



# Table of Contents

## [1. Introduction: 3D Models for Various Purposes](#)

## [2. Application Use Case Driven Analysis](#)

### [2.1 Viewpoint: Aerial Bird's Eye View](#)

### [2.2 Viewpoint: Street Level, 1st or 3rd Person](#)

### [2.3. Viewpoint: Virtual Reality \(VR\) Use](#)

### [2.3 Sketching of building masses in an area plan](#)

### [2.4 Illustrate a plan with identifiable semantics: planned services \(e.g. a cafe\), but not building designs](#)

### [2.7 Detailed Models: Photorealistic and Construction models](#)

### [2.8 City models](#)

### [2.9 Terrain / heightmaps and ground texturing](#)

### [2.10 Environment, lighting and materials](#)

### [2.11 Audio](#)

### [2.12 Interactivity](#)

#### [Interactable parts](#)

## [3. Technical restrictions and solutions](#)

### [3.1 Realtime 3D](#)

#### [Level of Detail](#)

### [3.2 Web publishing](#)

## [4. Modeling recommendations](#)

### [4.1. Identify the needs: What Questions to Ask](#)

### [4.2 Case: VirtualCampus](#)

## [REFERENCES](#)

# 1. Introduction: 3D Models for Various Purposes



3D models are typically created for a specific purpose. In games, buildings drawn at the background in a flight simulator are very different from the ones depicted in a first person shooter. Both the end result and the process to create the models vary greatly. In non-game (non-entertainment) applications a sketchy buildings quickly drafted with SketchUp are very different from detailed construction and e.g. electrical system designs in construction models. Depending on the field and use case features such as amount and style of detail, granularity of the objects (grouping, partitioning), nature and style of materials, how the shapes are constructed and many other aspects of the 3D models can differ. Notably also the aesthetic styles differ: depending on what the model needs to communicate to its viewers, the *desired* look can for example require **less** detail than what would be technically feasible.

Here we strive to find a constructive approach to somewhat generic purpose 3D (city) modeling. In the following chapters we analyze in more detail the requirements from various use cases. We also pursue to identify common and differing aspects of both the process and end results.

## 2. Application Use Case Driven Analysis

In this section we analyze 3D city models in various use cases. While much of the discussion is on visualization, we strive to cover essential elements of *functionalities* in different 3D applications from games to building information models (BIM). Key dimensions in the analysis are:

- viewpoint (aerial, on the ground, indoors/outdoors etc.)
- interactions (pickability, animation ability etc)

- style / amount of detail (texturing, granularity, level of detail)

In this document the emphasis is on city modeling and city planning oriented use, but we also consider game like applications.

## 2.1 Viewpoint: Aerial Bird's Eye View

This perspective is very common for map based visualisation of cityscapes.

- Can be quick to load and open fast: no textures, colours with vertex colours, does not require lot of memory. An example in VirtualOulu (original release 2014) opening view in Figure 1 below.
- Granularity per building, block, or even large area (batching in systems like ViziCities and Mapgets)
- Similar to Google Maps Earth and other aerial scan based rough models



**Figure 1.** VirtualOulu original release light start view (~7MB download), on Meshmoon (the low detail modeling by Ludocraft and hosting and app development work by Adminotech).

## 2.2 Viewpoint: Street Level, 1st or 3rd Person

Dropping the camera down to a street level requires more detailing not just on the terrain, but also the building walls. Street level view depending on use case requires:

- Requires much more detail, especially for immersion
- Counterside: not everything fits to memory (e.g. gfx memory) nor can be rendered, therefore, for instance, scene management, memory management, culling is needed
- Similar to FPS / RPG games
- Example: VirtualOulu street level (Figure 2 below)
  - all surfaces have texture images



**Figure 2.** The terrace chairs in front of the restaurant are individual objects -- allows ease development of features such as human characters being able to sit on them. This view is from low detail version of VirtualOulu.

