Ilari Kukkonen

# Real world location in a 3D real-time simulation

Bachelor's thesis
Degree Programme in Design

2017

South-Eastern Finland
University of Applied Sciences

South-Eastern Finland University of Applied Sciences

| **Author (authors)** | **Degree** | **Time** |
|---|---|---|
| Ilari Kukkonen | Bachelor of Design and Arts | April 2017 |

**Title**

Real world location in a 3D real-time simulation

29 pages

**Commissioned by**

Creanex Oy

**Supervisor**

Marko Siitonen, Lecturer

**Abstract**

The goal of this thesis was to describe the process of 3D modeling the central hospital of Kotka in 3ds Max, and consider what actions could have been taken to achieve more optimal results. The modeling was made for Creanex Oy as a part of an ambulance simulator project, set in the area of the central hospital of Kotka. The 3D models were produced in Autodesk 3ds Max, and textured with Adobe Photoshop CC and Gnu Image Manipulation Program (GIMP).

The objective of the thesis was to create the main buildings of the hospital, as well as a terrain model that would include a road leading to the hospital. The area would follow the layout of the real location, and needed to be recognizable in the simulation. The thesis inspects the processes used in creating large environment assets. The goal is not to focus on small, technical details, but to give an idea of what issues should be considered before starting the production of a large-scale 3D environment.

The production began with a block out of the major elements of the scene. After the initial block out, the work load was divided into terrain and buildings. Each portion required different approaches and 3D techniques.

The thesis analyses the results and tries to find alternative methods and solutions that may have made the processes easier, or possibly led to better results. Project planning, iteration and execution of the project are under close evaluation. Attention was also paid to the factors that slowed down development.

**Keywords**

Real-time simulation, 3D modeling, Computer graphics

## CONTENTS

# 1 INTRODUCTION

The purpose of the thesis is to report the process of creating a 3D environment based on a real location for a real-time rendering engine. The goal of the project was to create a 3D model of the central hospital of Kotka, as well as the terrain surrounding the hospital. The 3D models were to be used in an ambulance simulator created by Creanex Oy, who was the commissioner of the thesis. Creanex Oy is a company located in Tampere, specializing in real-time simulations and virtual training environments for different kinds of machinery.

This thesis will analyze the 3D content production methods used by the author, for the reader to gain a better understanding of the process of creating real-time environments, as well as what could have been done differently in the project. The project began in September 2016 and paused for the duration of December 2016. Production continued the 6th of January 2017, and lasted until the deadline of 2nd of February 2017.

The author of the thesis is a game design student in Xamk University of Applied Sciences. The roles of the author included the planning of the 3D area, production of the 3D assets, creation of textures and collision models, as well as exporting of all materials in correct formats for the real-time engine. The author had no participation in the implementation of the assets into the target engine.  The project supervisor was Joonas Korpela from Creanex Oy, who was also the contact person between the author and the Creanex office.

# 2 PROJECT OVERVIEW

The project began in the beginning of September 2016 with Creanex Oy as the commissioner of the project. Creanex Oy was to create an ambulance driving simulator for Kymenlaakso University of Applied Sciences. The author performed distant work from Kouvola, visiting the Tampere office every two weeks, or when otherwise required. A weekly progress report was sent to project supervisor Joonas Korpela to keep Creanex Oy up to date.

The primary goal of the project was to create a 400-meter by 400-meter area containing the hospital and necessary environment elements. The piece was to be attached to the company’s own, already existing simulation environment in a way that it would be possible to drive from the existing city to the hospital within the simulation. Because of this, the 3D content would have to be produced in real-world scale. The primary 3D application used in production was 3ds Max 2014, which was used for 3D modeling and preparing the model for exporting. Due to the author’s previous experience with different versions of 3ds Max, it was possible to work efficiently without problems related to the usage of the application itself. Adobe Photoshop and Gnu Image Manipulation Program (GIMP) were the primary image editing tools. They were necessary for creating original textures, as well as editing existing ones to fit the 3D models.

Other software used in the project included an application called CRNXMeshViewer for previewing 3D models and their materials. CRNXMeshViewer aided in the visualization, as well as in the testing of the 3D models and textures. In case the mesh conversion was not performed or had failed, or the material file was not correctly constructed, the application would present an error and refuse to open. The target engine was Ogre 2.x.

# 3 3D PRODUCTION

3D meshes are built out of polygons. On each side of a polygon is an edge, and on each corner of the polygon are vertices, connected by an edge. Polygons are generally four-sided. Four-sided polygons are often called ‘quads’, whereas polygons with more than four sides are called “n-gons”. “N-gons” are generally not desirable in finished 3d models, as real-time renderers can have trouble processing them. (Chopine 2011, 21-22.)

3D production for real-time applications differs from production for film, animation or other pre-rendered media. While the work description of a 3D modeler is very similar in both cases, the level of detail in the work differs, as real-time applications are unable to render the same amount of detail that goes into a 3D model created for film. Real-time models are limited to a low level of geometry, and are allowed only a few texture maps, whereas a film asset may be worked on for weeks and have a higher amount of geometry, as well as more texture maps with a higher level of detail. While both jobs have similar descriptions, the goals and working methods are different, as the game artist works in polygons, and the film artist may work with methods that are not suitable for a real-time environment. (Wagstaff & Dreakhshani 2016, 11.)

In a game environment, modeling is a constant trade-off in detail versus economy. Models are required to follow strict guidelines, such as the model's polygon count, in order to maximize performance. (Wagstaff & Dreakhshani 2006, 20.) The more polygons a 3D mesh has, the more detail it has. However, rendering it in a real-time application will also be more demanding. (Chopine 2011, 33.) When imported into a real-time engine, the polygons of a model are changed into triangles, regardless of how many corners each polygon has. Because of this, one should not rely on the displayed polygon count of a modeling application, unless they are sure it displays the polygon count as triangles. (Polycount 2016.)

3D graphics have always required the support of software engineers, as they create the tools that allow the creation and rendering of 3D models and other effects (Wagstaff & Derakhshani 2006, 37). The result might be a pretty 3D environment, but the 3D artist must walk the line between visual and technical aspects of 3D evenly (Wagstaff & Derakhshani 2006, 29). Common technical knowledge regarding 3ds Max and 3D real-time applications was required in the project, as the work included exporting 3D models and their collision models, as well as interaction with different file formats.

## 3.1 Texture maps and file formats

Texture maps are 2D images that cover the surfaces of 3D objects (Ahearn 2006, 41). To create good textures, it is beneficial to understand photography how different surfaces work in real life. This means understanding concepts such as specularity, luminosity, reflectivity and displacements. It is possible to combine knowledge of photography and painting with this knowledge, and use it to create various real or imaginary surfaces as 3D-rendered textures. (Wagstaff & Derakhshani 2006, 21.) In this project, the goal was not to create perfectly realistic materials, but to accurately convey the hospital and its surroundings in the simulation, in a manner that made them recognizable to the user.

The process of mapping textures onto a 3D model is called UV mapping. The process consists of creating 2D coordinates from the geometry of the 3D model, onto which the textures can be applied. The process is best described as cutting and spreading a 3D model onto a flat surface. In modeling applications, those cuts are called seams. (Simonds 2013, 109.) The coordinates, called UVs or texture coordinates, are stored into the vertices of the model, and range from zero to one, which indicate the sides of the texture map. If the coordinates go below zero or above one, in other words outside of the UV range, the texture will tile. (Polycount 2016d.)

The surface color of an object is defined by the diffuse map. The diffuse map contains only the color information of a surface. (Ahearn 2008, 67.) Normal maps are RGB images that modify the surface normals of a 3d object, in order to depict detail that is not actually present in the geometry. Surface normals refer to the direction the surface of the mesh is facing, and how the mesh is lit. Each pixel of a normal map contains information on how to modify the surface normals of a model. (Unity3D 2017.) Roughness textures are greyscale texture maps that control how rough the surface of the material is. Rough materials scatter reflected light in more directions than smooth materials. (Epic Games 2017.)

