinate system in which each point on a plane is determined by a distance from a reference point and an angle from a reference direction. <sup>31</sup>
<b>Rendering</b>	Rendering is a process of generating an image from a 2D or 3D model. <sup>32</sup>
<b>Software development kit (SDK)</b>	Software development kit is a collection of different software development tools that are used in the creation of an application. <sup>33</sup>
<b>Usability testing</b>	Usability testing is testing of how easy the user interface is to use. There are usually 5 components that are viewed in these tests: learnability, efficiency, memorability, errors and satisfaction.[8]

---

<sup>26</sup> <https://developers.google.com/ar/discover/>

<sup>27</sup> <https://developer.apple.com/arkit/>

<sup>28</sup> [https://en.wikipedia.org/wiki/Game\\_engine](https://en.wikipedia.org/wiki/Game_engine)

<sup>29</sup> <https://www.hillcrestlabs.com/posts/what-is-an-imu-sensor>

<sup>30</sup> [https://en.wikipedia.org/wiki/Orthographic\\_projection](https://en.wikipedia.org/wiki/Orthographic_projection)

<sup>31</sup> [https://en.wikipedia.org/wiki/Polar\\_coordinate\\_system](https://en.wikipedia.org/wiki/Polar_coordinate_system)

<sup>32</sup> [https://en.wikipedia.org/wiki/Rendering\\_\(computer\\_graphics\)](https://en.wikipedia.org/wiki/Rendering_(computer_graphics))

<sup>33</sup> [https://en.wikipedia.org/wiki/Software\\_development\\_kit](https://en.wikipedia.org/wiki/Software_development_kit)

## **II. Accompanying Files**

The archive file contains following files:

- roomMapperAr.apk – the installation package for the application
- demo.mp4 – the demo video of the use of the application
- demo.obj – the exported file that was made during the demo session
- roommapperar\_answers.csv – the answers by the testers, who participated in the usability testing.
- roommapperar\_questions.pdf – the questions asked from the testers

### **III. Source Code**

Source code for the application can be found here: <https://bitbucket.org/ottsaaar/room-mapper-ar/src/master/>