We created an optimized version of the original 9 city blocks for VirtualOulu to be used in close bird's eye view and used in Service Fusion concept (Hickey 2012) -- but has been used as a low poly version on a street level as well (for example in Playsign's car driving demo / publishing a city model on a web in an early FIWARE test, and later in Historical Oulu prototype etc. at the University of Oulu)



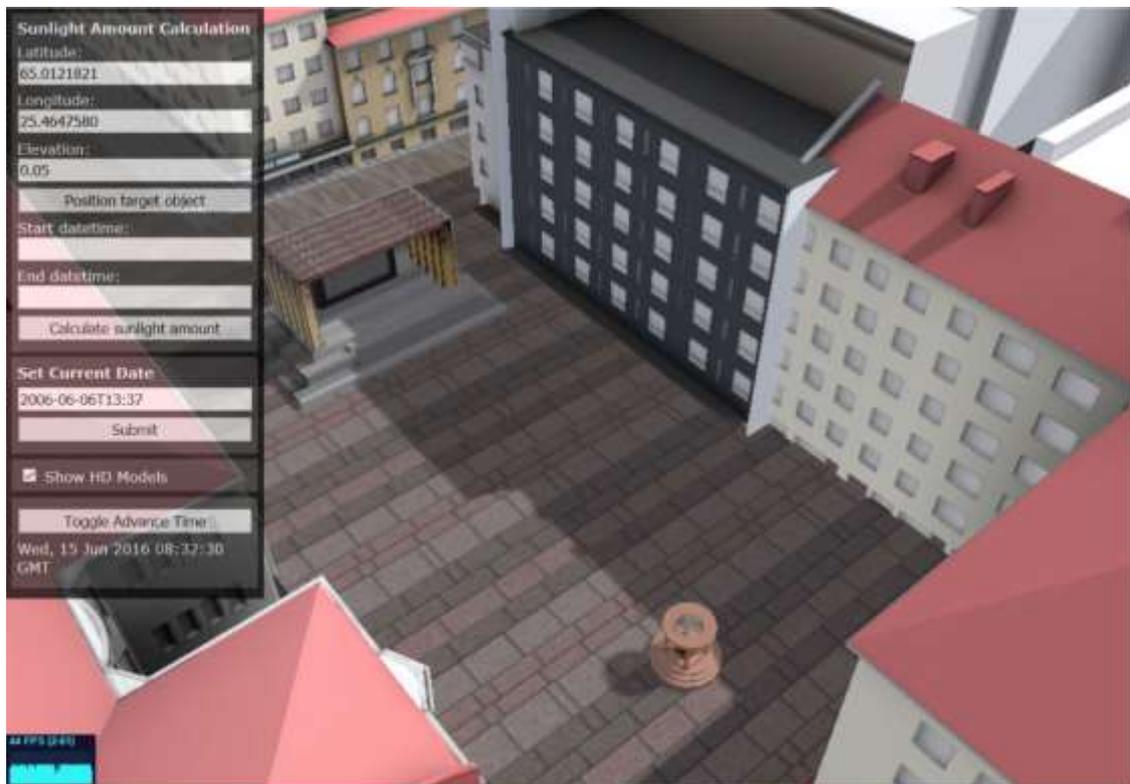
**Figure 3.** Very light and small version of the 9 blocks model, here used in <http://playsign.github.io/OuluThreeJS/>

### 2.3. Viewpoint: Virtual Reality (VR) Use

The concept of VR has been recently adopted to mean immersive, high definition and first person experience -- the kind of you get with Oculus, Vive head-mounted displays. Use of VR

use sets additional challenges for 3D content: it may need to be both very detailed and realistic, but also run very fast (VR has to reach stable and high FPS, which critical in a whole different way than with a map view looked on small phone display, for example).

VR bring the user's viewpoint very close to the ground. This sometimes brings surprises for the developer who has been modeling the scene with a regular desktop display and possibly even from a bird's eye view. Incorrect texture sizes in tileable textures become very apparent. The appearance or lack of seemingly insignificant visual details become very apparent. The VirtualOulu and Kizhi scenes provide examples of VR use.



**Figure 4.** Rotuaari in VirtualOulu scene before modifications for VR. Texture sizes, the level of detail and material quality look fine from this perspective.