A height map is a black and white image, which can be used to modify the geometry of an object. The height map changes the object’s surface by pushing the vertices affected by white values upwards, and the vertices affected by dark values downwards. This can be translated into numerical values - 0 equals black, 1 equals white, and 0.5 equals grey. As grey is in the middle of the scale, it does not affect the geometry. (Pixologic 2016.) Compressing texture maps is important, as it may increase performance and decrease the size of the texture file (Polycount 2016b). In this project, the texture files were required to be in the DDS image format. DDS stands for ‘Direct Draw Surface’. It is a texture container file format, and can contain many different compressed or uncompressed image formats in various bit depths, and may be stored in various layouts. (Polycount 2016c.)

It is also necessary to keep in mind the vertex count of a 3D model, as it will ultimately be a more important factor for the model’s performance than the polygon count alone. In addition to the polygon count of the object itself, smoothing groups, as well as UV splits, contribute to the vertex count of a 3D model. (Polycount 2016.) Each vertex that is split in the UV map, is read by the engine as two separate vertices (handplane3d 2013). This is useful to keep in mind while planning the UV mapping and texturing of a model.

Finished 3D models need to be exported from 3ds Max in a specific file format, so that it is possible to import them into the target real-time engine. In this project, the required file format for 3D model files was Ogre’s “.mesh” format. 3ds Max could not export models in this format by default, so a plugin called Easy Ogre Exporter had to be installed. After exporting the 3D mesh from 3ds Max, it was necessary to use other applications, created by Creanex, in order to convert the mesh files to a more specific format.

Texture files were required to be in DDS image format. This was done by opening the texture file in GIMP and exporting it as a DDS file, using the GIMP DDS Plugin. The export settings had to be set differently depending on the type of texture that was being exported. Diffuse maps, normal maps and diffuse maps with transparency were all required to be exported differently. The exported DDS files were then converted to a more specific format, which the GIMP DDS Plugin was not able to produce, using a command line application called texconv.exe. The company's 3D modeling documentation contained the specific instructions on how to export and convert texture files.

To display the textures correctly on the 3D model, the model required a material file, which was automatically exported from 3ds Max alongside the 3D mesh. The material file determines how the surface of the 3D object appears when rendered (Ogre 2014). In the project, materials were to be named and assigned correctly within 3ds Max, as they would have to be identical in the material file. This is necessary to allow the engine to identify the correct materials, as well as to keep materials organized. If a material was used on an object, but not specified in the material file, the information would be stored in the mesh file and be brought into the engine, which was not desirable.

Texture maps included in the material files mainly consisted of diffuse maps and normal maps. By using roughness textures, it would have been possible to create more variety in the reflections of different surfaces, but the author decided that they would not be used in most cases, as applying a unique roughness texture made very little difference compared to the default roughness value of the renderer. Instead, the default roughness value of the renderer was used.

Whenever a 3D model was exported, a collision model had to be exported with it. A collision model is necessary for collision detection, which prevents two objects from passing through each other (Rani 2014, 11). In this project, it was necessary to make the collision models as close to the shape of the original visual models, in order to get accurate collision detection. The collision models were created with a PhysX plugin for Autodesk 3ds Max 2014 by NVidia. Exporting collision models was usually one of the last steps of asset creation.

## 3.2 Simulation

A simulation can be described as a method to observe the behavior of model of real or imaginary systems (Raczynski 2006, 1). Some simulations exist to mimic a specific experience or environment, which for example an operator of dangerous and expensive machinery may face in their work (Becker & Parker 2011, 4).

Simulators generally include mock-ups of the actual equipment used to control the vehicle or machinery that is being simulated. The virtual environment replicates the way the vehicle is controlled, as well as the environment it operates in. (Becker & Parker 2011, 10.) Because of this, it was very important to keep in mind the primary goal of the project, which was to create an environment where the ambulance could operate in. As the goal of a simulator is to make the system convey the experience of a real situation to the user as accurately as possible, considering details such as the width of the roads, as well as major landmarks and features of the location to make it recognizable to the user was important.

# 4 TERRAIN

The first phase of the project was the creation of the terrain. Making sure that the terrain was the correct size and fit the existing simulation environment was important to continue the project smoothly. This was achieved by using terrain height data. A height map was put into 3ds Max as a background image, on top of which the terrain was modelled. The height map was provided by Creanex Oy. Google Maps, as well as Bing Maps were also used in the project, and proved to be a valuable resource for more detailed reference pictures of the area. The height map and a top down image of the buildings can be seen in Figure 1.

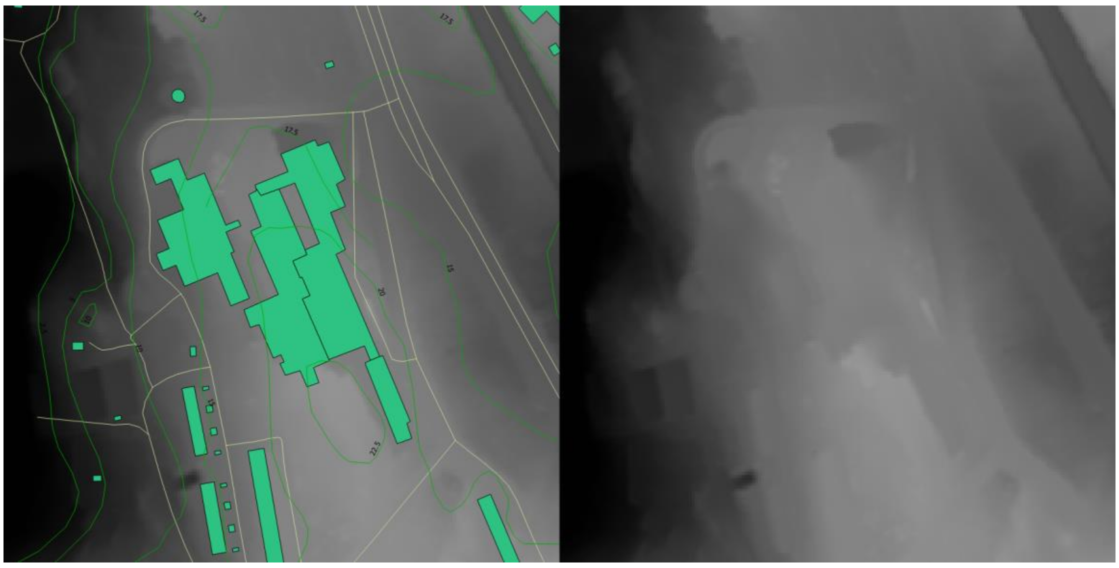
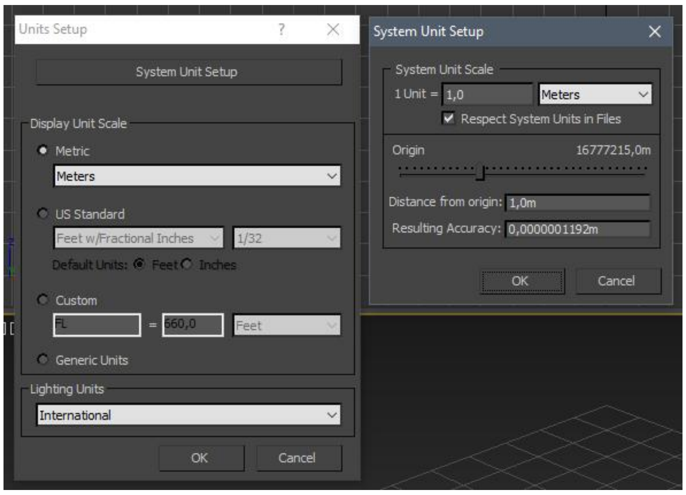


Figure 1. Height map with buildings overlain on top of it (left) and an unedited height map (right) (Kukkonen 2017)

Before beginning, it was important to set units within 3ds Max to meters, in order to create the assets in the scale they would appear in the engine. This was done by going to Customize>Units Setup>System Unit Scale in 3ds Max. Changing the units in the Units Setup Dialog was not enough, as it would not affect the actual scale of the geometry (Autodesk 2016b). It was important to set the units correctly before creating any geometry, as changing the unit scale later in the project would have caused glitches and visual artifacts in the viewport (Autodesk 2016c). The Unit Setup windows can be seen in Figure 2. Regularly changing the display units to millimeters can be beneficial to check if objects are in their correct positions. During the project, it was discovered that 3ds Max 2014 does not always display accurate location data when the display units are set to meters. For example, a displayed value of 5,0 meters can, in fact, be for example 5000,033 millimeters. While the offset is very small, it may still create gaps between objects. The problem is more pronounced when working with modular parts.



