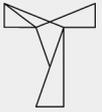


TYGRON

Tygron Model Pipeline Manual

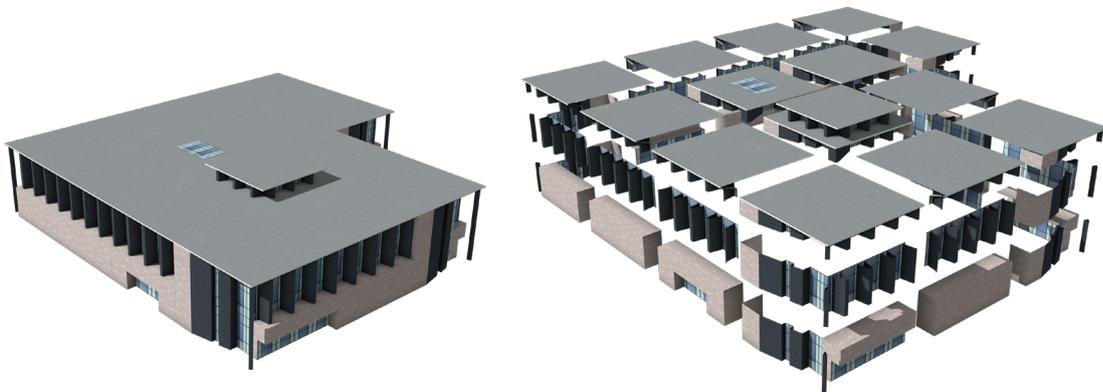


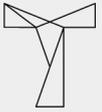


TYGRON

Contents

 Introduction	3
 Tiles & Grid	4
 Building Blocks	4
 Asset Types	8
 Naming Conventions	13
 Model Requirements	14
 Texture Requirements	16
 Finalising	18
 Exporting	21
 Blender Quick Start	31





TYGRON

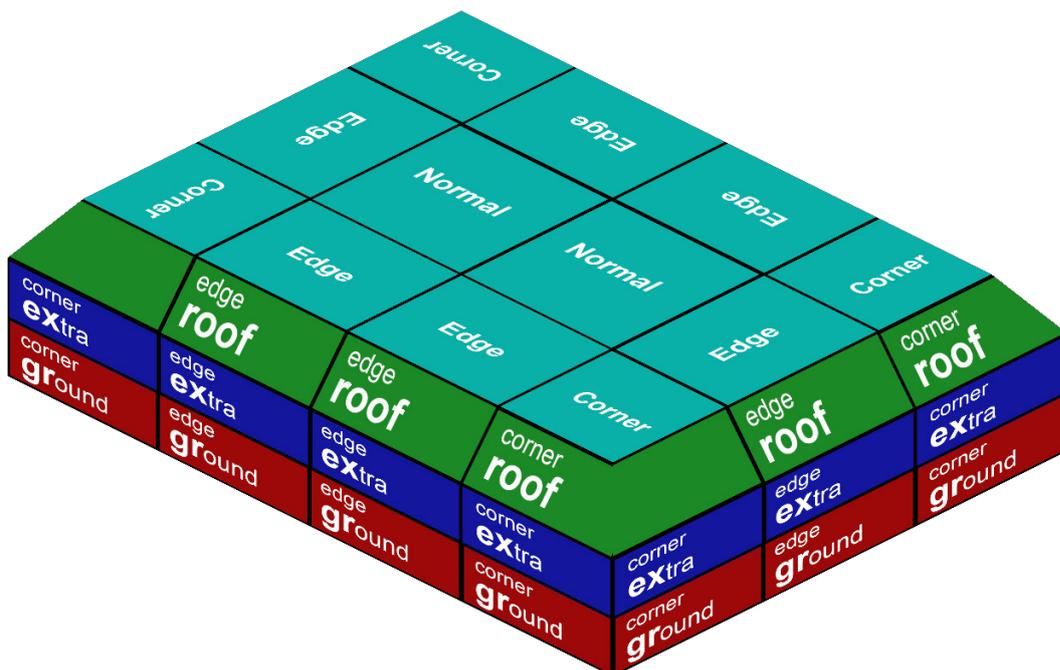
Introduction

To make the best use of models and graphics in an organised way, Tygron uses a grid based system for buildings and various other assets. Every model in the engine is modular and aligned with this grid, making it fit perfectly with the surrounding assets. By using this system, every selected area can be loaded and drawn without problems.

Because Tygron's way of making models can be very different than with another engine, this manual should provide you with the necessary details.

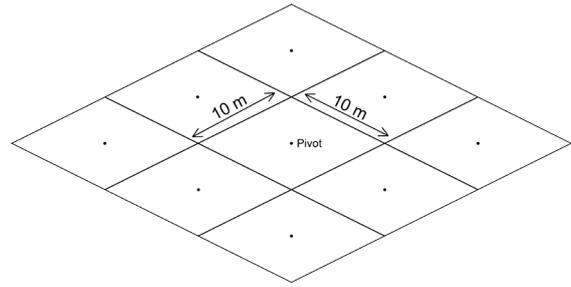
This manual is written to let you get familiar with models inside the Tygron engine. You will find how every model is made to work with the engine, how the grid works, all things to keep in mind and the possible pitfalls that you could encounter.

After you've read this manual, you should have a basic idea how the Tygron Engine handles its models. You know how to make models that are applicable with the engine and how to name and export them into the engine. You know what to avoid when making a model and what to keep in mind when finished with a model.