**Figure 5.** The pedestrian street Rotuaari in a VR scene. Texture sizes has been scaled down significantly. Material shaders have been enriched with normal maps and masks for reflective and emissive parts. New content has been added for visual details. However, there are no dynamic shadows due to using Unreal Engine forward rendered optimized for VR.



**Figure 6.** The Kizhi virtual museum. The table and chair to the right seem like very insignificant objects in a desktop display. However in VR, they become very noticeable.

Although VR has the tendency on making normal maps look less convincing than with regular desktop displays (LaValle 2016), the scene might look unrealistic if they are missing altogether.



**Figure 7.** The Kizhi Virtual Museum VR scene. Otherwise high quality textures have no normal maps. Many users comment the wooden terrace “looking more like plastic”.



**Figure 8.** In VirtualOulu VR scene, normal maps were either generated from the existing textures, or general normal maps were used with plain albedo materials (such as with concrete buildings).

Part of the VirtualOulu scene was made with very simple models and textures. Surprisingly, these simple models were somewhat easy to enhance for VR purposes. The simple geometry is fast to render which is crucial for VR. In addition, simple base-color textures were easy to utilize for Physically Based Rendering (PBR) paradigm. In PBR, the base texture should only contain the color of the object. All light properties are mapped using additional shaders. In both Kizhi model, and the simple VirtualOulu models, the assets had albedo/diffuse maps with ambient occlusion lighting baked in Blender. Luckily however, the source materials also included the non-baked textures. Using only base textures, additional lighting properties, such as normals, reflective and emissive properties could be added to material shaders and let the game engine compute all lighting (either dynamically, or baking through game engine). The

simple albedo textures of VirtualOulu were also easy to utilize for generating masks that pick out windows from the rest of the building for lighting related masks.



**Figure 9.** In the texture on the left, the windows can be masked using a color picker tool. On the texture on the right, the windows have to be picked manually.

## 2.3 Sketching of building masses in an area plan

- In urban planning, when an area is planned, but not the buildings in the area yet
- Kind of an area model
- Correct kind of building masses are useful, often focus on height too
- No details or features of the houses on purpose: for example colours, windows or balconies would distract people from considering the area, not buildings



**Figure 10.** Sometimes an even more diagrammatic style is desired, with just white colour and outlines.

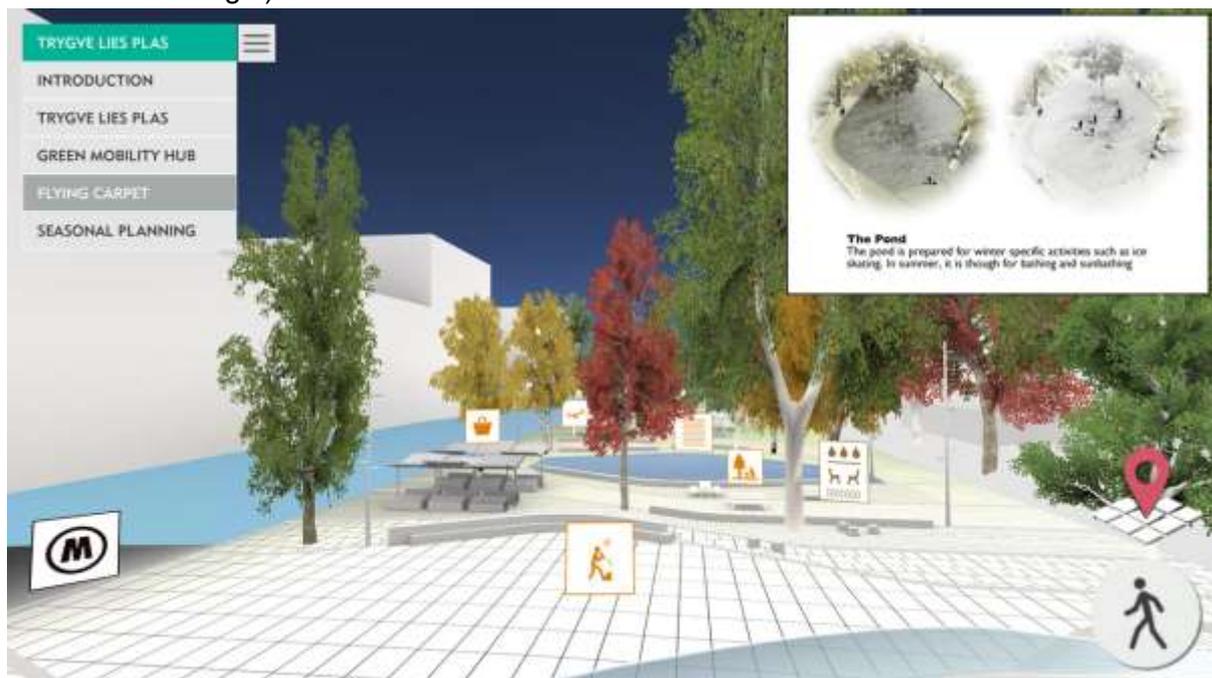
Such outlines can be difficult / heavy (or error prone, with lighter settings) to render well in real-time processing (using a post processing shader to detect edges and draw them). Can require pre-processing or even authoring: in the example case are premade geometry (line meshes) - possibly nice with extra mesh information for custom line drawing.