Display Unit Scale
[✓ Metric] Meters
US Standard Feet w/Fractional Inches 1/32
Default Units: [✓ Feet] Inches
Custom FL = 660,0 Feet
Generic Units
Lighting Units: International

System Unit Scale
1 Unit = 1,0 Meters
[✓ Respect System Units in Files]
Origin 16777215,0m
Distance from origin: 1,0m
Resulting Accuracy: 0,0000001192m

Figure 2. Setting up units in 3ds Max (Kukkonen 2017)

To create the terrain, a plane the size of 400 by 400 meters was created. The plane was given 100 subdivisions in order to have enough geometry for distorting the terrain with a height map. High amounts of vertices will allow the height map to provide smoother results (Ahearn 2008, 198). However, the geometry had to be cleaned up and optimized later in production. A Displace modifier was applied to the plane, and the height map was inserted into the Map slot of the modifier. A displace modifier modifies an object’s geometry, and its effect can be controlled in a variety of ways, such as with a bitmap (Autodesk 2014b). The results provided by the modifier can be seen in Figure 3.

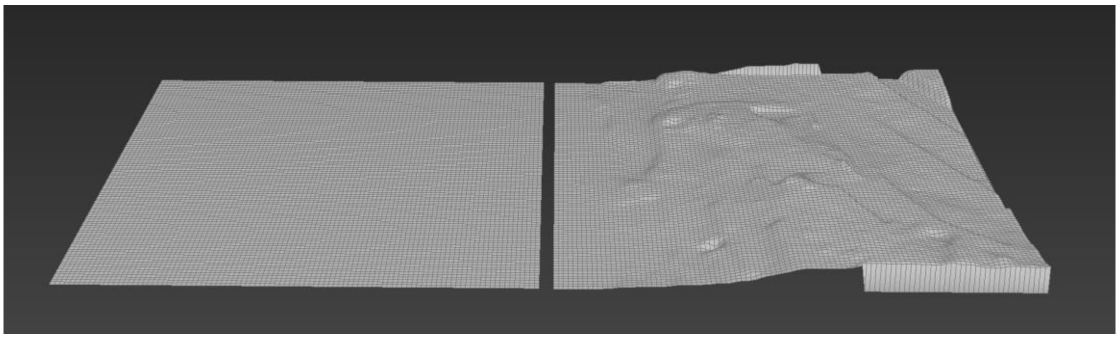


Figure 3. Terrain plane without a height map (left), and the height map applied (right) (Kukkonen 2017)

The height map that was used was flawed, as it had captured the height of the trees and buildings of the area, and as such would not offer an accurate representation of the terrain. Because of this, the terrain had to be edited by hand using the Paint Deform tools inside 3ds Max. The Paint Deform tools are a part of the Graphite Modeling tools in 3ds Max, located in the Freeform tab. The specific tool used was called Push/Pull, which allowed for fine-tuned, freeform modifying of the terrain by simply clicking and dragging the mouse along the model (Autodesk 2016a). Although getting the terrain to the correct height was made easier by the Paint Deform tools, the lack of clear reference for the terrain provided a minor challenge.

## 4.1 Roads

The first iteration of the roads was created by using the Line tool inside 3ds Max and tracing them along the roads in the reference image. A line in 3ds Max is a spline object, which can be freely drawn on the viewport and modified from its vertices. Splines can be created by going to the ‘Shapes’ section of the ‘Create’ panel, and selecting ‘Splines’ from the drop-down menu. (Autodesk 2014c.) From its properties, the spline object can be transformed into a rectangle shape, which follows the line. These rectangles were used as a starting point for the roads. The terrain needed to be created so that it would fit with another, existing simulation terrain. To ensure a seamless connection, the roads would have to snap accurately to the end of the road in the other terrain piece. This heavily affected the planning of the roads on the hospital terrain.

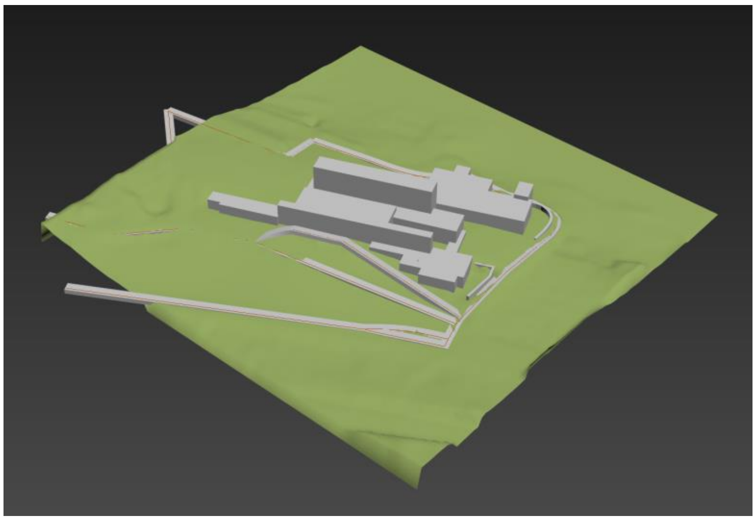


Figure 4. First blockout of the hospital area (Kukkonen 2017)

Initially, it was planned that the roads would be separate objects for the terrain, as seen in the initial blockout of the area in Figure 4. However, it was later decided that stitching the roads to the terrain provided cleaner results. The inspiration for this decision came to the author after researching methods for creation of roads and terrain in video games. Screenshots of the terrain from the game ‘FlatOut’ by Bugbear (2004), shared in a forum thread called ‘Track and terrain’ on polycount.com described how the terrain in the game was created. The screenshots were originally uploaded to Bugbear’s official community forum thread called ‘Making of Flatout 1’.

