

Eyob Woldegiorgis

# 3D Model Data Source for Wikitude

---

Helsinki Metropolia University of Applied Sciences

Bachelor of Engineering

Media Engineering

Thesis

29 April 2014

Author(s) Title	Eyob Woldegiorgis 3D model data source for Wikitude
Number of Pages Date	33 pages + 2 appendices 29 April 2014
Degree	Bachelor of Engineering
Degree Programme	Media Engineering
Specialisation option	Software Engineering
Instructors	Kari Salo (Principal Lecturer) Petri Vesikivi (Principal Lecturer)
<p>The purpose of this Bachelor's Thesis was to design an augmented reality based mobile application for Museum of Finnish Architecture. In this document, a brief overview of augmented reality technology, augmented reality browsers, theoretical aspects and finally the practical work have been described.</p> <p>The practical part was carried out in two parts. The first part was, a Wikitude based augmented reality mobile application that provides information about the architecture of buildings around Helsinki city. The information is offered in English, Finnish, Russian and Swedish languages. Also, a demonstration was made on how to support the application with a 3D model data was conducted.</p> <p>The content management system was made using MySQL-database and PHP as a backend programming language. Furthermore, ARML, HTML, Wikitude ARchitect API and CSS have been used for the application and demonstration part.</p> <p>The application has been tested, and the outcome was positive but it differs slightly depending on the platform. The application is currently published and available for users with Android, iPhone, Windows or Blackberry phones and tablets.</p>	
Keywords	Augmented Reality, AR, Wikitude, ARchitect, Mobile applications, Smartphones



# Contents

## List of Abbreviations

1	Introduction	1
2	Overview	2
2.1	Evolution of Augmented Reality (AR)	2
2.1.1	History of AR	2
2.1.2	AR today	4
2.1.3	AR in the future	4
2.2	Why AR?	5
2.3	Application areas of AR	5
2.3.1	AR applications on mobile devices	6
3	AR browsers and 3D model	7
3.1	AR Browsers	7
3.1.1	Junaio	9
3.1.2	Layar	9
3.1.3	Wikitude	10
3.2	3D models and Wikitude	12
3.2.1	Wikitude SDK	12
3.2.2	Basic APIs used for developing 3D model on Wikitude	14
3.2.3	Wikitude 3D encoder	22
3.2.4	Problems and Known Issues	23
4	Design and Implementation	23
4.1	Methods and materials	23
4.2	Development of ARML based content management system	25
4.2.1	Data management	26
4.2.2	Testing	28
4.3	Demonstration of 3D models on Wikitude AR browser	30
4.4	Benefits of the application	31



4.5	Limitations	32
4.6	Future development	32
5	Conclusion	33
	References	34
	Appendices	
	Appendix 1. Create new POI page partially	
	Appendix 2. Edit POI page partially	



## List of Abbreviations

AR	Augmented Reality
POI	Points of Interest
3D	Three Dimensions
2D	Two Dimensions
SDK	Software Development Kit
ARML	Augmented Reality Markup Language
HTML	Hypertext Markup Language
KML	Keyhole Markup Language
VE	Virtual Environment
VR	Virtual Reality
URI	Uniform Resource Identifier
GPS	Global Positioning System
SQL	Structured Query Language



## 1 Introduction

The invention of smartphone has introduced lots of advanced and sophisticated technologies to ordinary users. Those augmented reality technologies that seemed far from reality, when they were seen on scientific fiction movies in the past, are now available on our mobile devices.

Today augmented reality technology mobile applications are growing at a rapid rate. The number of uploads are estimated to reach, 400million by the end of this year [1]. This shows how high the demand is and how promising it is to build a career related to developing augmented reality based applications.

The aim of this project is to provide a web application that enables users to publish their content on mobile augmented reality browsers regardless of their programming experience. Additionally, the other purpose is to demonstrate the possibility of using 3D content on augmented reality browsers. The web application was made using PHP programming language and MySQL-database. On the other hand, the demonstration part was developed using Wikitude ARchitect, which contains HTML, CSS and JavaScript web technologies. The application was developed for Finnish Architecture museum and it provides information about architectural background of buildings in Helsinki.

## 2 Overview

AR (Augmented Reality) is not a technology that came along with smartphones. However, mobile phone operating systems like iPhone OS X and Android has a big role on its recognition and popularity these days. [2, xxi] Before that, AR only appeared on sci-fi movies, and special military forces. Due to that, until recently, it was considered as “the future” technology [3, 1].

Before going any further, it is wise to define AR, and determine which sorts of applications are classified as AR applications? AR can be defined as the visual appearance that is created using computer generated data, such as images, audio, video and the real-time environment [4, 1]. In another words, AR is a means of interaction between computer and humans, humans and humans, and humans with computers [5, 1]. A better definition could be the following quotation:

AR is to mingle what is not really there with what is there as seamlessly as possible and to present users with an enhanced, or augmented, display of the world around them [6, xix]

This chapter highlights about the evolution of AR, which is from the very beginning until now. After that, this study glances the future expectation on the applications of AR and finally answers the question, why AR is chosen as a study topic.

### 2.1 Evolution of AR

As every human invention, AR also has come a long way to reach its current appearance. From the big bulky Sensorama, in 1957 we have now reached in the era where AR can be experienced in a small compact hand held device.

#### 2.1.1 History of AR

In 1957, Morton Heilig started building the first AR device called Sensorama (as shown in figure 1), and finished it in 1962. The device makes the user to feel the motion, visualize the event in 3D, and even sense the smell. [7; 8] The demo film of this device was, riding a bicycle on the streets of Brooklyn. This makes the user to feel as if he or

she was sitting in the actual bicycle. As a result, the user feels the vibration from the chair, the 3D view of Brooklyn streets and also the smell of the streets. [8]



Figure 1. Sensorama, the first AR machine [9].

Thereafter in 1992, Professor Tom Caudell, was the first who used the term AR for the first time. By that time he was doing research on simplifying the manufacturing process of aircraft for Boeing Aircraft Company. As a result, he made a very sophisticated software that simulate where certain aircraft parts should fit during manufacturing progression. [6, 3; 8]

It is impressive that, those technologies were actually implemented in the late 50s and 90s. No doubt, these technologies contribute a big part for today's achievement in AR.

### 2.1.2 AR today

One of the most impressive technologies in the field of AR nowadays is Google Glass (figure 2). Google showed a demo version of the device on 2012, during their google developers' annual conference. The device is operated by a voice command. It has a small screen Prism projector that displays images or video in a 64cm wide screen from 2.4cm distance. [10; 11; 12]

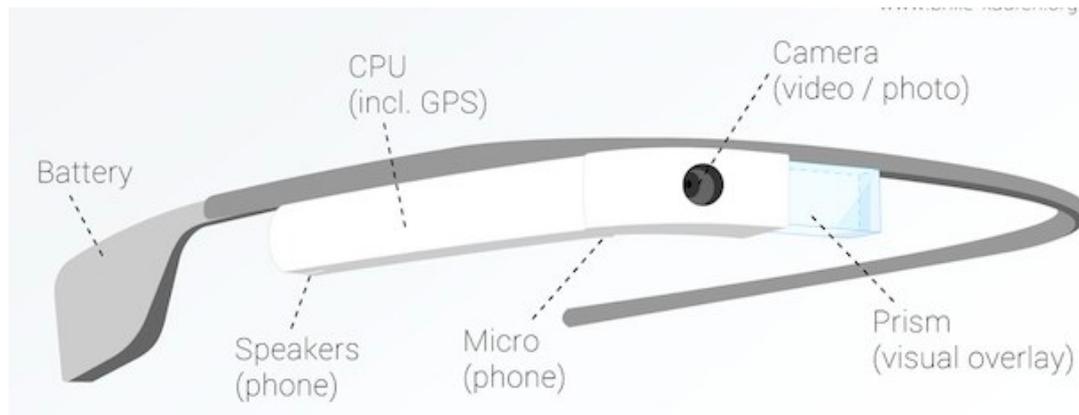


Figure 2. Google Glass [13]

Talking more about Google Glass would take us out of the scope of this thesis. However, it can be argued that it moved the AR achievement one step forward, which is why Time Magazine recognized Glass among the best inventions of the year 2012 [11].