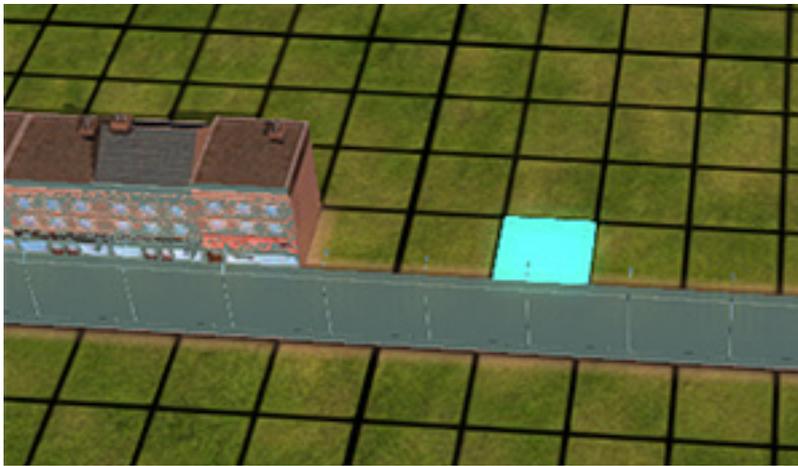


Tiles & Grid

The grid in the Tygron engine uses tiles of 10x10 metres. Every asset that is put into the engine must conform to these tiles.

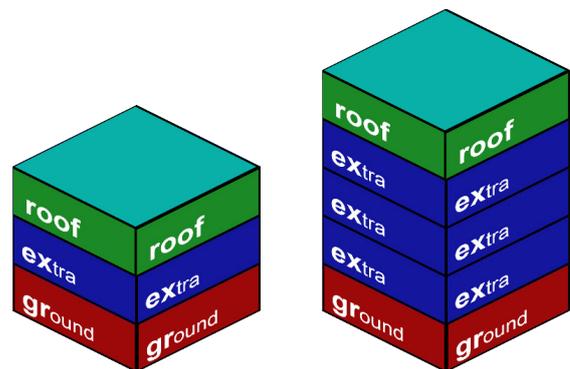


A single asset can be a minimum of 10x10 meters or several tiles wide and deep as long as they are square and have an uneven number of tiles. So an asset can be 3x3 tiles (30x30 metres), 5x5 tiles (50x50 metres) or any increasing uneven number of tiles. This is so that the pivot of the asset is always in the middle of the tile it sits on.



Building Blocks

In the Tygron engine most of the buildings consist of individual puzzle pieces that neatly fit together to form a complete whole. There are different types of tiles used for different purposes and different types of buildings that use a different set of tiles.



The tiles are separated both vertically and horizontally.

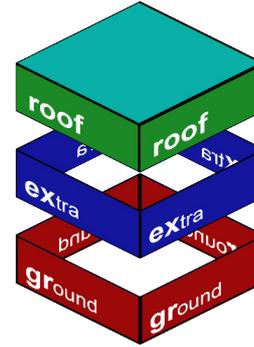
Vertically the building blocks are divided into three types. The ground level, the extra level and the roof level. Each level represents the storey in a building. Each level is generally 3,2 metres high.

Ground level

The ground level contains the base of the building and is always placed on the underlying terrain in the engine.

Extra level

The extra level is used to fill in the space between the ground and roof levels. This layer is used to be able to create buildings of all heights. The extra level is duplicated when an extra storey is needed in the building.



Roof level

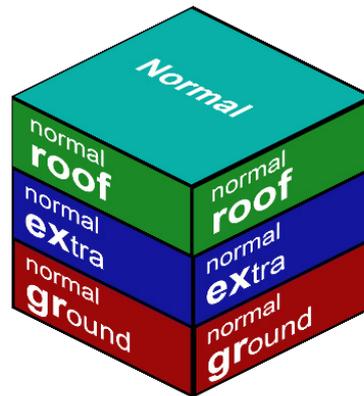
The roof level sits on top of the extra level to cap it off. The roof level can, but doesn't need to, represent a storey.

Block types

The blocks are also separated horizontally. These include the Normal, Edge, Corner, Multiple and Diagonal block types. Each of these types also have ground, extra and roof levels.

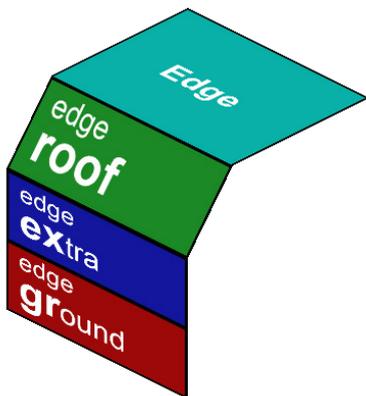
Normal

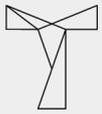
The Normal block type is the standard block type. It has geometry on all sides and is used in all instances where none of the other types of blocks are used. This can be alone standing or clustered together to fill in larger areas.



Edge

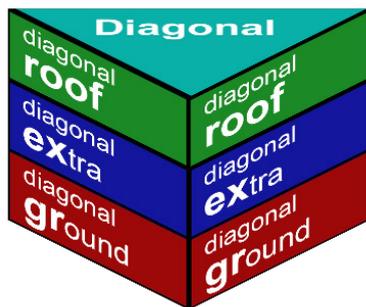
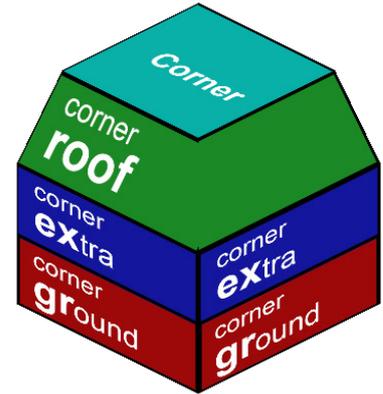
The Edge block only has one side. It is only used on the sides of buildings and structures.