**Figure 11.** A different style is rendering nice ambient (global illumination) light to the building mass boxes so that the shapes are visible (and not just white featureless mass), in which case outline lines are not required.

## 2.4 Illustrate a plan with identifiable semantics: planned services (e.g. a cafe), but not building designs

- Placing of services a key in an area plan
- Granularity of a separate 3D scene object per feature in the area is ok technologywise. For example a bench as an individual scene object in a group of benches, is ok in the sense that works technically well for relatively small amounts (tens), and allows easily many kinds of interactivity (like sitting on a particular bench, or coloring it).



**Figure 12.** A different style is rendering nice ambient (global illumination) light to the building mass boxes so that the shapes are visible (and not just white featureless mass), in which case outline lines are not required.

## 2.7 Detailed Models: Photorealistic and Construction models

Detailed models such as building information models (BIM) are used to guide the implementation and planning of construction. For building virtual cities these models can be overly complex, but they can still be used after optimization in many ways in visualization or as a basis for modeling more low poly buildings with realistic proportions. Photorealistic models with detailed photo like rendering, with realistic material properties and lighting are another form of detail-heavy models that are used mainly for architectural visualization and interior design. These models do not contain details, such as piping inside the wall, in a way the BIM models do, but they are usually restricted to small areas, which also restricts their usability for building larger scenes.