### 2.1.3 AR in the future

It seems like the future is bright for AR developers and for AR technology. There are a number of fields of areas that would benefit from AR technologies such as, language translation, and bionic contact lens technology, which will be discussed in the study.

Language translation is one big issue these days. People travel often for business or pleasure to another country; however it is hard to interact sometimes due to language difference. Nowadays, AR technologies are solving that problem. Google has developed live audio translation for Android devices; at the same time iPhone also has similar application called Translate. These applications are at an early stage, and therefore they are not supporting live audio translate. However, studies to enable live audio and texts translating are in progress [2, 292].

It is believed that ionic contact lenses would be the future of AR technology. A Professor from Washington University, Babak Parviz along with researchers from Aalto University, are developing a contact lens that enables to display digital images, and texts over the real view [14]. One can imagine, how fascinating it would be if the outcome of this technology become reality.

## 2.2 Why AR?

This question can be described in two ways. The first one is from the developers' point of view, and the other way from the consumers' perspective.

The objective of a technology is basically measured by the amount of problems that it solves for the society or even for a single individual. AR seems to answer the needs of the society; as a result, the number of AR app users is increasing year by year. In 2014, it is estimated that 400million AR apps would be uploaded [1]. Furthermore, it is estimated that the revenue from mobile AR applications will reach \$5.2 billion by 2017 [14]. From these numbers one can tell how high the demand and the profit from AR applications are. Due to that, developers should take part in building the future AR applications.

## 2.3 Application areas of AR

The application area of AR is vast, but the major fields of applications are healthcare and medical science, learning and education, business, industrial, military, culture and entertainment business.

In 2010, the US Postal Service (USPS) has launched AR application that enables users to select the right size of box that suits for their package. All the users need is a computer with a webcam. First the user puts the package in front of the camera to the virtual simulator, and then he or she will able to choose the right size of box. This actually has a big contribution in saving the customers' time from waiting the usual long line in the post office. [3, 2-3]

Entertainment fields are some of the biggest fields of usage for AR applications. Today, commercials during sport activities are aided by AR technologies. The film industry, especially sci-fi has a long reputation in augmenting computer-generated objects in movies.

These days, media such as television broadcasts, are aided by AR applications. The classical green screen background no longer aids weather forecast programs. Instead, using AR technology images are captured in real time and the results are displayed on a 3D map. It can be noted from this how vast the application areas are. This gives developers multiple options to which field they should specifically develop applications.

### 2.3.1 AR applications on mobile devices

Ever since smartphones' market overtook that of personal computers, several software companies and developers are turning their face towards developing mobile applications. Similarly, the number of mobile AR applications is increasing rapidly. Considering the broadness of AR applications, it may not be possible to mention all of them. As an example a few top applications are, Wikitude Drive which is a navigation application where directions and driving instructions are fed over a live camera view; Wikipedia, YouTube, and Flickr AR applications, which indicate users Wikipedia info, YouTube videos, and Flickr images around them respectively. [2, 26-33]

### 3 AR browsers and 3D model

Just as web applications run on web browsers, the same way AR applications run on AR browsers. This chapter will cover first common terminologies used by the AR browsers field, and give a brief introduction of the three main browsers, which are Junaio, Layar and Wikitude. At the end, the final subtopic widely covers the main steps and classes needed in developing a Wikitude application with 3D contents.

#### 3.1 AR Browsers

Below is a list of terminology with definitions used in the study:

**Real View:** There are two ways to display the contents on AR browser. Displaying the content on a traditional map view or on real live camera view, the latter one is usually referred as real view.

**Virtual object:** The computer generated graphics, 3D object, video, audio, and texts in general. Any content that is shown on the screen except the real camera view are referred to as virtual object.

**Point of interest (POI):** This term is going to be used frequently in this thesis. POI is a single geographical location (Latitude, longitude, Altitude) where the output content is displayed. As can be seen from figure 3 the image is a real camera view, the info bubble shows POIs. Each point of interest has an icon, name and the distance from the user.

**Radar:** It is a circular object usually located on the top of the screen (Fig. 3). Each point of interest is represented by a dot on the radar. The radar shows the location of each POI and indicates which direction the user should point the device to the specific POI.

**Marker based AR:** Special markers are used to track the image, and then using image recognition software, the target image is identified. Thereafter, action such as overlaying 3D object, animation or displaying any content takes place. [2, 5]

**Markerless AR:** Unlike marker based AR, markerless ARs do not use special markers, rather they just display the AR content. [2, 9]

**Location based AR:** Figure 3 is a good example of this. POIs were located using LLA (Latitude, Longitude, Altitude), which is how the distances of the POI relative to the users location are identified. As can be seen, the distances of the POIs are mentioned in the bracket.

**Target:** It is an image and related data used for image recognition on an AR application.

**Tracker:** It is used to analyze or track a target image through a live camera view, and it is used for image recognition in an AR application.



Figure 3. POIs shown on a camera view [19]

Most AR browsers demand, which are run on a mobile, as a GPS sensor to identify the location of the user, compass for the user's direction and optionally camera to display the POI on the screen (as shown in figure 3).

The main benefits of using AR browsers are listed below:

- Can be programmed with less programming knowledge

- Less marketing needed since AR browser providers already promote their application
- Platform independency, meaning the same program will work on Android, iPhone and Windows phone without any modification. [2, 21]

Limitations or drawbacks of using AR browsers:

- No full control over the application.
- Users have to launch the provider's application first to view the content of the application. [2, 22]
- Technical problem on the provider will affect the application.

There are hundreds of AR browsers available these days, but with respect to functionality, only Junaio, Layar and Wikitude AR browsers are widely used.

### 3.1.1 Junaio

Junaio was first released in 2009, and its purpose was to enable users to share their 3D objects with friends, which is more likely as a social AR network. The company that released Junaio, Metaio, has a reputation on the field of AR. They were formerly have known to create image recognition solution. Even though, Junaio was released a bit later than the other major browsers, they still have lots of users.

The content build in Junaio is called channels. In one channel developers have the flexibility to create, simple POIs and for every POI can be augmented with 3D objects and animation, as well as location (latitude, longitude, and altitude). Exceptionally Junaio offers image recognition, which can overlay either 3D or video file. [2, 45]

### 3.1.2 Layar

Layar was developed in 2008 and it was the first AR to be released on the market. Because of its popularity, it is preinstalled on most mobile devices.

Applications built to run on Laya platforms are called layers. Developers can include on their layer the following functionalities.