Corner

The Corner has two sides. It is used just for capping off the corners of buildings and structures.

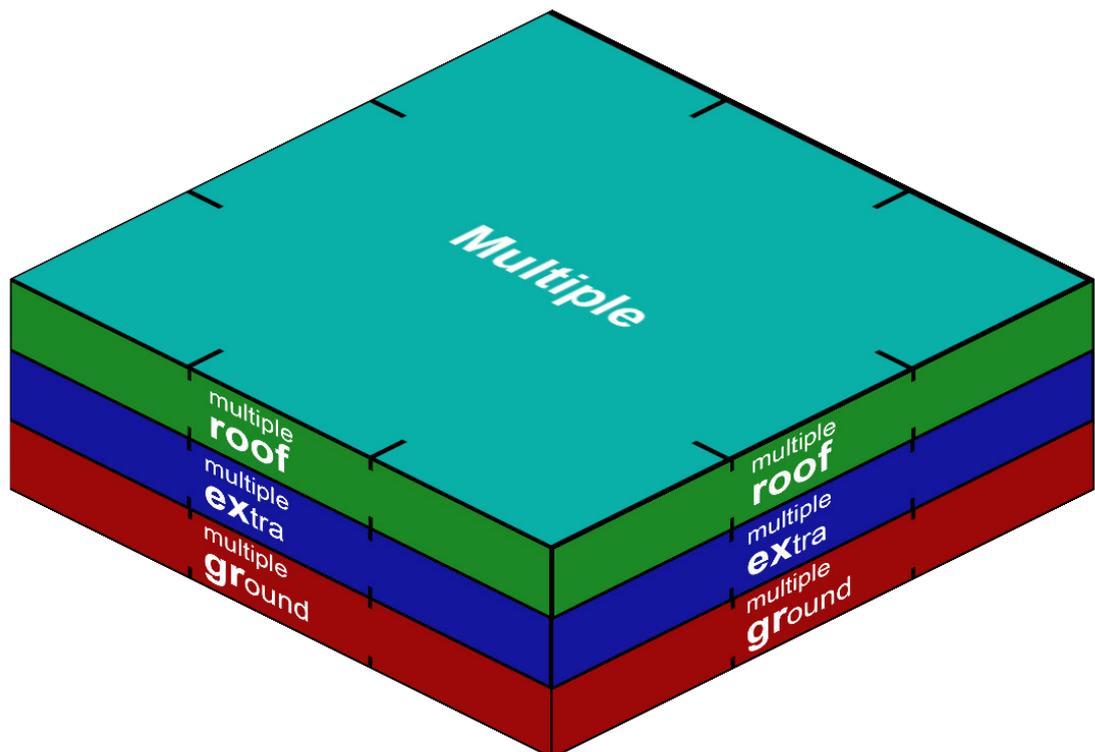


Diagonal

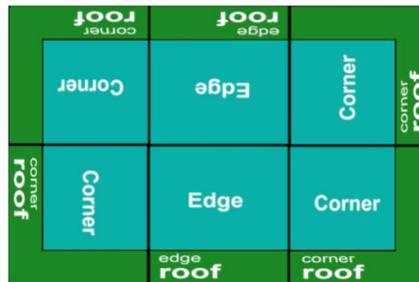
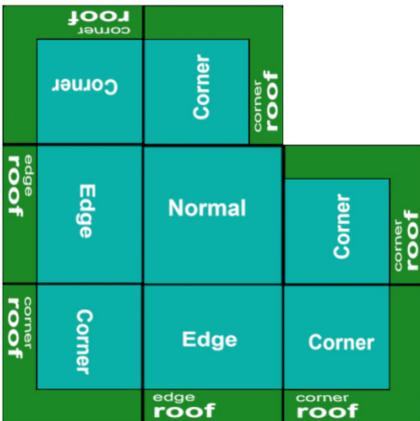
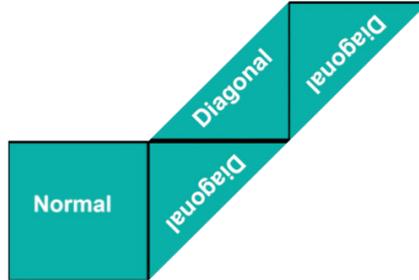
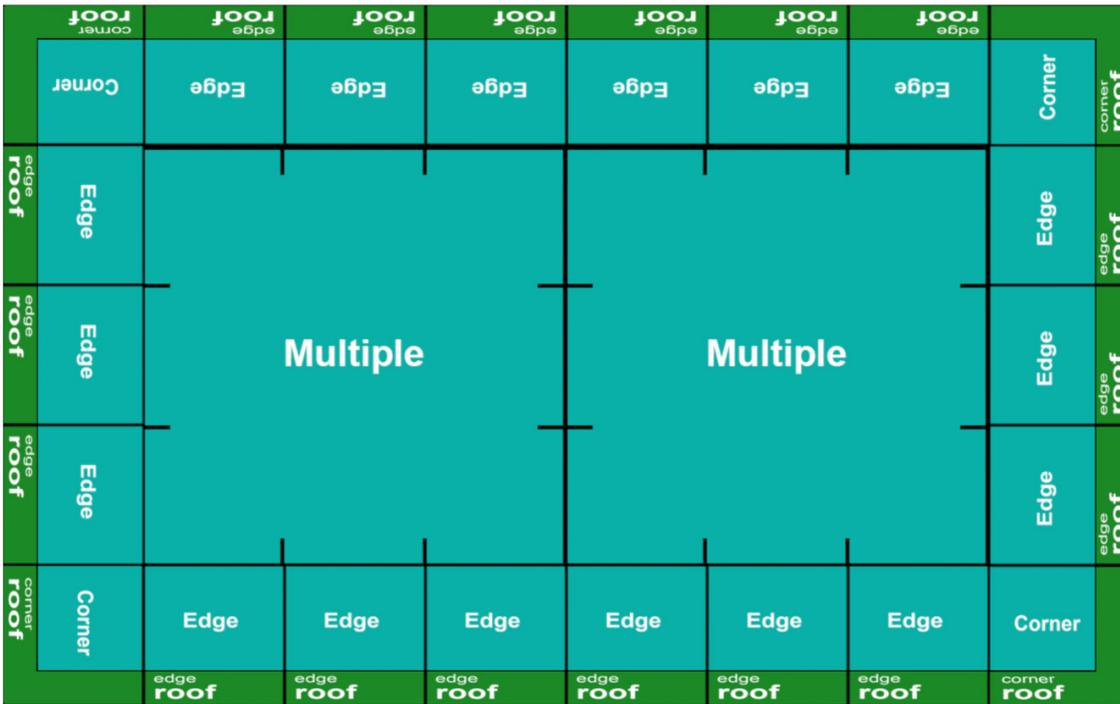
The Diagonal block has three sides. It is basically half of a Normal block cut diagonally. It is used for diagonal parts of a building or structure. It can also sometimes function as a Corner except that it leaves room for other things on the open half of the tile it occupies.