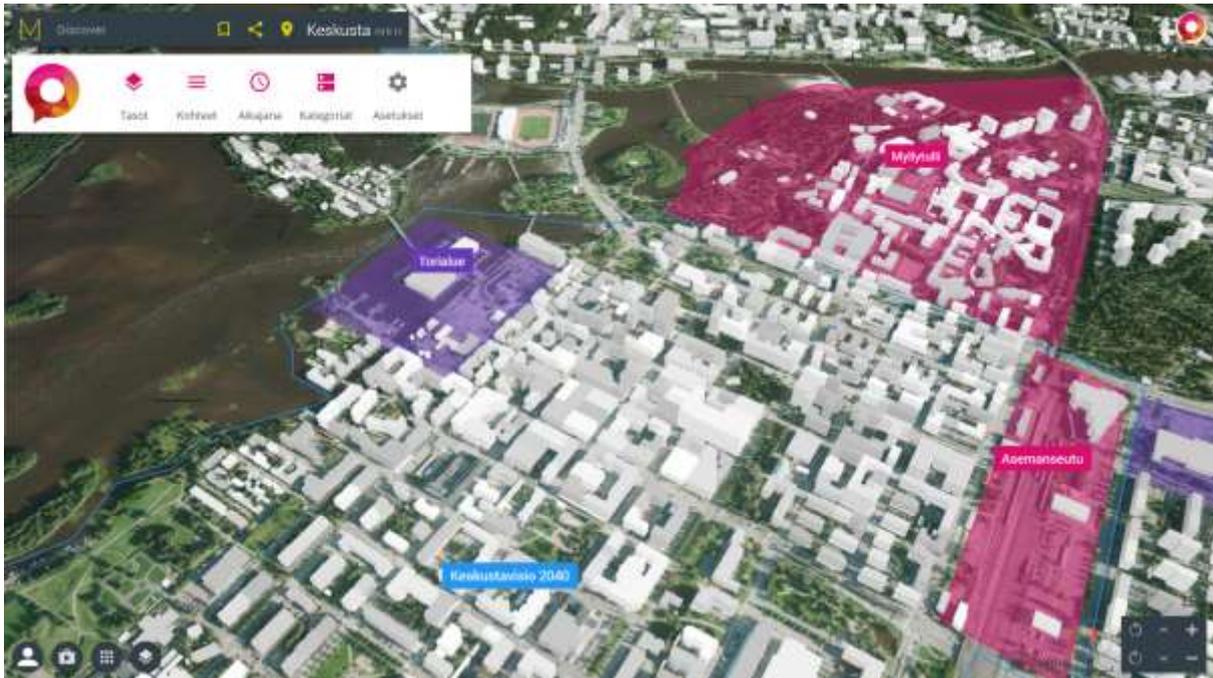
## 2.8 City models

City models are used for visualization of large urban landscapes. They can be procedurally generated from map data in many ways (Figure 14. & 15.)



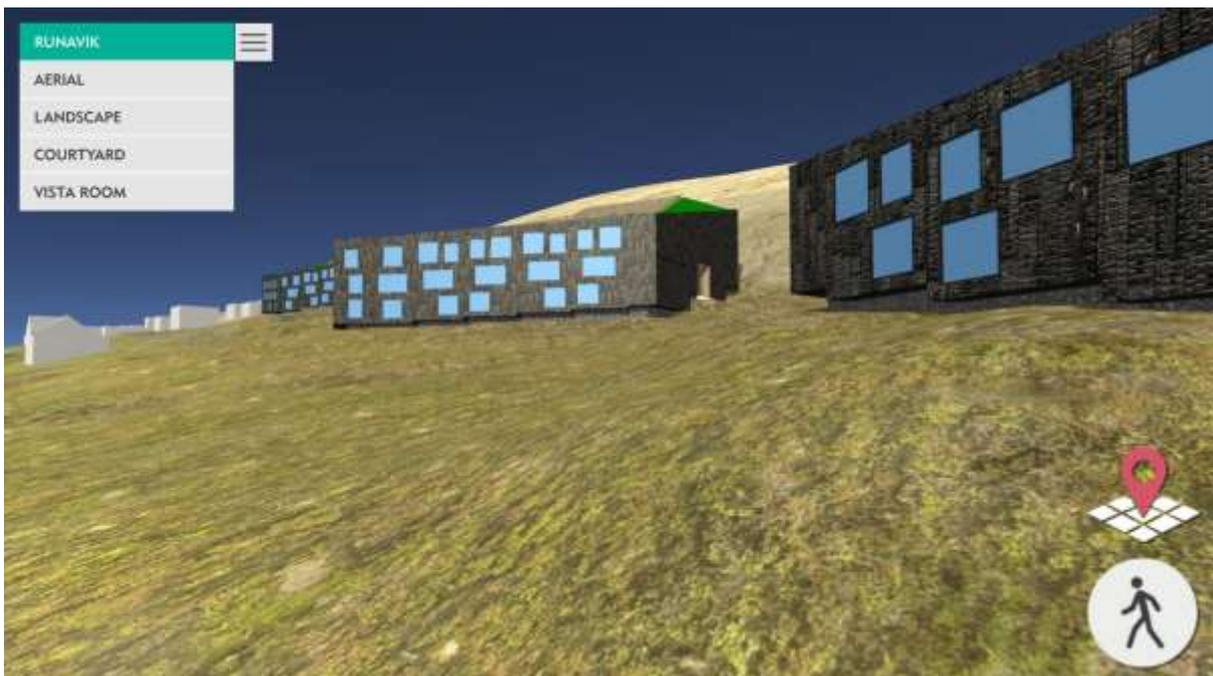
**Figure 14.** NYC DoITT CityGML in Cesium (open source WebGL) Million buildings on Manhattan demo <https://cesiumjs.org/NewYork/>

Typically there is one object per building in city models. For drawing in graphics engines, batched to single large objects. Individual objects are still identifiable from the merged mesh data.