## IV. OBJ Example

This an example of an OBJ file. This current file is of a room that only contains 4 walls.

```
#project.obj
#Friday, 10 May 2019
#00:37:25
#-----

#project
#-----
g project
#4walls
#-----
g 4walls
#Wall(Clone)
#-----
g Wall(Clone)
v 0.4587688 -0.9846086 -0.6196147
v 0.4531536 1.276457 -0.6188213
v -0.890749 -0.9846083 -0.4187175
v -0.890749 1.276458 -0.4187175
v 0.4587688 -0.9846086 -0.6196147
v 0.4531536 1.276457 -0.6188213
v -0.890749 -0.9846083 -0.4187175
v -0.890749 1.276458 -0.4187175

vn -0.3303168 -0.7834687 -0.5263722
vn -0.3303168 -0.7834687 -0.5263722
vn -0.3303168 -0.7834687 -0.5263722
vn -0.3303168 -0.7834687 -0.5263722
vn 0.3303168 0.7834687 0.5263722
vn 0.3303168 0.7834687 0.5263722
vn 0.3303168 0.7834687 0.5263722
vn 0.3303168 0.7834687 0.5263722

f 1/1/1 3/3/3 2/2/2
f 3/3/3 4/4/4 2/2/2
f 2/2/2 4/4/4 3/3/3
f 2/2/2 3/3/3 1/1/1
#Wall(Clone)
#-----
g Wall(Clone)
v -0.890749 -0.9846083 -0.4187175
v -0.890749 1.276458 -0.4187175
v -0.6379719 -0.9846083 1.279235
v -0.6379719 1.276458 1.279235
v -0.890749 -0.9846083 -0.4187175
v -0.890749 1.276458 -0.4187175
v -0.6379719 -0.9846083 1.279235
v -0.6379719 1.276458 1.279235

vn -0.304342 -0.92711 0.2187308
vn -0.304342 -0.92711 0.2187308
vn -0.304342 -0.92711 0.2187308
vn -0.304342 -0.92711 0.2187308
vn 0.304342 0.92711 -0.2187308
vn 0.304342 0.92711 -0.2187308
vn 0.304342 0.92711 -0.2187308
vn 0.304342 0.92711 -0.2187308

f 9/9/9 11/11/11 10/10/10
f 11/11/11 12/12/12 10/10/10
f 10/10/10 12/12/12 11/11/11
f 10/10/10 11/11/11 9/9/9
#Wall(Clone)
#-----
g Wall(Clone)
```

```

v -0.6379719 -0.9846083 1.279235
v -0.6379719 1.276458 1.279235
v 0.8798809 -0.9846083 1.053267
v 0.8798809 1.276458 1.053267
v -0.6379719 -0.9846083 1.279235
v -0.6379719 1.276458 1.279235
v 0.8798809 -0.9846083 1.053267
v 0.8798809 1.276458 1.053267

vn 0.3652274 -0.919812 0.1433699
vn 0.3652274 -0.919812 0.1433699
vn 0.3652274 -0.919812 0.1433699
vn 0.3652274 -0.919812 0.1433699
vn -0.3652274 0.919812 -0.1433699
vn -0.3652274 0.919812 -0.1433699
vn -0.3652274 0.919812 -0.1433699
vn -0.3652274 0.919812 -0.1433699

f 18/18/18 20/20/20 19/19/19
f 18/18/18 19/19/19 17/17/17
f 17/17/17 19/19/19 18/18/18
f 19/19/19 20/20/20 18/18/18
#Wall(Clone)
#-----
g Wall(Clone)
v 0.8798809 -0.9846083 1.053267
v 0.8798809 1.276458 1.053267
v 0.5078106 -0.9846085 -0.6138622
v 0.5078106 1.276457 -0.6138622
v 0.8798809 -0.9846083 1.053267
v 0.8798809 1.276458 1.053267
v 0.5078106 -0.9846085 -0.6138622
v 0.5078106 1.276457 -0.6138622

vn 0.1595271 -0.7993301 -0.5793293
vn 0.1595271 -0.7993301 -0.5793293
vn 0.1595271 -0.7993301 -0.5793293
vn 0.1595271 -0.7993301 -0.5793293
vn -0.1595271 0.7993301 0.5793293
vn -0.1595271 0.7993301 0.5793293
vn -0.1595271 0.7993301 0.5793293
vn -0.1595271 0.7993301 0.5793293

f 26/26/26 28/28/28 27/27/27
f 26/26/26 27/27/27 25/25/25
f 25/25/25 27/27/27 26/26/26
f 27/27/27 28/28/28 26/26/26

```

## V. Tasklist for Testers

The testers were asked to do the following tasks:

- Create a new project
- Create a new room
- Start mapping the room
- Try changing the height of the room
- Select a wall object and try to manipulate it in *AR Mode*
- Switch to *Editing Mode* and try manipulating it again
- Add a furniture object and manipulate it in *AR Mode*
- Switch to *Editing Mode* and try manipulating it again
- Add a door to the room using manual door placement
- Add a door using automatic door detection
- Exit to room menu and create a new room
- Try the application yourself

To keep the testing as productive as possible these little tasks helped a lot, because the testers had certain goals that they needed to accomplish and that made them focused.

## **VI. License**

### **I, Ott Saar**

herewith grant the University of Tartu a free permit (non-exclusive licence) to reproduce, for the purpose of preservation, including for adding to the DSpace digital archives until the expiry of the term of copyright,

### **RoomMapperAR – A Mobile Augmented Reality Room Mapper,**

supervised by Raimond-Hendrik Tunnel

2. I grant the University of Tartu a permit to make the work specified in p. 1 available to the public via the web environment of the University of Tartu, including via the DSpace digital archives, under the Creative Commons licence CC BY NC ND 3.0, which allows, by giving appropriate credit to the author, to reproduce, distribute the work and communicate it to the public, and prohibits the creation of derivative works and any commercial use of the work until the expiry of the term of copyright.
3. I am aware of the fact that the author retains the rights specified in p. 1 and 2.
4. I certify that granting the non-exclusive licence does not infringe other persons' intellectual property rights or rights arising from the personal data protection legislation.

*Ott Saar*

*Tartu, 10/05/2019*