Multiple

The Multiple block type is mostly the same as the Normal type. Except that it's several blocks wide and deep. It can be a minimum of 3 by 3 blocks in size. There is no absolute maximum as long as the number increase in an uneven number. Though Multiple blocks of larger than 3x3 are rarely used.



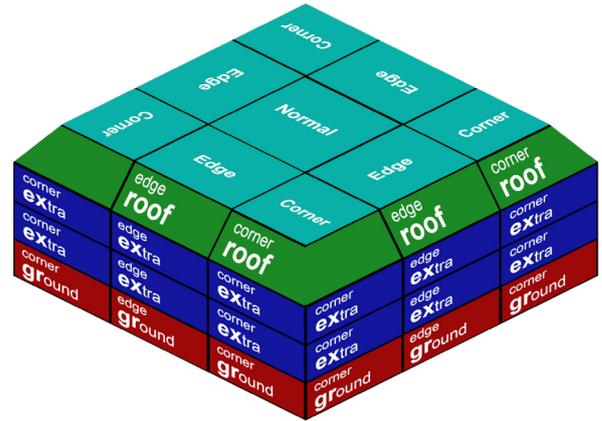
Construction examples



Asset Types

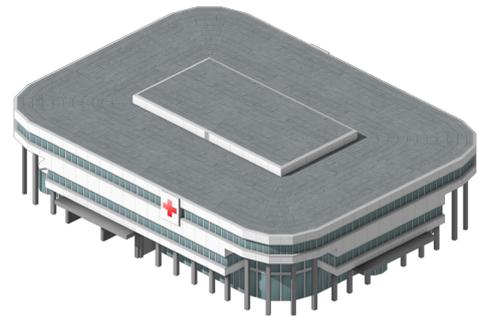
Highrise Building

The highrise building type can be pretty much anything, from a garage to a skyscraper. These types of buildings form the bulk of the assets in the engine. They are defined by utilising all of the available different building blocks



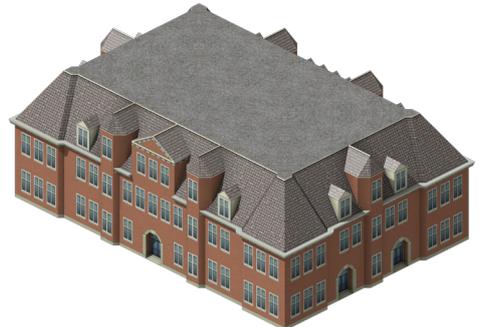
Normal

- ground level
- extra level (multipliable)
- roof level



Edge

- ground level
- extra level (multipliable)
- roof level

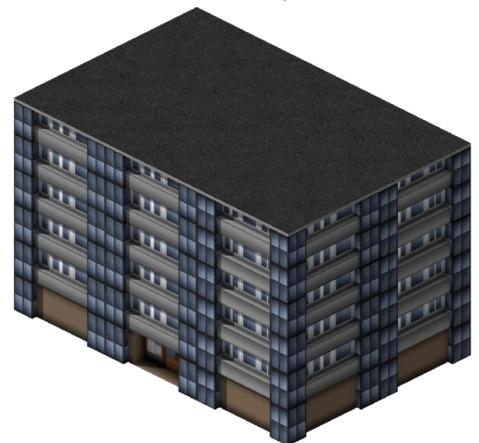


Corner

- ground level
- extra level (multipliable)
- roof level

Multiple (3x3, 5x5, 7x7, 9x9)

- ground level
- extra level (multipliable)
- roof level

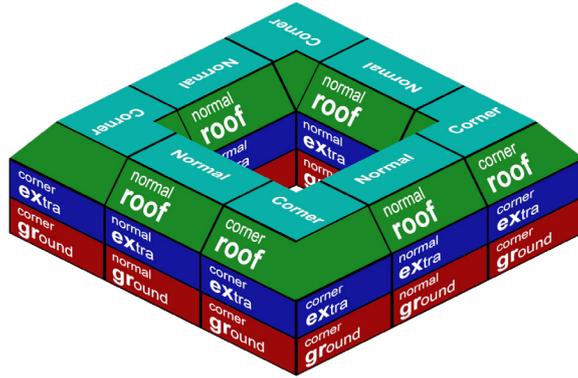


Diagonal

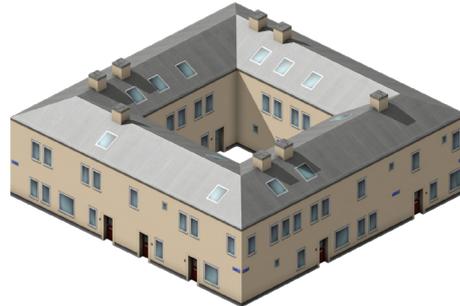
- ground level
- extra level (multipliable)
- roof level