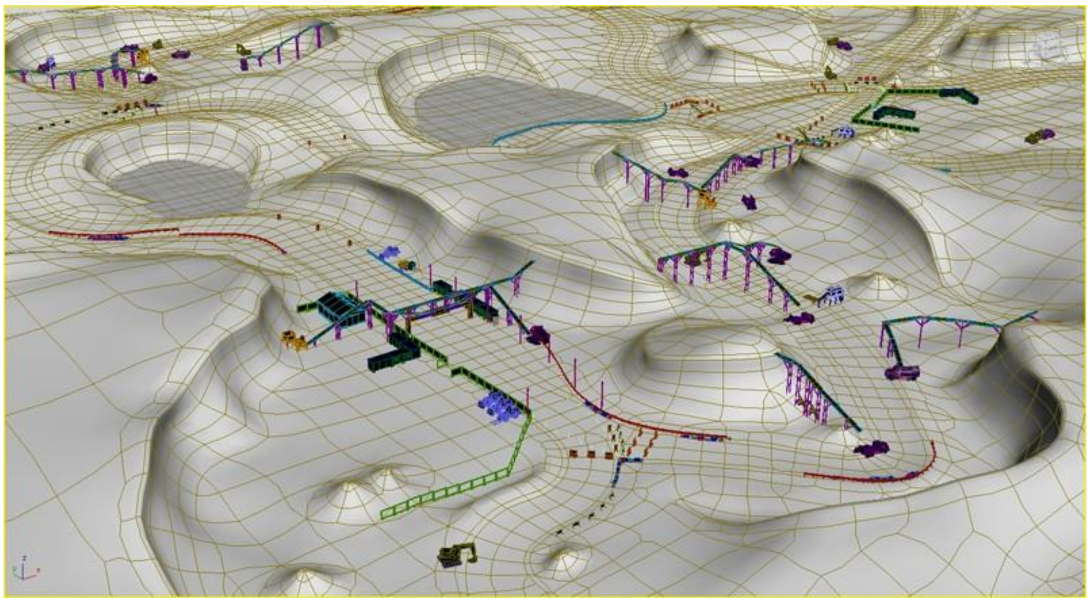


Figure 5. Terrain topology of a FlatOut level (Merikanto 2012)

As can be seen in Figure 5, the geometry follows the curves of the road. By utilizing this information in the building of the terrain, the hospital terrain mesh became easier to work with. Assigning materials to the mesh was also made more convenient, as there was now a clear definition between the road and the surrounding terrain.

To stitch the roads to the terrain, the previous procedure was followed, however after creating the standalone road model, the bottom part was removed, leaving only the top plane of the road. Using the ‘Conform’ tool, the road model was conformed to the terrain. The ‘Conform’ tool is located in the Freeform tool tab, and it is used to conform a surface onto another object’s surface (Autodesk 2014a). While the tool can occasionally provide unpredictable results, in this instance it saved the author a lot of time. The geometry overlapping with the road model was removed from the terrain geometry, and finally the road was attached into the terrain. The two parts were then stitched together, as seen in Figure 6.

While stitching the road to the terrain provided overall cleaner results, it would create problems in case the mesh had to go through changes, as the stitching process would have to be repeated. This problem was emphasized by the irregular shape of the road and the intersection on the hospital entrance, which had to be modelled by hand, with no accurate reference to rely on.

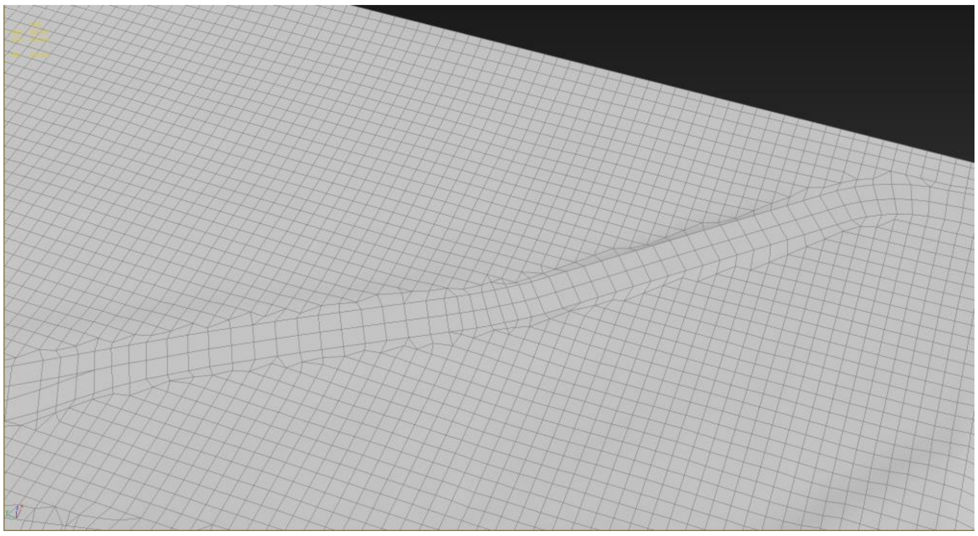


Figure 6. Results of stitching the road to the terrain (Kukkonen 2017)

Towards the end of production, it became more apparent that the optimal technique for creating the roads would have been to use the ‘PathDeform’ modifier in 3ds Max. This modifier allows a 3D model to align itself to follow a spline object, in this case a line (Autodesk 2016d). Using this modifier, it would have been possible to create a small, UV-mapped and textured piece of road that could then have been made to follow a line, forming it into any desirable shape using the modifier. This would have simplified the texturing process of the road.

### 4.1.1 Texturing the road and terrain

Texturing the road required a different approach as compared to other objects in the scene. The model was large, which meant that a tiling texture would have to be used. However, the model also included small details in the form of road markings, such as lines, arrows and crossings. In order to display these details in a sufficient resolution without using an exceptionally large texture map, multiple UV channels had to be used. This process allows layering texture maps on top of each other, and may also be called ‘multi-texturing’ (Ahearn 2008, 11).

In order to layer texture maps, it was necessary to create multiple UV channels. This was done by applying a UVW Unwrap modifier to the desired object. By default, the modifier displays the first, base texture channel. This can be changed by switching from ‘1’ to ‘2’ on the Map channel selection on the UVW Unwrap modifier sidebar. Choosing ‘Move’ in the appearing dialog box causes the existing UV map to be copied to the next channel. This was to be avoided, unless the author in fact wished to override the UV information in the next map channel with the original. Choosing ‘Abandon’ would leave both channels’ UV maps intact.

Despite the renderer displaying all the layered textures correctly, positioning individual, non-repeatable details on the road - such as arrows - proved difficult, as there was no method for disabling tiling for individual texture maps. As it was, the material would tile all texture maps contained within it. To get around this problem, it was necessary to cut appropriately sized geometry for the details to occupy. The texture was then mapped accurately onto the geometry, and assigned a unique UV channel. This method would allow textures which required tiling to repeat along the road, while containing details that were unique and un-repeatable. While this was not the most optimal solution, it provided functional results in the time that was available. An example of road markings can be seen in Figure 7.

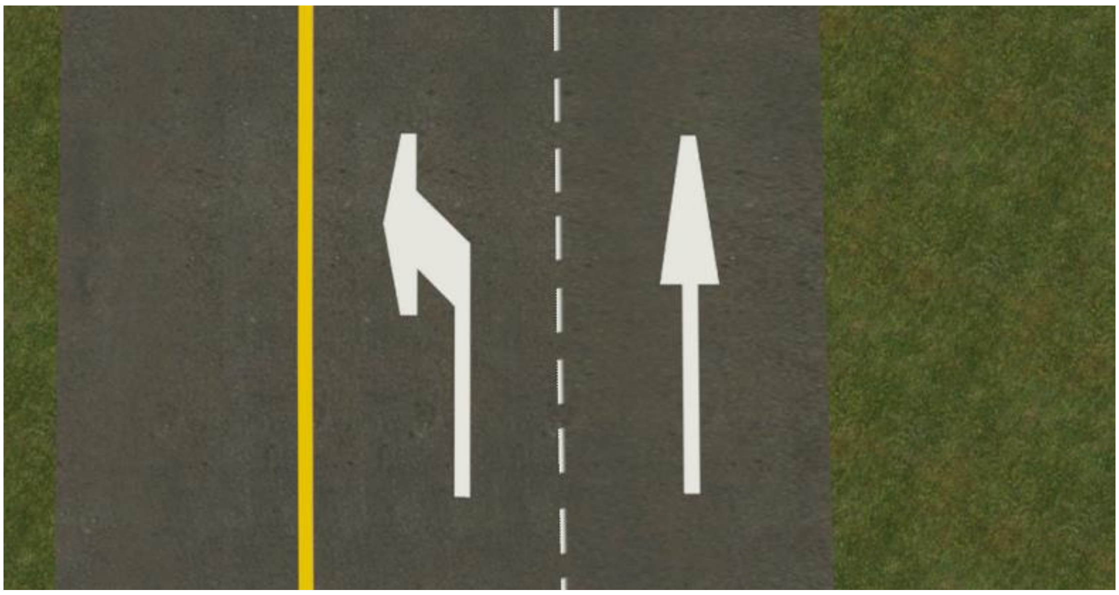


Figure 7. Example of road markings (Kukkonen 2017)

Multi-texturing was not a technique that the author had experience working with previously. Taking advantage of it in the project allowed for great flexibility when thinking about the details that would have to be displayed on the road.