- 3D objects: icons can be replaced with 3D objects.
- Event handlers: action when a user approaches a specific POI.
- Audio: audio play event handler can be triggered when a user is in a range to certain POI.
- Authentication request can be included to restrict the users of a layer. [2, 37]

### 3.1.3 Wikitude

Since Wikitude is used for the development of this thesis' application, more focus will be given on it. Wikitude was first released in 2008 for iPhone and Symbian devices [5, 26]. At the moment it runs on iPhone, Android and Blackberry.

Wikitude has a relatively simple but rich content compared with the previously mentioned AR browsers. For instance, anyone can develop Wikitude content (Wikitude contents are called worlds) using Google Maps interface without writing complicated code [2, 26]. However, that is not the only way to develop Wikitude world. By using the following methods, one can develop from simple to complicated worlds.

#### **Keyhole Markup Language (KML)**

KML is an XML based markup language that is basically used to display geographical data on Google Earth and Google Maps. [15] In addition to that, the content can be displayed on Wikitude AR browser.

The next code [15] shows a simple KML file, the first line indicates that it is a valid XML code. It shows that, the whole content inside the element `<Placemark></Placemark>`. The placemark element represents the POI. Every POI has name, description and point. Furthermore, the element "Point" has a sub element "Coordinates" which is the latitude, longitude and altitude of the POI.

```

<?xml version="1.0" encoding="UTF-8"?>
<kml xmlns="http://www.opengis.net/kml/2.2">  <Placemark>
  <name>Simple placemark</name>
  <description>Attached to the ground. Intelligently places itself
at the height of the underlying terrain.</description>
  <Point>
    <coordinates>-
122.0822035425683,37.42228990140251,0</coordinates>
  </Point>
</Placemark> </kml>

```

## Augmented Reality Markup Language (ARML)

Just like KML, ARML is also an XML-based schema developed by Wikitude. Since ARML is a subset of KML, it starts with <kml> tag. Some of the main tags are described in the table below.

Name	Required	Description
ar:provider	Yes	In this field, usually the organization or a person's website written as an attribute.
ar:name	Yes	The provider's name
wikitude:icon	Yes	A 32X32 pixel image that is going to be displayed on POI
name	Yes	The name of POI. NOTE: it is different from <i>ar:name</i> mentioned above, which is the name of the provider
Point	Yes	This field has the coordinate of the location, which are latitude, longitude and optionally altitude.

Table 1. Basic ARML tags [17]

### Web Services:

One of the great features of Wikitude is, that unlike in the other AR browsers, the contents are provided on Wikitude server. Once the developer published contents on KML or ARML format on Wikitude, the contents will be hosted on Wikitude server. However, if one wants to publish dynamic content on Wikitude, that can be done by using web service.

## **ARchitect**

In addition to ARML and KML, there is another way to create Wikitude world. Wikitude offers API (Application Programming Interface), which enables developers to create a world using HTML5 and JavaScript programming languages. Developing in ARchitect is more flexible and rich in content than KML and ARML, but it requires more programming skill.

Some of the features that are supported features are:

- 3D models
- Animation of 3D and/or 2D models
- Image recognition

Since one of the development parts of this thesis was done using ARchitect, the development workflow and API documentation will be discussed in detail on the next subtopic.

### 3.2 3D models and Wikitude

As discussed earlier, the only way to develop 3D supported Wikitude application is by using ARchitect world. ARchitect worlds are developed using Wikitude SDK.

#### 3.2.1 Wikitude SDK

Wikitude SDK is a software library, which enables developers to develop cross platform AR applications. The main idea is that by using web technologies like HTML5, CSS and JavaScript programs, one can develop an application that runs on Android, iPhone, Windows and Blackberry devices. That is why, nowadays, developers are turning their face towards cross platform applications rather than native applications.

## Scaling

Scale is a sensitive issue when it comes to AR applications. One of the main reasons is that the size of one object depends on the distance from the viewer. The more farther away the viewer moves from one object, the smaller the object looks. Due to that, Wikitude ARchitect uses *Scaled Distance Unit* (SDU) instead of pixel or any other size units. 1 SDU is defined as the size of an object with 1m height and 10m away from the viewer. [17]

The pictures below illustrate how to scale by using SDU. The height of the trackable object is always referred to as 1 SDU (figure 4). For instance if a circle with a radius of 0.25 SDU is put on the trackable 2D object, the circle will be displayed 0.24 unit radius with respect to the trackable object (figure 5).

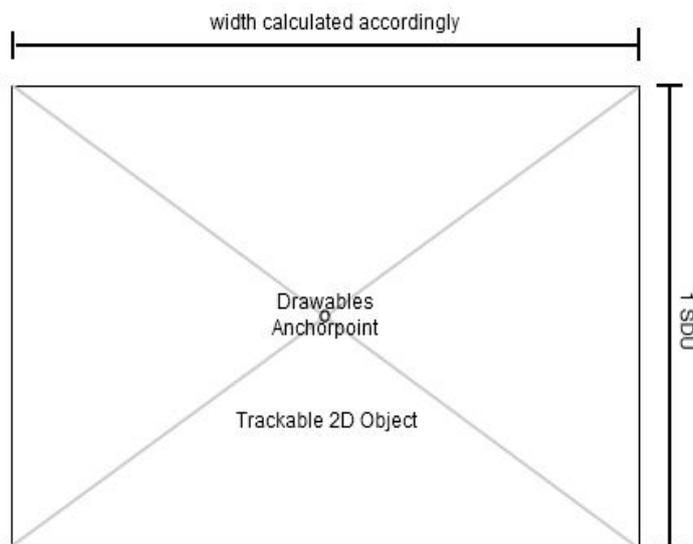


Figure 4. SDU

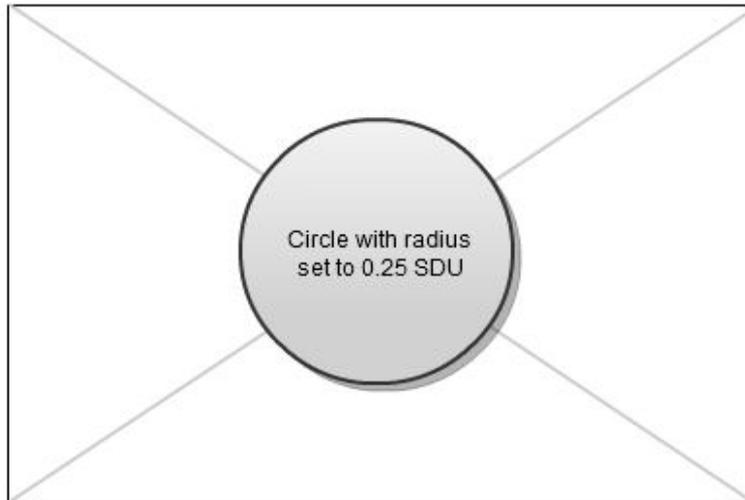


Figure 5. Circle scaled with SDU [17]

However, scaling with SDU also has drawbacks. That is, if the viewer is approaching towards the drawable object, it starts to get bigger and bigger and finally covers the screen view. In the other hand getting farther will make them to vanish from the screen. In order to avoid this, Wikitude ARchitect uses *Distance Based Scaling* (DBS). In DBS two values are defined as,

- *cutoff\_min* = 10m
- *cutoff\_max* = 20km

For the distance value less than *cutoff\_min*, the size of a drawable object will stay the same. On the same way, for a distance greater than *cutoff\_max* the object will not get smaller anymore. [17]

### 3.2.2 Basic APIs used for developing 3D model on Wikitude

The first step to develop by using Wikitude SDK is to be familiar with the Wikitude JavaScript API, provided on the Wikitude ARchitect documentation. The basic API will be described step by step.