**Figure 15.** View on Oulu in MAPGETS - using the official service data (WFS) for the building outlines -- is similar to what Openstreetmaps provides globally. The area colorings derive from City Info which is a separate system, an application on MAPGETS. WFS layers do have area informations too, such as administrative boundaries (<https://mapgets.com/>).

## 2.9 Terrain / heightmaps and ground texturing



**Figure 16.** View on Oulu in MAPGETS - using the official WFS data for the building outlines -- is similar to what Openstreetmaps provides globally (and if this is not the wms then is from osm) - <https://mapgets.com/>

Depending on the case, the elevation of ground can be irrelevant or a key part of the model. For many environments the data on ground elevation is already available as height maps. Modifications to terrain, in for example a building project, may need to be done to heightmap in order to use the efficient terrain systems in game engines (for level-of-detail etc). For scenes depicted in Figures 14-15. simple aerial view of the ground on flat surface is enough, but if ground texturing needs to cover large areas with enough detail for street level first person use, texture tiles and so-called “splat mapping” is needed.

## 2.10 Environment, lighting and materials

Besides buildings and terrain, environmental settings with city models include light and possible other environmental effects such as weather. In city planning, realistic daylight is often aesthetically pleasing and sought after. From practical point of view it can be used in 3D models for viewing the shadows cast by new buildings, and for calculating solar energy production potential. A principal technical consideration with 3D city model lighting is whether to use a static or real-time dynamic lighting. A powerful combination in real-time graphics today is combined use of precalculated lighting information and real-time shading (e.g. baked realtime global illumination).

Regarding authoring of materials, physically based rendering (PBR) has emerged as a standard technique that works across several authoring tools and game engines.

## 2.11 Audio

Audio is related to both use of 3D for real-life planning purposes, and use of 3D for virtual environments. In planning, it can be useful for noise analysis. For immersive VR, and sense of presence, audio may be even critical key component.

For such uses, we should include sound sources and audio properties of the environments as part of city models. Game engines and the web support it well, also in realXtend a simple Sound component made good sense.

## 2.12 Interactivity

When planning how to create your 3D content, you should consider whether and what kind of interactions are to be done with it. For example with a building, should you be able to open the doors? Such requirements can be very application specific. Also, it can be specific details deep in the application features which require some attribute in the 3D data, so the requirements for the models can be even difficult to know without careful analysis or test.

### Interactable parts

- should we able to open doors? what about windows?
- do we need to know where roads and pathways go?
- what else we need to interact or know about in the particular case?

## 3. Technical restrictions and solutions

### 3.2 Web publishing

When publishing 3D models on the web there are basically two choices:

- a) Web like use with fast interaction, no large downloads at start but constant streaming of data as needed
- b) Package the Web 3D application with the data as a download (wait before use). Similar to games which may have even large downloads up front.

Both ways of publishing set certain common requirements and targets, such as that the downloads should be as small as possible. But they have differing recommendations too for granularity and even compression formats.

In scenario a) the user interface of the 3D city model and its applications should meet generic criteria for web usability, including support for a range of popular devices, reasonable loading times, and simple and clear navigation. Web-based 3D city models are problematic on all these fronts. WebGL requires suitable hardware and browser support and its performance varies greatly between different devices. The size of downloaded files and memory requirements of large 3D scenes with detailed building models and textures are huge compared to conventional web pages with just text and images. Navigating in 3D spaces is relatively complex and there are little conventions for generic web outside the genre specific specialized controls that only gamers are familiar with.

In the original Web publishing of VirtualOulu (2014, by Adminotech using WebTundra on three.js), the challenge of large amounts of data to be downloaded was addressed in two ways: (1) The web UI sequence is designed so that the first scene to be loaded is a light bird's eye view that loads quickly and performs well on most devices (Figure 2(a)), and (2) during navigation in the immersive street level view (Figure 2(b)) only 'relevant' nearby objects of the scene are loaded dynamically on-demand. We employ a web friendly dynamic on-demand (un)loading of assets instead.