The only texture on the terrain, excluding the road, was a single tiling grass texture. The texturing of road details was one of the last phases of the project, and after the technique was clear to the author, there was not enough time to test it on other areas of the terrain. Possible ways to add variation to the terrain texture would have been to take advantage of multi-texturing, in order to reduce the systematic repetition of the grass texture as seen in Figure 8.

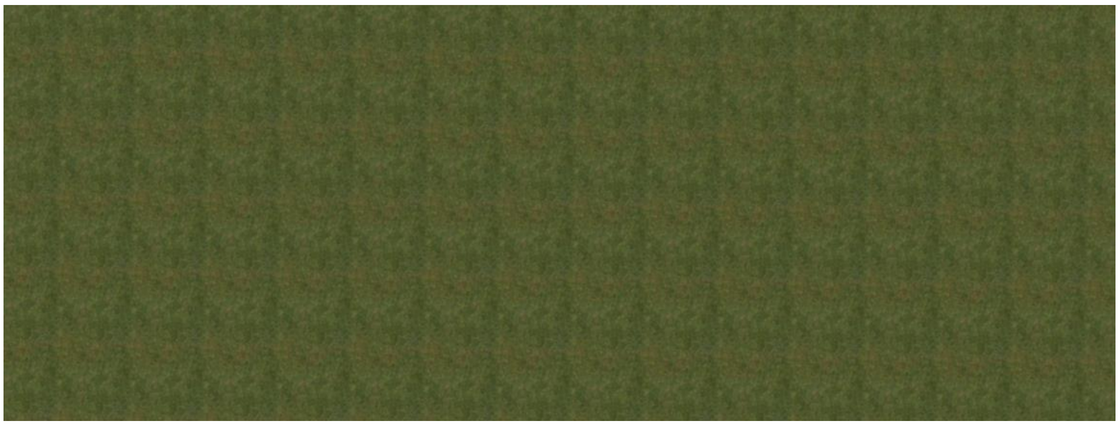


Figure 8. Repeating grass texture from 3ds Max (Kukkonen 2017)

The article “Terrain Advanced Texturing” in the documentation for Epic Games’ Unreal Engine 3 (2012) provides information on methods for hiding texture repetition. The method described as “Multi-UV Mixing: Reducing tiling through scalar mixing”, and is created by mixing a single texture in two different scales (Figure 7). While the article describes how to produce the effect for Unreal Engine 3, something similar could have possibly been conceived in 3ds Max, using basic multi-texturing techniques. Comparing Figure 8 to Figure 9, the difference is significant. However, due to time constraints, this was not possible to test.

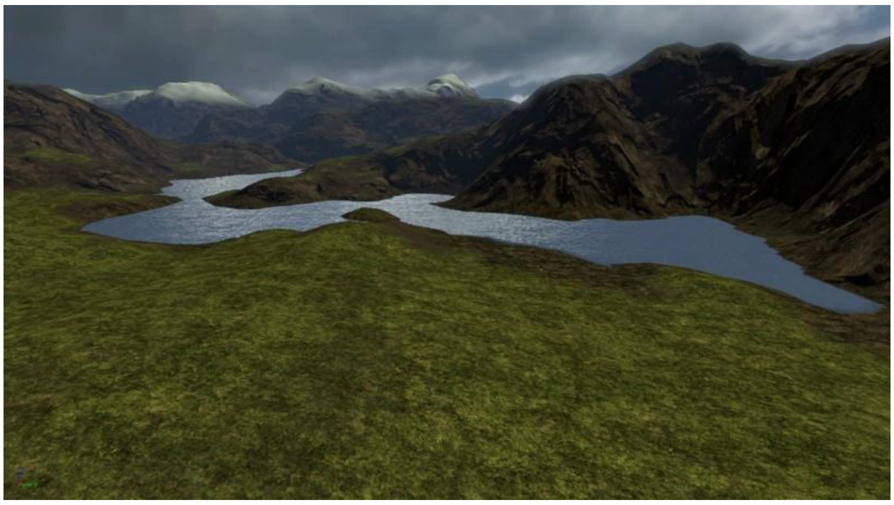


Figure 9. Results of Unreal Engine 3’s Multi-UV Mixing (Epic Games 2012)

Placing a handful of unique decal textures on the terrain would also have broken the repetitive effect. Decals are transparent textures, which are commonly used to place smaller details, such as splash marks, bullet holes or road markings on large surfaces (Polycount 2015).

### 4.1.2 Trees and traffic signs

There was no need to model trees, as Creanex owned existing 3D models of spruces and pines. This was beneficial for the author, as he had little experience modeling trees. The pines looked too thin when put into the engine, so it was decided that only spruces would be used. Using existing 3D models was a good way to save time during the project. However, as the branches of the trees were textured on one-sided planes, it was necessary to duplicate each plane and flip their surface normals, in other words the direction the polygons were facing, as generally, when working in 3D, a polygon is only visible when the viewer is facing it. For example, if one was to place the camera inside a 3D model, they would be able to see through it, as the normal are all facing away from the camera. (Ahearn 2016.) This procedure was necessary to display the branch texture on both sides of the plane it was mapped onto.

Trees were placed on the terrain by using the Object Paint tools in 3ds Max. These tools allow for free-hand placement of objects on any surface in the scene (Autodesk 2016e). When placing trees, a small, but useful detail is to keep in mind how the trees are grouped. There were two different sizes of trees, a large and a small tree. As Neil Blevins describes in his “Clumping or Grouping” -article on his web page (2012), placing objects in groups can make them more readable for the brain, and more aesthetically pleasing. This knowledge was applied to the project, as seen in Figure 10.

The trees were not exported into a traditional mesh format. Instead, a special script was executed in 3ds Max to determine the location of each tree. The script created an XML file, which contained the positions for all trees. This file would be sent to Creanex, who could place instances of a single tree model to all the positions in the XML file. This prevented the need to export an unnecessarily large 3D model, as combining all trees together to a single object would create a mesh with an exceptionally high vertex count.