**ARchitectObject:**

ARchitectObject is a superclass for all classes created in ARchitect. Due to that, the developer cannot call this class directly. ARchitectObject has a method *destroy()*, which the developer should call this method in order to free memory which is occupied by an already created object. This method can be used by all subclasses of ARchitectObject. [17]

ARchitectObject has the following subclasses under it

- ActionArea
  
- Animation
  
- Location
  
- Drawable
  
- ARObjct
  
- Sound
  
- ImageResource
  
- Tracker

The following picture (figure 6) demonstrated a class diagram of ARchitectObject.

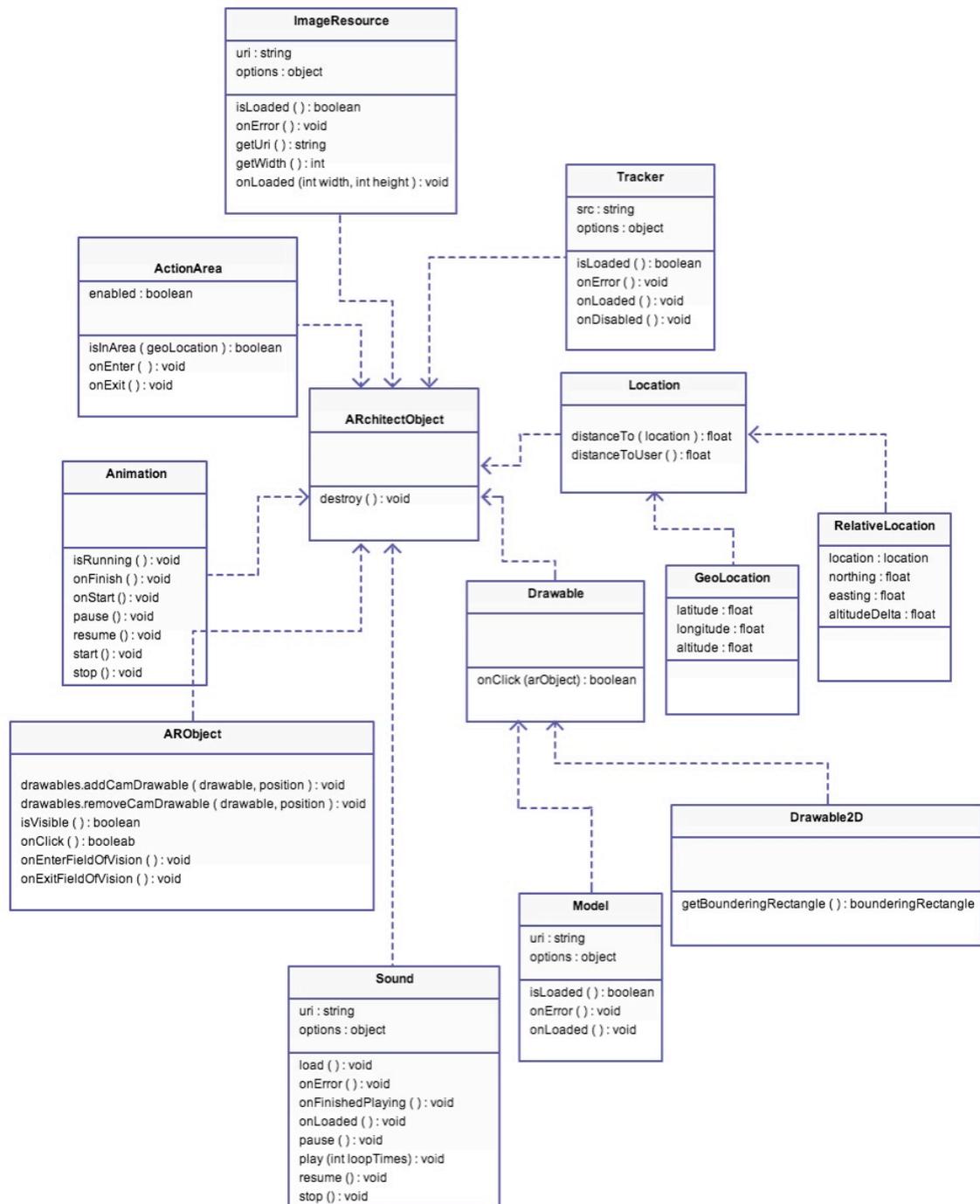


Figure 6. ARchitectObject class diagram.

One thing that should be noted about these classes is, since they are abstract classes, they must not be called directly [17]. The diagram in figure 6 shows the relationship and methods of subclasses under an ARchitectObject. These classes, methods, their subclasses, and a sample code examples will be discussed next in detail.

## Location

Location is used to define the geographical target location for the specified AR application. Location has two main subclasses. These are, *GeoLocation* and *RelativeLocation*. In *GeoLocation* a certain location represented with latitude, longitude and altitude. For example, it may require putting virtual objects (2D or 3D) in certain exact location. A JavaScript variable can be created like

```
var location = new AR.GeoLocation(60.77317, 23.069929, 0);
```

The other way to declare where a location is, relative to the application user's position or relative to the given location assigned in advance. Which is how far it is towards north and east. Relative location is defined as,

```
RelativeLocation (location, northing, easting, altitudeDelta)
```

If the location is relative to the application user (or the device), the value of attribute "location" will be null.

```
var location2 = new AR.RelativeLocation(null, -10, -20, -10);
```

In the above code, location2 was assigned 10m towards south, 20m towards west and 10m down relative to the device.

## ActionArea

ActionArea is a two dimensional geographical area to trigger some action on enter and exit of that specific area. Which is, when the user enters and leaves the action area `onEnter()` and `onExit()` will be executed respectively. In addition, it has a boolean method `isInArea()` which checks if the user is in the action area or not.

```
var location = new AR.GeoLocation(60.77317, 23.069929);
```

```

var actionRange = new AR.ActionRange(location, 500
onEnter : function() {
    //content of the function
});

```

ActionArea has a subclass called *ActionRange*, which defines a circular area or range for a given location. ActionRange (location, radius, options), where radius is given in SDU (500 SDU in the previous code), and optionally functions can be called inside action range as shown in the previous code.

## Animation

As its name describes, this class is used to animate 2D and 3D objects. Animation is an abstract class, and has two subclasses. These are, AnimationGroup and PropertyAnimation. GroupAnimation is used to run group of animations either in sequence or parallel. On the other hand, Property Animation creates animation by enabling changes to a numeric value/property of an object. Animation also has seven helpful methods.

`isRunning()` : This boolean method checks if the animation is running or not.

`inStart()` : A void method, and it is fired when the animation starts.

`onFinish()` : Also void method will fire when the animation is done, all functions that run right after finishing animation should go in this method.

`pause()` : This makes to pause the animation, when animation is on progress, otherwise this method has no effect.

`resume()` : If the animation is resumed, this method enables to continue the animation from the paused point.

`start()` : To start the animation.

`stop()` : To stop the animation.