Row House (one deep)

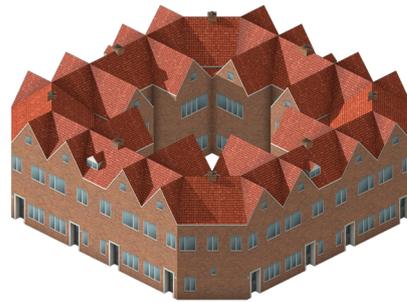
Row Houses are characterised by the building being only one deep. the Normal block on this building also fullfills the function of the Edge block. Back gardens are also part of the same function and are placed behind the houses. Though they are separate models.



- Normal (edge)
 - ground level
 - extra level (multipliable)
 - roof level



- Corner
 - ground level
 - extra level (multipliable)
 - roof level



- Diagonal
 - ground level
 - extra level (multipliable)
 - roof level

Complete (standalone building)

A Complete is a standalone building that occupies only one tile on the grid. Though it can rise from the ground, it only has a Normal block type. Generally Normal block types need to fit tightly inside the tile so they can connect with other adjacent Normal blocks. But in this case the Normal block does not need to connect with the four sides of the tile.

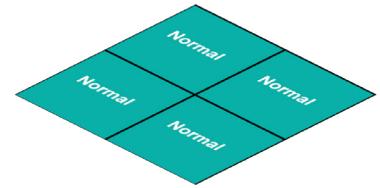


- Normal
 - ground level
 - extra level (multipliable)
 - roof level



Furniture

Furniture differs from the standard assets in not having any extra and roof levels. Furniture is placed on top of other assets as extra details (such as roof details). Often this is used as detailing for roofs and such.

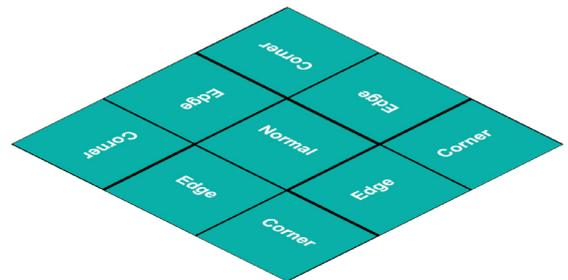


Normal
ground level

Multiple (3x3, 5x5, 7x7, 9x9)
ground level

Large Gardens

These types of gardens use all block types except that they only use ground levels.

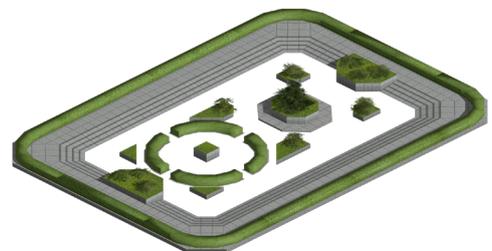


Normal
ground level

Edge
ground level

Corner
ground level

Multiple (3x3, 5x5, 7x7, 9x9)
ground level



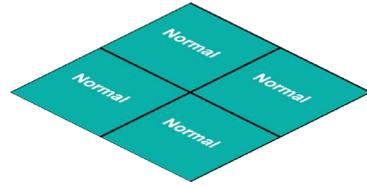
Diagonal
ground level



Generic

Generic asset types are the most basic of grid based assets. They lack Edges and Corners.

Normal
ground level



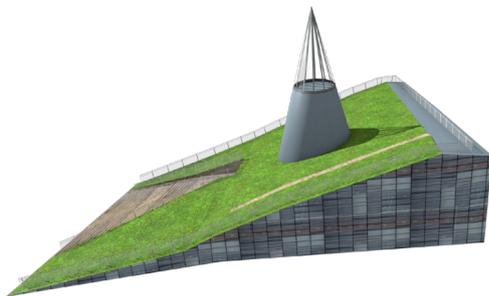
Multiple (3x3, 5x5, 7x7, 9x9)
ground level

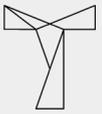


Diagonal
ground level

Landmarks

Landmarks are models that don't conform to many of the standard rules of the grid system. Landmarks are generally unique buildings that stand on their own. Therefore they don't have to be stackable or be adjusted to tightly fit the grid.





Block Variations

Beyond the different types of building blocks all assets are made up of. There can also be variations within a block type. Depending on the desired result, a block type can have any number of variations.

The Tygron engine will pick a random variation within each block type to create buildings that vary from place to place. One Edge block can have a solid wall while another can have a doorway in it for example. Both are interchangeably applicable as an Edge block wherever this is appropriate.



Naming Conventions

Each separate block piece is given a specific name so that the system can find it and place it where it belongs. The name starts with the actual name of the object. Be this a villa or a hospital. Longer names can also be divided by an underscore. After the object name comes the block type such as normal or edge. Then comes the level. At the end the variation type is identified with a number beginning with the number 1. Each part is divided by an underscore.

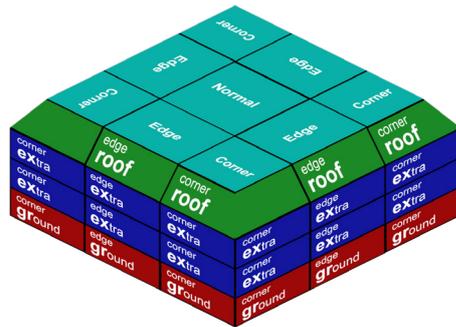
There is an exception for the Furniture asset types and any other types that do not have an Extra and Roof level. For Furniture, the ground level name is replaced with the name 'furniture'. For assets that lack Extra and Roof levels, the name for the level is removed from the asset name.