The large amount of data of a detailed 3D model is also critical for memory use in the user device. With WebGL, if an application tries to use too much graphics memory it simply crashes (the browser tab, in case of Chrome). Our dynamic (un)loading of assets on-demand works as follows. Objects are prioritized according to camera view frustum and distance, so that only nearby objects are loaded to fit within memory constraints. This also reduces the CPU load of the renderer when it processes the scene. A k-d tree is used to get nearby objects efficiently. When user navigates to a new area in the scene, corresponding new objects are loaded on-demand and the ones left behind are removed from the scene and from the memory. Neither Tundra nor WebTundra feature this kind of dynamic scene management so it was implemented for VirtualOulu specifically in the application code. In the future, it would be desirable to refactor this functionality into a generic scene manager.

## 4. Modeling recommendations

In order to get suitable well working 3D models the requirements need to be identified. Next, generic guidelines (regarding the desired nature of the models) for clean and real-time friendly 3D authoring are needed. Further one should not assume that deliverables will meet requirements right away, integrating models into the existing 3D landscape almost always requires extra effort therefore the deliverables should include all available source materials and the original work files for models and textures.

### 4.1. Identify the needs: What Questions to Ask

In the following there are few very practical questions to guide authoring the requirements for modelling work, this is especially relevant if one outsources the modelling.

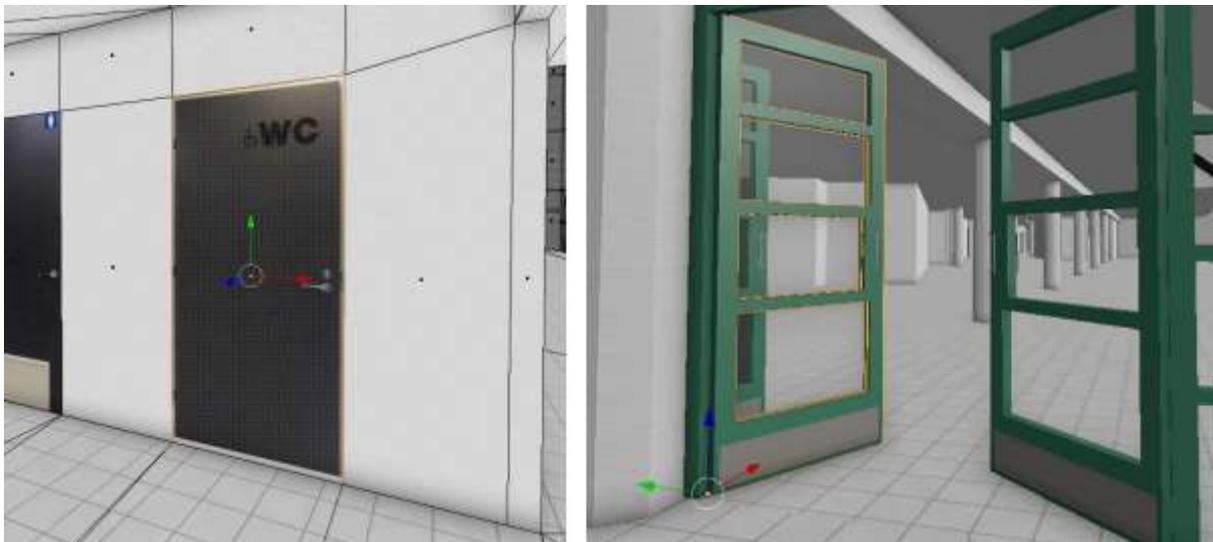
- What is the required Level of Detail (LOD)? The LOD level may vary even inside a modelled area (Figure 17).
- What is the required granularity of your scene? How many objects per building are there in VirtualOulu (also variants where many objects per building) or on large map views one may combine hundreds of buildings into one, where building information models (BIM) have small granularity for semantic useful data, for example positions of the electric plugs etc. In general, small scale interactivity and information additions require equally small granularity in the models
- What is the desired aesthetics and use case for the models?
- What can be created procedurally or autogenerated and what needs to be modelled manually?
- What are you Interaction and semantic needs?
  - Do we need to know which objects are doors?
  - Do we need to be able to open doors? what about windows? etc.
  - Roads, number of aisles and other traffic information perhaps? (is already in traffic apps, domain specific information in that sense)
- Is our landscape such that we need realistic elevation in terrain or can the city scape be build on a flat piece? Building a scene on a flat terrain is much easier that having to consider ground elevation. Later on elevated terrain also affects the mapping of animation for user avatars and other characters or objects in the scene. On the other hand, elevation may be a key esthetic feature in creating a realistic skyline for your.
- How precisely do the proportions and measurements of you models need to follow the proportions of the physical environment? This is especially relevant if building indoor scenes for visualizing sensor data or movement that is tied to a real-world coordinate system.