## Drawable

Drawable class is responsible for representing any graphical data on a coordinate. The coordinate system is expressed as a right handed system, which is x-axis always to the right, y-axis to up and z-axis points out of the screen as shown on figure 7.

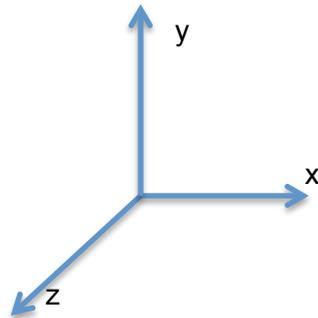


Figure 7. Right handed coordinate system

In the case of GeoObject (see ARObject) up and right are considered on to earth's surface. On the other hand, if trackable 2D object is used, directions are used relative to the tracked image. Drawable has two subclasses called Drawable2d and Model.

Drawable2d class is used for representing 2D objects, and it has Label, Circle, ImageDrawable and HtmlDrawable subclasses. These subclasses are used to represent labels, circle drawings, images and html tags respectively. For example, the following one line code can be used to draw the circle with a radius 0.25 SDU (shown on figure 5).

```
var circle = new AR.Circle(0.25);
```

Similarly using Model subclass 3D models are represented. However, the 3D models used in Wikitude should be in .wt3 format (the conversion method explained in section 3.2.3)

## ARObject

This class has methods and properties, which are needed to display 2D and 3D objects in AR scene. There are two ways that enable to display these objects. The first one is locating an object in certain geo location, which in this case ARObject's subclass *GeoObject* is used.

```
GeoObject (location, option)
```

GeoObject has location and optional properties such as enabled, renderingOrder, onEnterFieldOfVision, onExitFieldOfVision, onClick, drawables.cam as a parameter. [9] To support with example, ARObject will be created for previously assigned variable location.

```
var geoObject = new AR.GeoObject(location);
```

Or with properties included,

```
geoObject = new AR.GeoObject (location,
    {
        drawables : {
            cam : drawable1
        }
    });
```

The previous code demonstrates, how to assign an object at a certain location and adding a drawable to the object that can be displayed through the cam view.

Another way of displaying objects is by using trackable 2D objects. In this case, the subclass of ARObject *Trackable2dObject* is used.

```
Trackable2dObject (tracker, targetName, options)
```

Theses three parameters, tracker is the path of the target, targetName is the name of the specific target in the tracker and same property options mentioned on GeoObject. As an illustration, the circle created on drawable section can be displayed with the following code.

```
var tracker = new AR.Tracker("http://URI/tracker.zip");
var trackable2DObject = new AR.Trackable2DObject(tracker, "circITr",
{
    drawables : {
        cam : circle
    }
});
```

What this code does is, first the tracker URL is assigned. On the next line of the code, from the tracker zipped file gets a tracker called circITr and if the tracker finds an image similar to circITr in the camera view, it will draw the circle as shown on figure 5.

## Sound

Wikitude SDK has a specific class dedicated to sound files. This class enables audio files to be used in AR applications. Class Sound has a property called “*state*”; which at a given instance, sound has exactly one of the following states.

- INITIALIZED
- LOADING
- LOADED
- PLAYING
- PAUSED
- ERROR

First, sound instance has INITIALIZED state when the “sound” object is created using the code shown after this paragraph. After initialization, there are various methods used to change its state.

```
var sound = new AR.Sound("http://URI/audio.mp3");
```

*load()* : a void method to load the initiated sound on a device. Calling this method changes the state of sound from INITIALIZED to LOADING, meanwhile after the loading is finished the state changes to LOADED.

*onLoaded()* : this method can be used to add some action after the audio file is loaded. For instance, to play the loaded audio file right after the file is loaded.

The rest of methods, examples, and a lot more subclasses can be found on Wikitude SDK documentation page [17].

### 3.2.3 Wikitude 3D encoder

Wikitude does not directly take 3D models. Instead the models should be converted to Wikitude 3D format using Wikitude 3D encoder. This encoder application can be easily downloaded from Wikitude website and installed in both Mac OS X and Windows platforms. This conversion has a number of advantages, first it compresses the file and mainly optimizes for mobile phone usage.

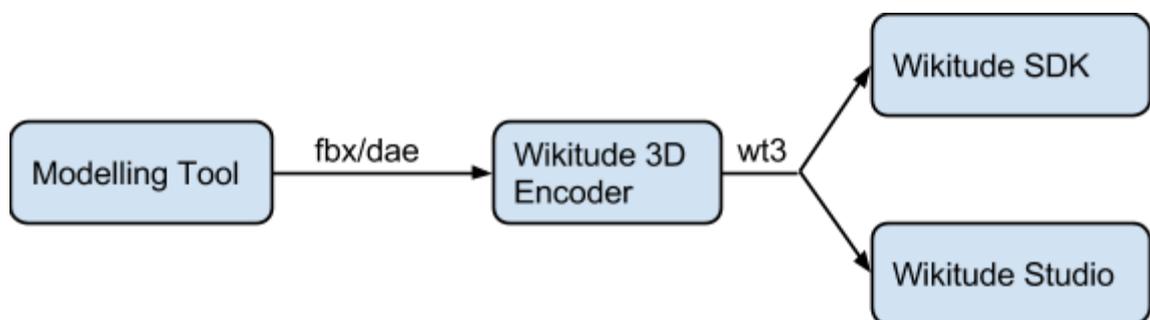


Figure 8. 3D model workflow [18]

The encoder only imports .fbx and .dae file formats. The recommended 3D modeling tools are, Autodesk Maya for .fbx files or blender for .dae files. After designing the required 3D model, the next step would be to convert to wt3 format using Wikitude 3D Encoder (as shown in figure 8). The encoded file could either be used on Wikitude SDK or Wikitude Studio. Using the SDK requires programming skill and understanding of previously discussed classes, but more flexibility and animation are also supported.

Nevertheless, using the Studio is easy to use and does not require any programming skills but animation is, however, not supported. [18]

### 3.2.4 Problems and Known Issues

Even though the codes are correct sometimes, the application might not display the 3D file, which is supposed to be displayed. This mainly happens due to fragmented 3D files. There are good practices to avoid some problems regarding 3D models, and yet there are still some issues pointed out by Wikitude, which hopefully will be solved in the near future.

Models with multiple joint might get distorted; this can be resolved by attaching all joints to a master joint. Another problem is that 3D models that showed and seemed working on Wikitude Encoder might disappear on devices. The recommended practice for this problem is to use as small texture size as possible. [19]

As Wikitude informed, baking animations (merging) and exporting as .dae file in blender does not work. In that case animations are not being able to import on during decoding on Wikitude Encoder. Another harmless issue is models with animation exported from blender, as FBX seems to show extra animation for camera, light and motion paths. [19]

## 4 Design and Implementation

This chapter is all about the practical work of this thesis. Development workflow, test results, problems, benefits and materials and software used will be explained and described in this chapter.

### 4.1 Methods and materials

The practical part of this thesis consists of two parts. The first part covers developing a content management system for Museum of Finnish Architecture's ARML based Wikitude application. The second part goes through the process of making a demo

application to show how 3D objects are displayed on Wikitude AR browser. To develop the application, the following programs have been used.