Naming order

name_type_level_number

Block type names

- normal
- edge
- corner
- diagonal
- multiple

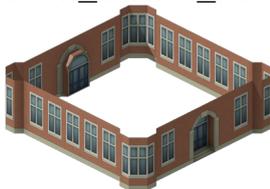


Level names

- roof = roof level
- ex = extra level
- gr = ground level

Examples:

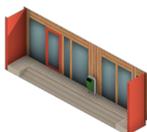
education_classic_normal_gr_1



garden_normal_furniture_4



sports_center_edge_gr_2



hospital_corner_ex_1

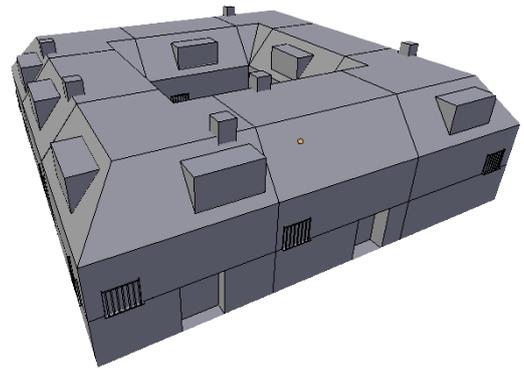


Model Requirements

The models for the Tygron Engine need to adapt to the requirements. These requirements are here to make sure the engine will run on the supported operating systems and/or hardware.

Polycount

The models should be as 'low poly' as possible. The more vertices you have, the more physical memory the engine requires to run. If you can keep the polycount as low as possible, you can load more buildings without dramatic framerate drop. Its best to find a good balance between a low polycount and enough details.

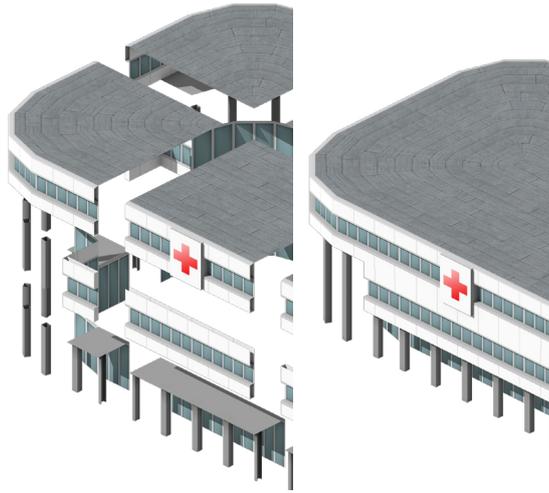


LOD

The LOD, or Level Of Detail, is an engine generated function wich helps improve the performance. This means it doesn't have to be made by hand. When zoomed out far enough, the Level of Detail function will cut away vertices close together, since it should be impossible to see the difference from far away. This is a function to keep in mind. For example, a building with a window extruded a little inside will lose it's frame and possibly make a hole in your model. Holes in models are not good since they're visible from high distance. When modeling your geometry, keep in mind that small details are probably suitable to be put in textures rather than in actual model geometry.

Sizes and Alignment

The model needs to be the correct size for the engine to place them properly. If the sizes are not correct, it will result in a building that has intersecting parts or leaves giant holes between its building blocks. The tiles of the Edge and Corner are 10x10 metres. The facades of these building blocks can deviate from being on the exact edge of the tile, to create interesting geometry. But the connecting sides of the models need to be on the exact edge of the tile or they will not fit together properly. The Normal has to be 10x10 metres since it also needs to fit in the middle of a building to cap the holes. The Multiple is 30x30 metres or bigger, increasing with uneven set of tiles. It needs to fit exactly within these sizes for the same reason as the Normal.



Texture Requirements

To make your models look like the actual things, you make use of textures. The textures in the Tygron engine have a few rules to make them work properly.

Resolution

The resolution of the textures use “the power of two”. It is recommended to combine textures of models in the same category (industrial, gardens, etc.) into one large texture. This doesn’t affect the resolution but it takes less memory since it’s a single image file having to load instead of multiple textures.

Shaders

Generally only a diffuse map is required. However, for buildings we also make use of specular maps mainly for the windows. No other shaders are used. So depending on the model it is advised to also add a specular map along with a diffuse in the .PSD file.



UVs

The way of mapping is all up to you, but we will give a few tips to make the most use of the texture space. Because buildings and models might use the same texture, it is recommended to use overlapping UVs. If every face has it's own unique UV space, using the same texture, it's a waste of space because you can use it more efficiently. When using overlapping UVs, the saved space can be used for different variations and your materials can get more quality.



Variations

Variations are not required, but can prevent a model from being a little boring or looking repeatable. Variations does not necessarily have to be different models, it can also be different UV positions for existing models. Using these techniques, an asset never feels the same when it's built in different places.

Finalizing

When you think you are completely ready with the model and the textures, you can start putting it through a final checklist.

Checklist

First, every part of the building needs to be checked if any vertices can be removed or merged together, without damaging your model. Every vertex counts. Make sure that there are no holes in the visible sides.

When you're sure the model is entirely clean, check your textures. The right size, the right format (.PSD). Make sure your UVs match up and don't leave any seams on the model.

Make sure all the different building blocks have their pivots are correctly set at the middle of their tile. It is recommended to use the building blocks you've made to create a mockup building in complex shapes to see how the blocks connect together.