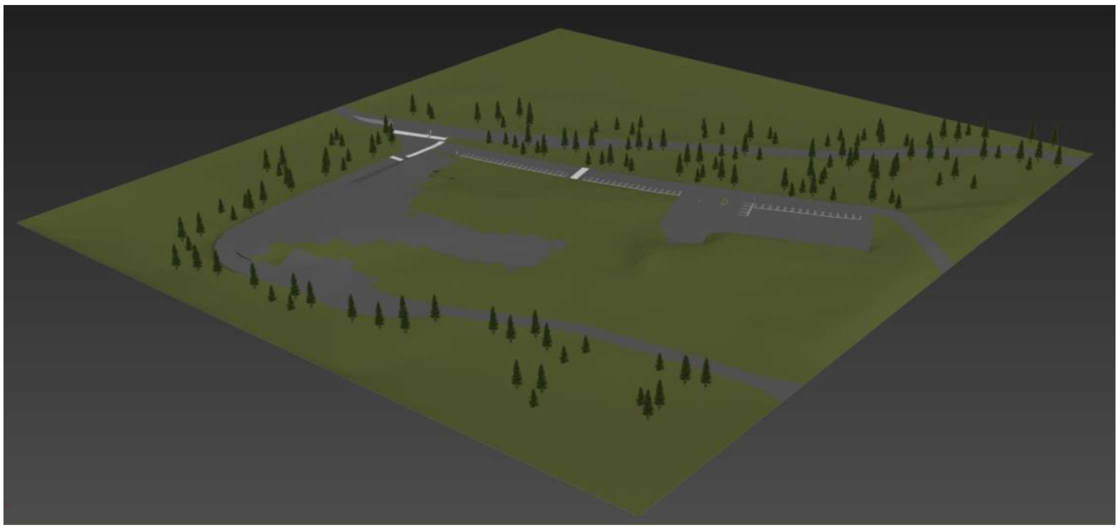


Figure 10. Trees placed across the terrain model in 3ds Max (Kukkonen 2017)

Traffic signs were added as the very last element of the terrain. Creanex Oy had existing 3D models of common traffic signs, which covered most of the signs around the hospital. The only additional traffic sign that had to be made was a lane divider sign, for which only some texture changes to the original signs was necessary. In addition to this, two signs exclusive to the hospital had to be added: a sign pointing to the direction of the hospital, seen in Figure 11, as well as a sign saying ‘Ensiapu’, pointing up the ramp, towards the entrance. These signs were made from scratch. In addition to traffic signs, a simple traffic island was placed on the entrance of the hospital area. Traffic signs were attached into the terrain, and exported as one object with it.

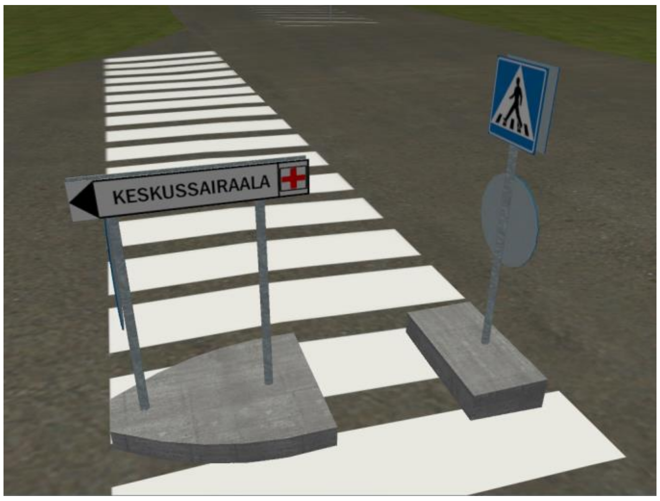


Figure 11. Example of signs from CRNXMeshViewer (Kukkonen 2017)

## 5 BUILDINGS

The buildings were initially modelled relying on the eye to get a block-out of the area for easier visualization. However, these block-outs were not accurate and another solution would have to be found. Seppo Pajari from Kastek Oy was contacted regarding documentation on the measurements of the hospital. No recorded measurements existed, except for an architectural 3D model created by Sweco Oy, which was provided to the author.

The 3D model was unusable for real-time applications due to its high vertex count. Smaller elements, such as windows and rails were included as unique geometry. However, the 3D model made it possible to create an accurate, low poly version of the hospital by modeling a new mesh on top of it. It also offered guidelines for texturing, such as accurate locations of windows and other secondary shapes. The building model also included small pieces of terrain along the edges of the hospital walls, which served as guides when searching for the correct height for the final terrain model. Having an accurate base model to use as a guide was exceptionally convenient, as it would have been impossible to make the building measurements precise by only relying on reference pictures.

## 5.1 Texturing the buildings

Multiple different approaches were considered when planning the texturing of the buildings. One of the major obstacles early on was the texturing techniques used on windows. The chosen method was creating geometry approximately the same size as the windows, on which the different window textures could be mapped onto. The results can be seen in Figure 12. The window textures were all lain out in rows on one texture sheet, with multiple different types of windows. The downside of this method was that the vertex count of the model was slightly increased, however the difference was minor.

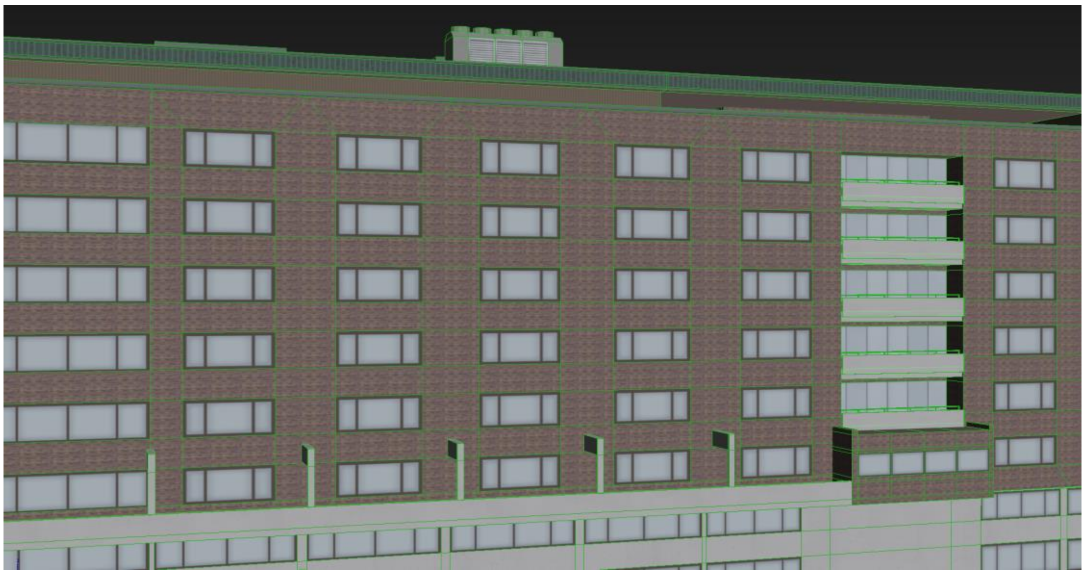


Figure 12. Geometry for windows (Kukkonen 2017)

Due to the size of the asset, it would have been inconvenient to create large textures that would cover large portions of the building. This would have required the texture files to be exceptionally large to maintain a sufficient level of detail. Additionally, it would have been a challenge to make changes to the texture, as well as align it correctly to the building were there to be changes to the geometry. Using multi-texturing to position the windows onto a separate UV channel could possibly have been a good alternative solution. In addition to giving access to easier modification and placement of the windows, this method would have allowed for a lower vertex count than the final chosen method.

As the buildings were modeled with major shapes in mind, extra geometry for the windows was added only after the main silhouettes were established. Adding geometry on models created a lot of unnecessary polygons in areas where they were not wanted. As soon as all windows were mapped onto the buildings, the buildings went through an optimization pass, in which the polygon count was reduced as much as was possible without breaking the existing texturing work.

Some textures for the buildings were created from photographs taken on location at the hospital in Kotka. The trip to the hospital was taken in October 2016. In addition to acquiring photo references, the trip provided valuable information of the area, sizes of structures and the overall scale of the elements in the scene. This information was valuable when constructing the scene in 3ds Max, as the author had real-life experience of being in the environment.

The photographs were edited in Adobe Photoshop to remove lighting problems, as well as to make them repeatable. The normal maps were created in an open source normal map baking program called xNormal. Within the program there is a tool called ‘Height map to normal map’. This tool reads the black and white data of a height map, which indicate the high and low points of the texture, and uses this information to generate a normal map. For this procedure, it was necessary to first change the texture into a black and white image, and modify the values in a way which would provide accurate results. Figure 13 provides an example of a texture created from photographs. Other textures used in the project were acquired from textures.com, an online texture database that offers royalty-free textures to be used in production (textures.com 2017). Some textures, such as the road texture and grass, were acquired from the existing texture library of Creanex Oy. The author decided to concentrate on the largest surfaces of the hospital. Many textures could be re-used in areas that required very specific textures, or existing textures could be slightly edited and repurposed to fit a specific area of the building.