- HTML5
- Bootstrap CSS to make the web application responsive
- ARchitect JavaScript library
- ARML
- PHP as a backend language
- Sql database

Tools and devices used for programming and testing are

- Windows and MacBook computers
- Sublime Text 2 (Text editor on Mac)
- Notepad++ (Text editor on Windows)
- WinSCP and FileZila FTPs (File Transfer Protocols)
- Smartphones (Samsung and iPhone) for testing the application
- Samsung tablet and iPad for testing the application
- Adobe Photoshop

Besides these, books, online materials, instructors' notes, lectures and images provided by the customer have been used for both theoretical and practical part of this thesis.

#### 4.2 Development of ARML based content management system

The main idea of the application is to publish dynamically updatable data into Wikitude AR browser. This was carried out using web service.

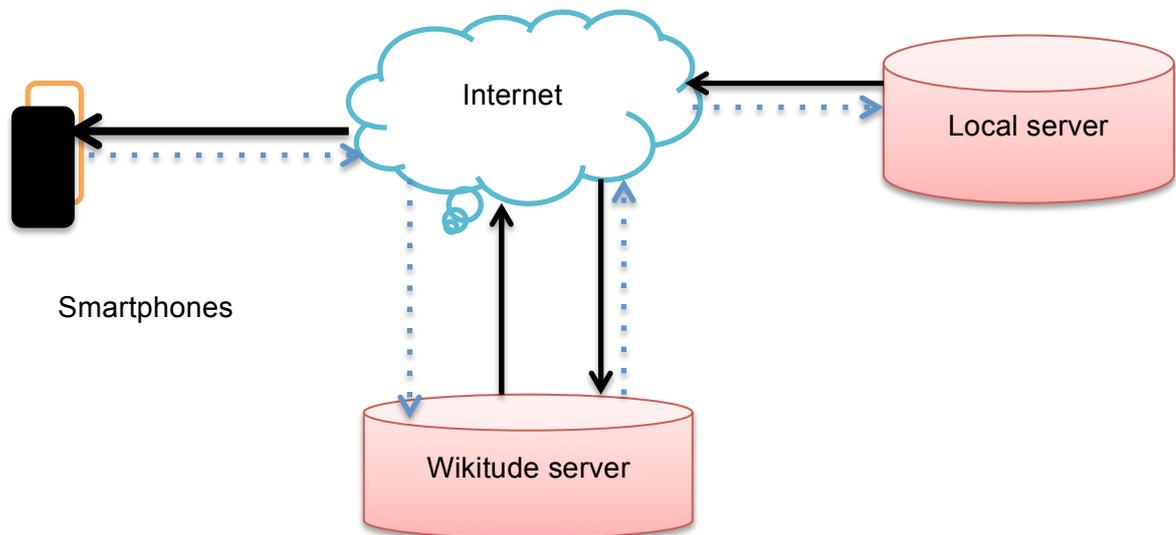


Figure 9. Application cycle diagram.

As illustrated in figure 9, first the user sends a request via smartphone using Wikitude AR browser (NOTE: the dotted line arrows show the path of user's request and the dark lines represent the respond of a web service). Then the request reaches to Wikitude server via Internet, and the Wikitude server sends a request to the web service. The web service fetches the data from the local server (in this case Museum of Finnish Architecture's server). This data is returned to Wikitude server as ARML file, and finally Wikitude displays the content on the user's device.

#### 4.2.1 Data management

This web service was primarily adopted and modified from a previously developed thesis project by Annina Alamäki [20]. The first step was to build the content management form. The form was created by PHP and it enables a user with administrator password (figure 10) to create and modify POI data. All the POI data are stored in MySQL database. The application is available in four languages, Finnish, English, Swedish and Russian. Each language data is stored in their corresponding table in the database, which means that four database tables were created.



The image shows a login dialog box with a light gray background. At the top, it says "Your password will be sent unencrypted." Below this, there are two input fields: "Name:" followed by a white text box with a blue border, and "Password:" followed by a white password box with a blue border. Underneath the password field is a checkbox labeled "Remember this password in my keychain". At the bottom right, there are two buttons: a white "Cancel" button and a blue "Log In" button.

Figure 10. Authentication is requested during accessing.

When using the web service, the first stop will be authentication. The user is requested to login with credential information. Currently there is no “create a new password” feature, so the credential information is pre-generated.

```
if (($username == $_SERVER['PHP_AUTH_USER']) && ($password ==
md5($_SERVER['PHP_AUTH_PW']))) {
    $authorized = TRUE;
    break;
}
```

The pre-generated password is encrypted with MD5 (Message-digest algorithm). The above PHP code checks the validity of user’s username and password entered when evaluating with the encrypted file.

#	Name	Type	Collation	Attributes	Null	Default
<input type="checkbox"/>	1 <b>ID</b>	bigint(9)			No	None
<input type="checkbox"/>	2 <b>provider</b>	varchar(100)	utf8_general_ci		No	http://www.mfa.fi/
<input type="checkbox"/>	3 <b>providername</b>	varchar(50)	utf8_general_ci		No	Arkkitehtuurimuseo
<input type="checkbox"/>	4 <b>prodesc</b>	varchar(250)	utf8_general_ci		No	Arkkitehtuurimuseo / Museum of Finnish Archi
<input type="checkbox"/>	5 <b>providerurl</b>	varchar(250)	utf8_general_ci		No	http://www.mfa.fi/
<input type="checkbox"/>	6 <b>tags</b>	varchar(100)	utf8_general_ci		No	guide, tourist, Arkkitehtuurimuseo, Architecture
<input type="checkbox"/>	7 <b>wikilogo</b>	varchar(255)	utf8_general_ci		No	http://dev.mw.metropolia.fi/wubushek/img/logo
<input type="checkbox"/>	8 <b>wikiicon</b>	varchar(250)	utf8_general_ci		No	http://dev.mw.metropolia.fi/wubushek/img/logo
<input type="checkbox"/>	9 <b>lat</b>	decimal(20,8)			No	None
<input type="checkbox"/>	10 <b>lng</b>	decimal(20,8)			No	None
<input type="checkbox"/>	11 <b>thumbnail</b>	varchar(250)	utf8_general_ci		No	None
<input type="checkbox"/>	12 <b>hiResImageUrl</b>	varchar(250)	utf8_general_ci		Yes	NULL
<input type="checkbox"/>	13 <b>hiResImageUrl2</b>	varchar(250)	utf8_general_ci		Yes	NULL
<input type="checkbox"/>	14 <b>hiResImageUrl3</b>	varchar(250)	utf8_general_ci		Yes	NULL

Figure 11. Rows of database table partially.

As the customer's (Architecture Museum of Finland) request, each data tables have forty-three fields (figure 11). Id, provider, provider name, provider's description, provider's URL, tags, provider logo and icons are provided by default, and users is not able to modify these fields.

The table is designed so that each POI has up to seven image attachments, six other types of attachments (audio, video, and pdf), latitude, longitude, name, email, address, thumbnail and icon. To create a new POI, at least name, latitude and longitude fields are required. Since all fields are displayed and called at the same time, dividing the data fields into pieces (normalization) was not needed.