Pitfalls

The pitfalls are the common mistakes you could encounter during the creation of a model. Please check the above list to save yourself from lots of trouble. Here are some examples of problems we've experienced.

Holes - geometry

When making a model that is not exactly matched up with the edges of the grid, we found holes in our models. Make very sure that in every possible position, it connects with the other building blocks seamlessly.



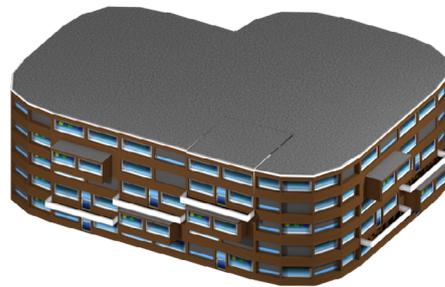
Holes - LOD

When creating your geometry, make sure that you don't leave vertices too close to each other. The LOD will merge or delete these vertices which can also create holes. If you have no choice, you can try placing a plane behind it. When the LOD eats the vertices, a plane showing the same texture as the deleted face(s) will mask the holes from great distances.

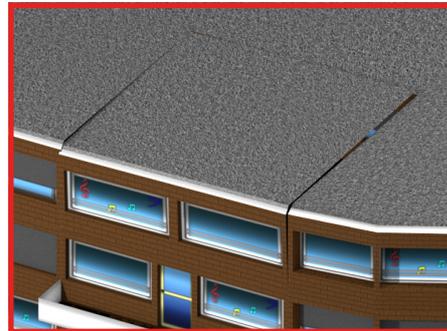


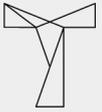
Alignments

To make sure your building blocks are going to fit together seamlessly, you must make sure the alignments are perfect. If a building block is not perfectly aligned, you'll see it being placed incorrectly immediately after you put together a building.



Check if every vertex on the bottom of a building block is exactly at 0.000 metres on the vertical axis. Check if every vertex on the sides are exactly at the edge of the tile they're on and check every top vertex (only on ground and extra levels) if they are exactly at 3,2 metres height. If these are all correct, make sure the building blocks all have their pivots on the centre of the tile, at 0.000 height.



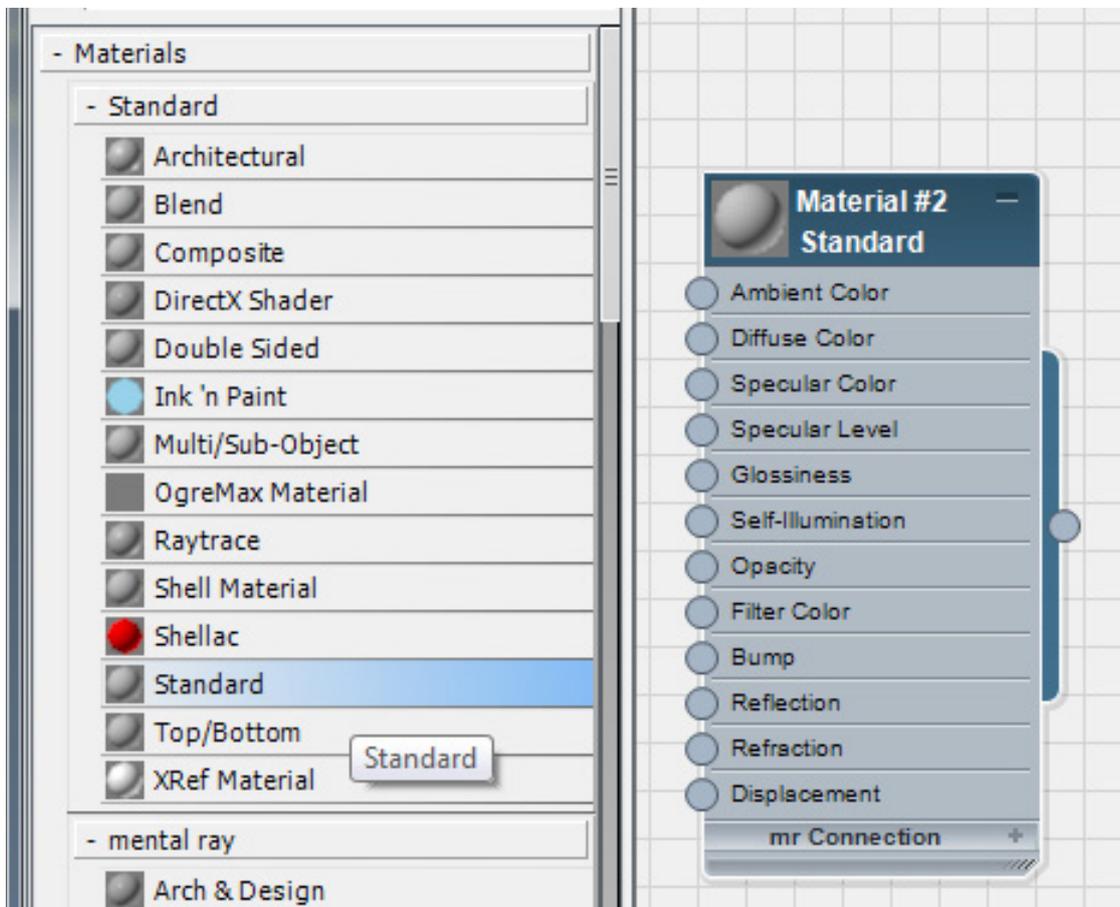


TYGRON

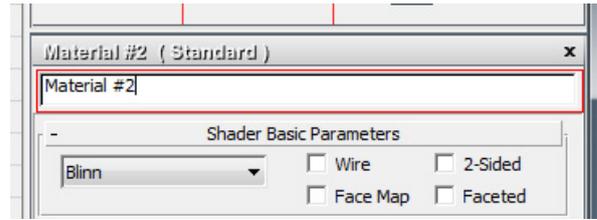
Exporting

So now that we have a model which is unwrapped, and a texture map accompanying it, we should make a material for it. From this point on, the process will continue in 3DS Max because of the OGREMAX plugin which we are using. OGREMAX exporter works for 3DSMax, Maya and Softimage. At Tygron we are using 3DSMax as our foremost 3D modeller and animator so we will continue with this product. If you want more info about the OGREMAX plugin then visit: <http://www.ogremax.com/>