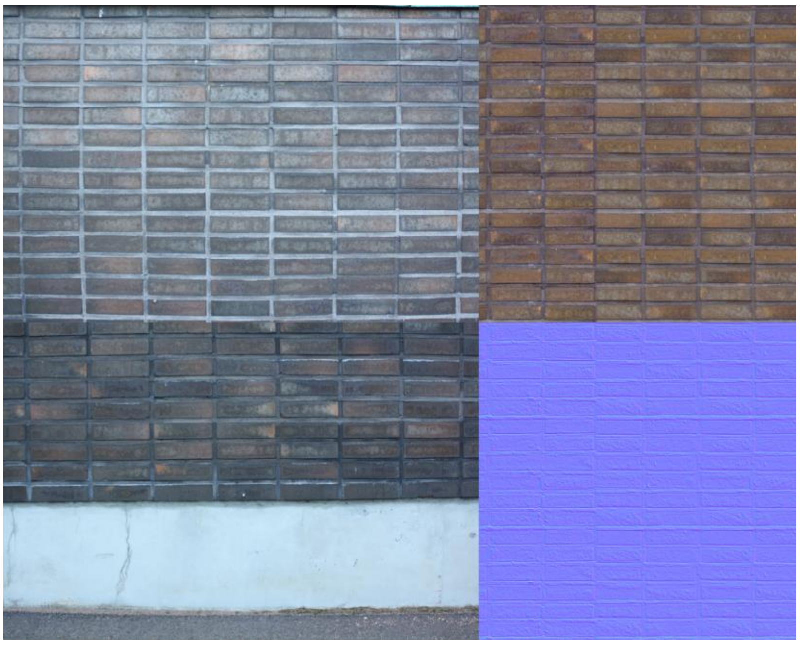


Figure 13. Unedited wall photographs (left) and final diffuse map and normal map (right) (Kukkonen 2017)

It was important to place the major textures onto the model quickly, as this helped with getting an idea of which textures were missing. While the plan was to mostly work with major textures, and copies of them, it was important to create a handful of unique textures in the form of trims. These trims would serve the purpose of breaking the uniformity caused by the repeating brick texture. They were also an important addition to the building's overall appearance, as the beams, and other trims, appeared on the real hospital. One of these trims was the green metal beam seen on the top edges of some of the buildings. The trim can be seen in Figure 14.

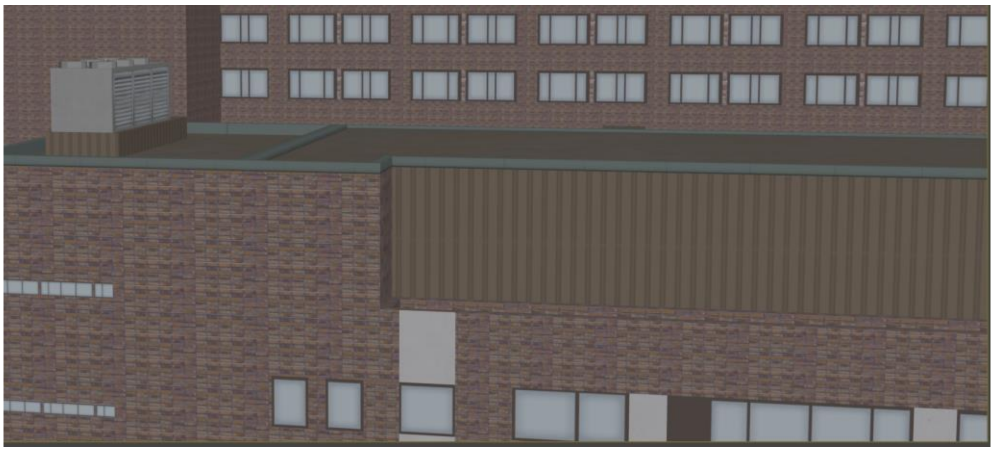


Figure 14. Metal trim (Kukkonen 2017)

## 5.2 Finalizing the hospital

The ramp in front of the hospital was one of the last things to be finalized. This part of the hospital needed to be correct, as it was a vital part of the path the ambulance would take within the simulator. As the ramp was not included in the architectural model, it needed to be created by relying on reference photos of the hospital. At this point in the process, it was time to go over hidden areas and smaller details that may have been overlooked.

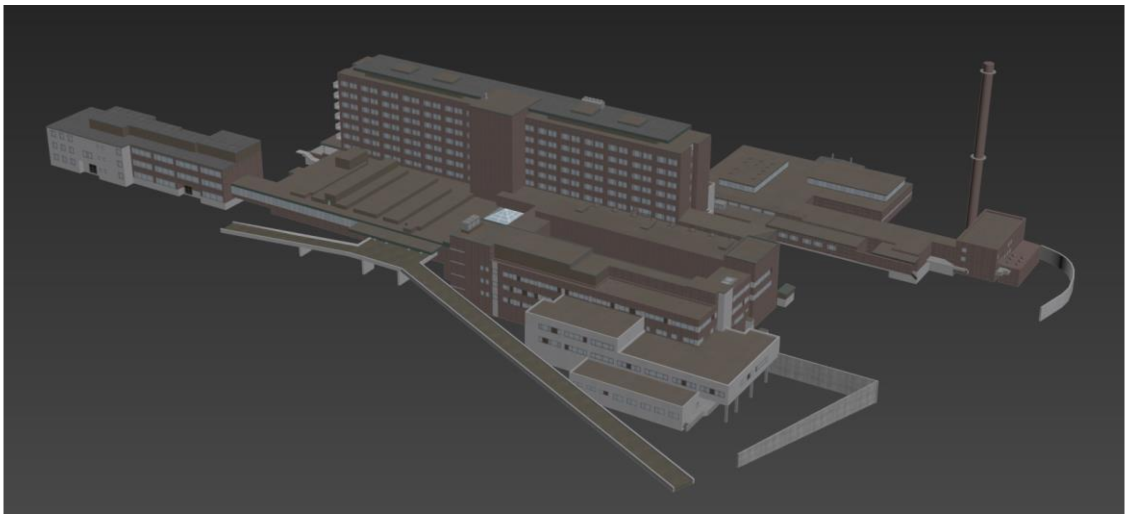


Figure 15. Finished hospital model in 3ds Max (Kukkonen 2017)

In addition, one of the buildings in the final model was not included in the architectural model, which meant that it also had to be replicated as closely as possible using reference photos. However, as it was not part of the primary buildings, deviations from the real building were unlikely to be noticed, as long as the main silhouette of the 3D building resembled the real one. The building can be seen on the left-most side of Figure 15.

# 6 IN CONCLUSION

While the author had prior experience working with many aspects of 3D-production, modeling a full environment based on a real location was a new experience. There were times during the project when it was necessary to research alternative techniques in order to get better results, however at times it was necessary to do what was possible in that situation and move to the next issue. This became more frequent towards the end of the project, when it would have been time to give all assets an additional pass of polishing.

The study showed that the effort that goes into creating a 3D environment from start to finish was ultimately underestimated. Modeling all required assets and placing all necessary elements to the scene takes time, and anything overlooked will have to be returned to later. Texturing is a phase that should not be underestimated, as good texturing can make a simple model look very impressive. However, that kind of detail comes at the cost of time. An example of a texture that could have been developed further is the window texture. While the results can be considered acceptable, the texture itself is not realistic, as it is composed of simple colors.