# Arkkitehtuurimuseo

name		
Central Railway Station	<a href="#">Edit</a>	<a href="#">Remove</a>
Central Post Office	<a href="#">Edit</a>	<a href="#">Remove</a>
National Museum	<a href="#">Edit</a>	<a href="#">Remove</a>
Kallio Church	<a href="#">Edit</a>	<a href="#">Remove</a>
Hotel Tomi	<a href="#">Edit</a>	<a href="#">Remove</a>
Hanasaari Power Station	<a href="#">Edit</a>	<a href="#">Remove</a>

[Add a new POI](#)      Arkkitehtuurimuseo by Metropolia ©

Figure 12. List of POI on the application

The administrator is entitled to create a new POI, and to edit or delete existing POIs. Each Person's name is displayed in a table with the option to edit or remove from the table (figure 12). During removing a POI, the user is requested to confirm removing via a popup window. If the user confirms the delete process, the POI will be deleted permanently from the database and it will not be displayed on Wikitude world either. On the other hand, when adding a new POI (Appendix 1), the content is directly stored on database and displayed on Wikitude world with corresponding location.

## 4.2.2 Testing

Devices	Results
iPhone	Description part missing on detail view, but the rest works well (see figure 9).
iPad	Description part missing on detail view, but the rest works well.
Samsung Galaxy Tablet/ Samsung Galaxy S 4	All contents are displayed, but it doesn't play the audios and videos.

Table 2. Test result for AR application

Testing was made both for the content management and AR application. Some problems had been discovered during testing of AR application as described in table 2. To figure out the faults found out during testing, Wikitude technical support was contacted. A person from the technical support informed that, audio playing from attachment is disabled in some Android devices; according to his suggestion, using ARchitect instead of ARML would solve the problem. Besides that, the application works quite well. Figure 13 shows the output of the application (world) on Wikitude AR browser. The application can be accessed from Wikitude AR browser under the category Education. When tapping the application icon, available POIs on their geographical location are displayed (picture on the left). When tapping one of the POIs, a popup window with name and description appears (middle picture). Finally the detailed view with images appears when tapping the popup display.

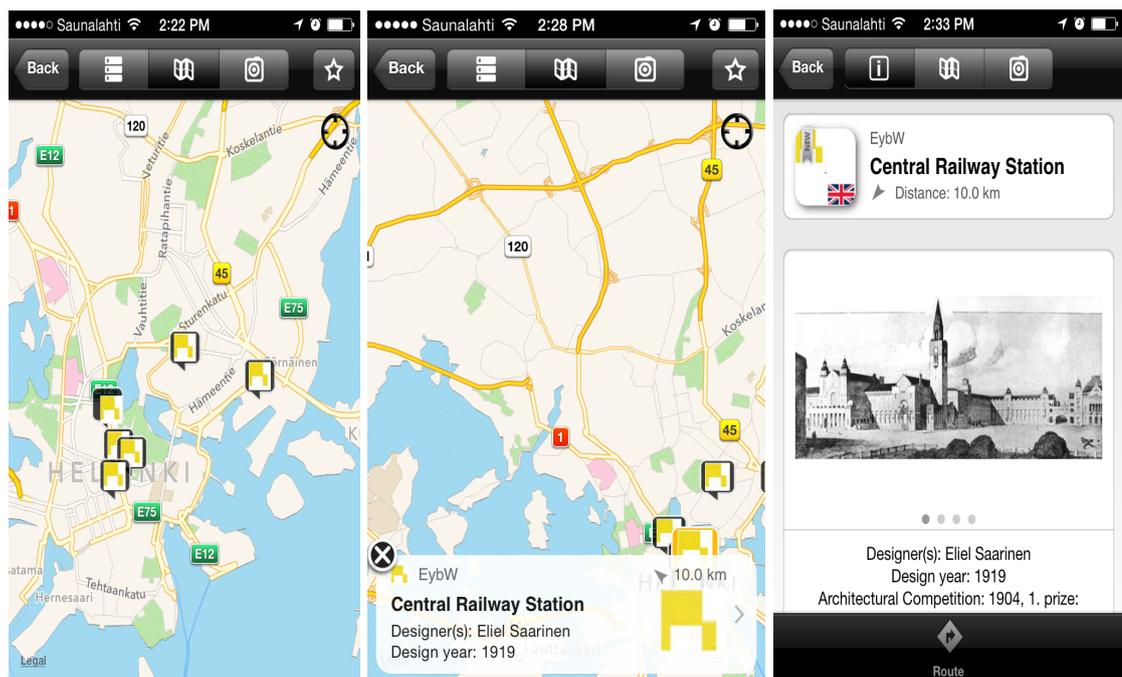


Figure 13. The output of the Arkkitehtuurimuseo world on iPhone.

Browsers	Status
Google Chrome	Works
Safari	Works
Mozilla Firefox	Works
Internet Explorer	Works
Samsung Galaxy S4/ Tablet	Works

Table 3. Test results for content management web application

As illustrated in table 3, the test for the content management system has positive results.

#### 4.3 Demonstration of 3D models on Wikitude AR browser

The objective of this project is to test how to execute 3D models on Wikitude browser. The main initiative idea is to view the model of buildings or towers, which existed previously but not anymore, and/or to view the model of a soon to be built building in their actual geographical location.

Primarily, the 3D model should be encoded to Wikitude format wt3 by using Wikitude 3D encoder, as mentioned in the previous chapter. After that, html file is created and architect.js included inside the head tag.

```
var tower = new AR.Model("tower.wt3",{
    scale:{
        x:0.1,y:0.1,z:0.1
    }
});
```

The above line of code is used to represent the 3D model assigned as "tower.wt3". One thing to be noted here is that the scale plays a big role on the visibility of the 3D model. The scale should not be too small or too big, but in order to lower the loading time, using an appropriate minimum size is advised.

```
//Code for geo location
var location = new AR.GeoLocation(60.77317, 23.069929); (a)

//Code for relative location.

var location = new AR.RelativeLocation(null, -10, -20, -10); (b)
```

The next question is, where to display the model. There are two options to display the model, as shown in the above code. The first one is displaying it at a certain geographical location (code “a”), and alternatively, location can be assigned relative to the user’s position (code “b”). For the demonstration purpose, relative location was used.

```
geoObject = new AR.GeoObject (location,
    {
        drawables : {
            cam : tower
        }
    });
```

Finally, the model is ready to be displayed on a camera view. The above code is used to display the 3D model on the specified location in AR camera view. The objective of this demonstration is to test the compatibility of Wikitude AR browser with 3D objects. Since the result was positive, further AR reality applications with 3D content could be developed using Wikitude SDK.

#### 4.4 Benefits of the application

The main functionality of this application is to offer required information about different locations via Wikitude AR browser. Since once the web service is installed; the application can be developed with low cost and less or no understanding of programming languages. It can be used by small and large scaled sectors, startup companies with low budget, nongovernmental organizations and even individuals; and it can serve as a promotional, recreational, educational, or awareness broadcasting channel.

#### 4.5 Limitations

Even though the developed ARML based world is fully functional, there are a few limitations that should be noted,

- POIs are displayed location based, that means the farther away the user is, the less chance there is to access the data on POIs.
- If the provider service (in this case Wikitude) is down, contents cannot be displayed.
- Users are required to download the provider's (Wikitude) application in order to access the application.
- Only users with smartphone and Internet connection can access the contents.

However, since AR is a rapidly growing technology, the limitations and drawbacks will hopefully be improved in a short period of time.

#### 4.6 Future development

The future development for this project might be to have an integrated AR application, which contains both 3D objects and POIs done in two sections. This can be carried out in two ways. One option is using Wikitude ARchitect world, which is one of the previously mentioned APIs.