### 4.2 Case: VirtualCampus

The oldest parts of the University of Oulu campus in Finland were taken into use early 1970 [Kampusvisio n.d.]. The campus main building has had numerous expansions since then resulting into a maze of corridors that stretch out from the about one kilometer long central hallway. Even the most recent modules of the campus building have been build when 3D modelling of any kind was not done, hence the source material for modelling are mainly

blueprints (.dwg), photographs, video and measurements taken on site. Some newly renovated areas have corresponding BIMs, but those have limited use considering our modelling requirements.

There are multiple stakeholders and projects involved in using the 3D models of the campus. The interests of these stakeholders vary from simple 3D visualisation of facility equipment to VR research where the visual requirements are much more demanding. Since the use of the models is this vast; for real-time visualization, games and serious applications, the complexity of the geometry should not restrict the use of the models on mobile applications. The models should though have a good quality geometry and textures like has been specified by Alatalo et al. 2016. For the end use the models are exported into an unspecified game engine hence the source files should be offered in format that can be exported into most 3D editors or game engines, such formats would be Autodesk FBX or Wavefront object. Being able to edit the original models' meshes is also necessary as the minimum detail requirements of the models do not match the requirements for the use of the models for games or VR experiments. Good example of this are the doors on campus site. The minimum requirement for the doors is to have them outlined from the model, the right size and shape to make it possible to later add a more detailed right size door or even animated one on the spot (Figure 17).



**Figure 17.** Doors in virtual campus, the minimum requirement is that doors are textures planar cutouts on a mesh (Figure left) ideally doors adjoining spaces are more detailed and have pivot points at the hinge (figure: right) for easier animation.

One specific modelling requirement for the 3D campus relates to the visualisation of sensor data. The proportions and measurements of the indoor model should match the proportions and measurements of the physical environment to make indoor locationing of data, items and events possible. This requires checking the measurements on site. If the modelling is outsourced to a distant location one should be ready to spend time doing the measurement checks and communicating those to the modellers.

# REFERENCES

- Alatalo, T., Koskela, T., Pouke, M., Alavesä, P., & Ojala, T. (2016). VirtualOulu: collaborative, immersive and extensible 3D city model on the web. In *Proceedings of the 21st International Conference on Web3D Technology* (pp. 95–103). ACM.
- Biljecki, F., Ledoux, H., Stoter, J., & Zhao, J. (2014). Formalisation of the level of detail in 3D city modelling. *Computers, Environment and Urban Systems*, 48, 1–15.
- Hickey, S., Pakanen, M., Arhippainen, L., Kuusela, E., & Karhu, A. (2012). Service fusion: an interactive 3D user interface. In *Proceedings of the 11th International Conference on Mobile and Ubiquitous Multimedia* (p. 53). ACM.
- Kampusvisio 2040. (n.d.). Retrieved December 22, 2016, from [https://issuu.com/oulunyliopisto/docs/kampusvisio\\_2040?reader3=1](https://issuu.com/oulunyliopisto/docs/kampusvisio_2040?reader3=1)
- LaValle, S. (n.d.). *Virtual Reality - LaValle*. Cambridge University Press. Retrieved from <http://vr.cs.uiuc.edu/>