Most tutorials and internet resources that could be found on the subject were either based on outdated materials, required software that was not available to the author, or were only possible to produce using internal tools in other target engines, such as in Unreal Engine 4. This required flexibility regarding modeling and texturing methods, as everything had to be done with the tools included in 3ds Max. This also revealed many previously unknown aspects of the software to the author.

The usefulness of having an accurate base model to use as a guide for the buildings should not be understated. Had it been necessary to only rely on reference pictures and guesswork, the results would not have been as good as they were in the final product. Similarly, the height map used as a base for the terrain offered a great starting point for the model, despite being slightly inaccurate due to the height date of trees and buildings included in it. Existing 3D models and other resources should be used whenever possible, as multiple small assets may take an unexpectedly long amount of time to create.

Despite considering the 3D models as successful, as a 3D artist and designer, the author felt that more passes should have been taken on polishing the buildings, but most importantly there should have been clear plans and break-downs on how each piece of the buildings was to be produced. The lack of a consistent plan led to many stutters in production, which caused the author to shift concentration to other areas of the project, leaving the problem that was faced pending.

The author was personally left satisfied with the work that was done. The most important thing is that the goal of the project was achieved, which meant that the environment pieces worked well together, and created a complete, recognizable area. The big picture was successful, but to anyone looking closer, it becomes apparent that there would have been room for improvement.

## REFERENCES

Ahearn, L. 2008. 3D Game Environments. Create Professional 3D Game Worlds. Burlington: Focal Press.

Ahearn, L. 2016. 3D Game Textures: Create Professional Game Art Using Photoshop. 4th Edition. Boca Raton: CRC Press.

Autodesk. 2016a. Paint Deform Panel. WWW document. Available at: https://knowledge.autodesk.com/support/3ds-max/learn-explore/caas/CloudHelp/cloudhelp/2016/ENU/3DSMax/files/GUID-CA6D812A-C92B-4037-8810-E0257C6B61AE-htm.html

Autodesk. 2016b. Units Setup Dialog. WWW document. Available at: https://knowledge.autodesk.com/support/3ds-max/learn-explore/caas/CloudHelp/cloudhelp/2017/ENU/3DSMax/files/GUID-69E92759-6CD9-4663-B993-635D081853D2-htm.html

Autodesk. 2016c. System Unit Setup Dialog. WWW document. Available at: https://knowledge.autodesk.com/support/3ds-max/learn-explore/caas/CloudHelp/cloudhelp/2017/ENU/3DSMax/files/GUID-22D2D2B9-9A82-47FA-88DD-67B1A91E2337-htm.html

Autodesk. 2016d. PathDeform Modifier (Object Space). WWW document. Available at: https://knowledge.autodesk.com/support/3ds-max/learn-explore/caas/CloudHelp/cloudhelp/2016/ENU/3DSMax/files/GUID-04686194-95E7-4164-9EFF-B1248F320589-htm.html

Autodesk. 2016e. Object Paint Tab. WWW document. Available at: https://knowledge.autodesk.com/support/3ds-max/learn-explore/caas/CloudHelp/cloudhelp/2016/ENU/3DSMax/files/GUID-2C745521-EAD4-4A9E-A2CF-7A4B6DDC486D-htm.html

Autodesk. 2014a. Drag and Conform Tools (PolyDraw Panel). WWW document. Available at: https://knowledge.autodesk.com/support/3ds-max/learn-explore/caas/CloudHelp/cloudhelp/2015/ENU/3DSMax/files/GUID-96452030-B2E8-404A-AF0E-4EB70EB44EE6-htm.html

Autodesk. 2014b. Displace Modifier. WWW document. Available at: https://knowledge.autodesk.com/support/3ds-max/learn-explore/caas/CloudHelp/cloudhelp/2015/ENU/3DSMax/files/GUID-12F0E0BA-3FCA-4C8F-8C9B-C77C8FB4A45C-htm.html [Accessed 10 April 2017]

Autodesk. 2014c. Line Spline. WWW document. Available at: https://knowledge.autodesk.com/support/3ds-max/learn-explore/caas/CloudHelp/cloudhelp/2015/ENU/3DSMax/files/GUID-1408D223-7A05-4163-A7FA-5314DF5352BE-htm.html [Accessed 10 April 2017]

Becker, K., Parker, J.R. 2011.The Guide to Computer Simulations and Games (1). Ebook. Indianapolis: John Wiley & Sons, Incorporated. Available at: http://site.ebrary.com/lib/alltitles/home.action [Accessed 5 April 2017]

Blevins, N. 2012. Clumping or Grouping. WWW document. Available at: http://www.neilblevins.com/cg_education/clumping/clumping.htm [Accessed 9 April 2017]

Chopine, Ami. 2011. 3D Art Essentials. Burlington: Elsevier Inc.

handplane3d, 2013. Controlling Shading Behavior. Video clip. Available at: https://www.youtube.com/watch?v=ciXTyOOnBZQ [Accessed 8 April 2017]

Ogre3D. 2014. -Material. WWW document. Available at: http://www.ogre3d.org/tikiwiki/-Material [Accessed 5 April 2017]

Pixologic. 2016. Displacement Maps. WWW document. Available at: http://docs.pixologic.com/user-guide/3d-modeling/exporting-your-model/displacement-maps/ [Accessed: 5 March 2017]

Polycount, 2016a. Polygon Count. WWW document. Available at: http://wiki.polycount.com/wiki/Polygon_Count [Accessed 1 April 2017]

Polycount, 2016b. Texturing. WWW page. Available at: http://wiki.polycount.com/wiki/Texturing [Accessed 5 April 2017]

Polycount, 2016c. DDS. WWW document. Available at: http://wiki.polycount.com/wiki/DDS [Accessed 5 April 2017]

Polycount, 2016d. Texture Coordinates. WWW document. Available at: http://wiki.polycount.com/wiki/Texture_Coordinates [Accessed 5 April 2017]

Polycount, 2015. Decal. WWW document. Available at: http://wiki.polycount.com/wiki/Decal [Accessed 7 April 2017]

Rani, K. Aava. 2014. Learning Unity Physics. Ebook. Birmingham: Packt Publishing Ltd. Available at: http://site.ebrary.com/lib/alltitles/home.action [Accessed 5 April 2017]

Raczynski, S. 2006. Modeling and Simulation : The Computer Science of Illusion. Ebook. Chichester: John Wiley & Sons, Incorporated. Available at: http://site.ebrary.com/lib/alltitles/home.action [Accessed 5 April 2017]

Simonds, Ben. 2013. Blender Master Class: A Hands-on Guide to Modeling, Sculpting, Materials. San Francisco: No Starch Press, Inc.

Textures.com, 2017. Frequently Asked Questions. WWW document. Available at: http://www.textures.com/faq.html

Unity3D. 2017. Normal map (Bump mapping). WWW document. Available at: https://docs.unity3d.com/Manual/StandardShaderMaterialParameterNormalMap.html¨ [Accessed 5 April 2017]

Epic Games, 2012. Terrain Advanced Textures. WWW document. Available at: https://docs.unrealengine.com/udk/Three/TerrainAdvancedTextures.html#Multi-UV%20Mixing:%20Reducing%20tiling%20through%20scalar%20mixing [Accessed 7 April 2017]

Epic Games. 2017. 1.4 – Roughness. WWW document. Available at: https://docs.unrealengine.com/latest/INT/Resources/ContentExamples/MaterialNodes/1_4/ [Accessed 9 April 2017]

Wagstaff, S., Derakhshani, D. 2006. Getting a Job in Computer Graphics : Real Advice from Reel People (1). Ebook. Alameda: Sybex. Available at: http://site.ebrary.com/lib/alltitles/home.action [Accessed 5 April 2017]

## LIST OF FIGURES