The other option would be the release of ARML 2.0, it has currently most functionalities using ARchitect world including 3D support and animation. The release will be good news for AR application developers, since it eases the developing process and furthermore, it has multiple functionalities.

## **5 Conclusion**

The aim of this project was to provide a web application that enables users to publish their content on mobile augmented reality browsers regardless of their programming experience. As a result, a content management web application was developed using PHP programming language and MySQL database. The final product was delivered to the client and used to publish their content on Wikitude AR browser.

No significant problems were encountered during the development phase of the project. However, due to fast growth rate of AR technology, finding up to date reference materials was a problem. Therefore, the majority of reference materials used for this thesis are online resources.

It should be acknowledged that the continuous follow up and support both from the instructor and the client has been a catalyst in speeding up the development period of this project.

## References

- 1 Mobile AR Content Downloads to Top 400million by 2014 [Online] Softpedia  
URL: <http://news.softpedia.com/news/Mobile-AR-Content-Downloads-to-Top-400-Million-by-2014-131196.shtml>. Accessed 4 April 2014.
- 2 Lester M. Professional Augmented Reality Browsers for Smart Phones, Programming for Junaio layer and Wikitude. West Sussex: John Wiley & Sons Ltd; 2011.
- 3 Roche K. Pro iOS 5 Augmented Reality. New York, NY: Apress; 2011.
- 4 Kipper G., Rampolla J. Augmented Reality, An Emerging Technology Guide to AR. Waltham MA: Elsevier Inc., 2013.
- 5 Craig A. Understanding Augmented Reality, Concepts and Applications. Waltham, MA: Elsevier Inc.; 2013.
- 6 Mullen T. Prototyping Augmented Reality. Indianapolis, Indiana: John Wiley & Sons Inc.; 2011.
- 7 Sensorama Simulator Patent [Online] Google Inc.  
URL: <http://www.google.com/patents/US3050870>. Accessed 5 April 2014.
- 8 Sung D. The History of Augmented Reality; 1 March 2011 [Online] Pocket-Init  
URL: <http://www.pocket-lint.com/news/108888-the-history-of-augmented-reality>  
Accessed 5 April 2014.
- 9 Inventor in the Field of Virtual Reality [Online] Mortonheilig.com  
URL: <http://www.mortonheilig.com/InventorVR.html>. Accessed 5 April 2014.
- 10 What does Glass do? [Online] Google Inc.  
URL: <https://sites.google.com/site/glasscomms/whatdoesglassdo>. Accessed 5 April 2014.
- 11 Best Inventions of Year 2012 [Online] Time Inc.  
URL: <http://techland.time.com/2012/11/01/best-inventions-of-the-year-2012/slide/all/>. Accessed 5 April 2014.
- 12 Project Glass: Live Demo at Google I/O; 1 June 2012 [Online] Google developers  
URL: <https://www.youtube.com/watch?v=D7TB8b2t3QE>. Accessed 5 April 2014.
- 13 Die Google Brille (Infografik) [Online] Brillen & Sehhilfen  
URL: <http://www.brillen-sehhilfen.de/google-brille/>. Accessed 6 April 2014.
- 14 Layar, Facts and Figures; August 2013 [Online] Layar  
URL: [http://static.layar.com/web\\_documents/Facts\\_Figures\\_Layar\\_August\\_2013.pdf](http://static.layar.com/web_documents/Facts_Figures_Layar_August_2013.pdf). Accessed 10 April 2014.
- 15 The Case for Non-Visual Augmented Reality and a Multi-Modal Approach [Online] Reality Augmented

- URL: <http://www.realityaugmentedblog.com/2012/07/the-case-for-non-visual-augmented-reality-and-the-multi-modal-approach/>. Accessed 5 April 2014.
- 16 Keyhole Markup Language [Online] Google Developers  
URL: [https://developers.google.com/kml/documentation/kml\\_tut](https://developers.google.com/kml/documentation/kml_tut). Accessed 5 April 2014.
  - 17 ARML 1.0 Specification for Wikitude [Online]. Openarml.org  
URL: <http://openarml.org/wikitude4.html>. Accessed 5 April 2014.
  - 18 Wikitude ARrchitect v3.1 API Documentation [Online] Wikitude Gmbh  
URL: [http://www.wikitude.com/external/doc/alr/module\\_AR.html](http://www.wikitude.com/external/doc/alr/module_AR.html). Accessed 5 April 2014.
  - 19 3D Models, Animations and the Wikitude 3D encoder [Online] Wikitude Gmbh  
URL: [http://www.wikitude.com/developer/news/-/blogs/3d-models-animations-and-the-wikitude-3d-encoder?\\_33\\_redirect=http%3A%2F%2Fwww.wikitude.com%2Fdeveloper%2Fnews%3Fp\\_p\\_id%3D33%26p\\_p\\_lifecycle%3D0%26p\\_p\\_state%3Dnormal%26p\\_p\\_mode%3Dview%26p\\_p\\_col\\_id%3Dcolumn-1%26p\\_p\\_col\\_count%3D1](http://www.wikitude.com/developer/news/-/blogs/3d-models-animations-and-the-wikitude-3d-encoder?_33_redirect=http%3A%2F%2Fwww.wikitude.com%2Fdeveloper%2Fnews%3Fp_p_id%3D33%26p_p_lifecycle%3D0%26p_p_state%3Dnormal%26p_p_mode%3Dview%26p_p_col_id%3Dcolumn-1%26p_p_col_count%3D1). Accessed 5 April 2014.
  - 20 Alamäki A. Augmented Reality Web-application for Mobile Environment. Helsinki Metropolia University of Applied Science; 2011.

Create new POI page partially

## Arkkitehtuurimuseo

---

### Arkkitehtuurimuseo Admin - English version

[Back](#)

**Name:**

**Designer(s):**

**Design year:**

**Credit:**

Arkkitehtuurimuseo by Metropolia ©

**Credit:**

**Architectural Competetion:**

**Height:**

**Style Defination:**

**Description:**

\* max. 2500 characters. Please avoid special characters. check more instructions in the manual.

Arkkitehtuurimuseo by Metropolia ©

**Latitude:**

## Edit POI page partially

# Arkkitehtuurimuseo

---

### Edit your POI

**Name:**

\*

**Designer(s):**

**Design Year:**

**Credit:**

**Architectural Competition**

Arkkitehtuurimuseo by Metropolia ©

**Style Definition:****Description:**

\* max. 2500 characters. Please avoid special characters. check more instructions in the manual.

In the 1920s, the United States was an important source of inspiration for Finnish artists. Eliel Saarinen, one of the most famous Finnish architects, moved permanently to the United States in 1923 following his success in the Chicago Tribune skyscraper competition. It is no surprise, then, that there was such demand for a skyscraper to symbolise the modernity of the young nation's capital city. However, the recession of the early 1930s destroyed such fanciful ideas and Hotel Tornii was the only skyscraper to be built. Although it was the highest building in Finland when it was completed, its 14

**Latitude:**

\* the exact latitude you will find out with the help of google maps. **Use always dots ".", no commas ","!** check more instructions in the manual.

**Longitude:**

\* the exact longitude you will find out with the help of google maps. **Use always dots ".", no commas ","!** check more instructions in the manual.

Arkkitehtuurimuseo by Metropolia ©

**Thumbnail:**\* please insert the correct thumbnail address for the poi.