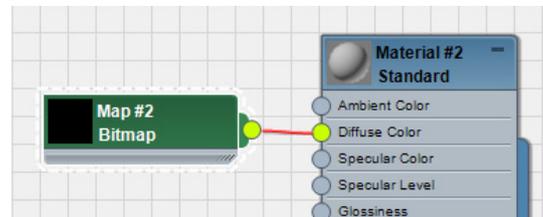
The setup which we will be showing is fairly basic, but there are some minor details we should be looking at. Go to 3DSMax and open your material editor, which can be found at the following location: Rendering -> Material Editor -> Slate Material Editor, or just press M on your keyboard. The material editor is node based and it's a good idea to get used to this. On the left side of the editor, you can see the Material/Map Browser which displays all the materials, maps and presets which are available to you. This list can vary depending on your chosen renderer. We should only be concerned with the standard list of materials. The standard list contains materials with various names, but the one which we will always be using is the Standard material.



Select the standard material and drag it to the window next to it. A new Standard material will be created, which is named Material # 2 in our case. Double click on the material node and notice that on your right side of the editor, a properties window is opened of the material you have just created. Go ahead and rename it to an appropriate name for your Asset. This is very important, because this name will be used to identify the material later on in OGREMAX. You can rename your material by clicking in the box above the properties dialog and typing in your new name. Rename the material to "american_house".

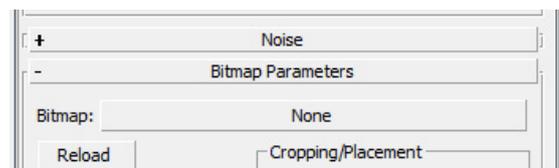


Now, we have to add the texture map, that we created earlier, to the material. This texture must be connected to the Diffuse slot of the material, so that the exporter can recognise it.



Go back to your Material/Map browser on the left side, and scroll down to Maps – Standard and select Bitmap. Drag it to the Diffuse color slot on your material, and notice that the connection turns to green. This means that it's a valid connection.

Release it, and a pop up windows browser will appear. Find your texture map in the OGREMAX folder, which should also be properly named. If no pop up window appears, double-click on the Bitmap map you have just connected, and the properties for this map will appear on you right side. You can add a texture under Bitmap Parameters next to Bitmap. This completes the Material setup.



There is one more thing we should do before starting to export. We should make a reflection map so that a nice reflection appears on the windows and other reflective objects on the model when it is placed inside our editor.

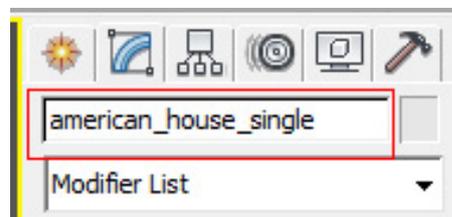
Save the specular image to the OGREMAX folder you created earlier, and make sure that it is named exactly the same as your diffuse texture but with one exception. You should put `_specular` at the end. Save the image as a `.jpg`. This is the only format which we will be using from now on (The textures in the editor are `.dds` but that will be explained later). For example, if you have named your image `house.jpg`, the resulting reflection map should be named `house_specular.jpg`. The reason for this, is that our Tygron OGREmodel Converter (which we will show later) can recognise which specular is part of which diffuse map. This is very important and it could save you several days of frustration. Another thing which is very important, is that everything you make should be given a name with a lower case. This includes extensions.

When saving your texture from Photoshop, you should always check the Use Lower Case Extension box so that there are no problems down the line.

So now we have 2 textures in our OGREMAX folder, our material is properly set up with the reference for the diffuse texture, we are ready to export the model.

Final Checks

Go back to 3DSMax and make some final checks to the model. Check if the model is properly named. Especially if it is imported from another package. Rename your model by selecting it, and entering a new name for the Asset. This is also important, because it will be transferred to the editor, so make sure it is something that is an obvious name for the Asset you have built. The name box is located in the upper right corner of the command panel.

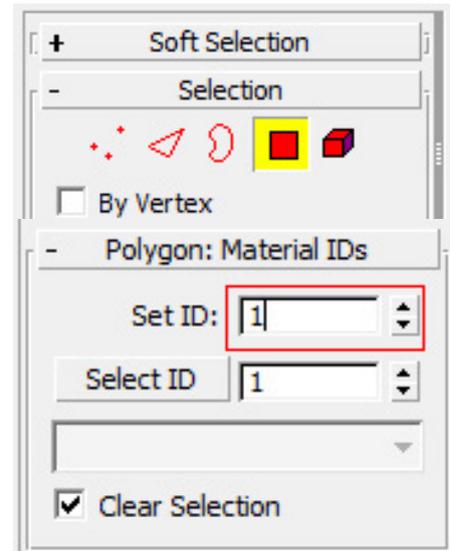


Editable Poly

Make sure that your model is Editable Poly and not Editable Mesh. To convert a model to Editable Poly, select it and right-mouse click to activate a menu. In the bottom right side is a menu named Convert to... Hover over it and a list of options will appear. Select Editable Poly and you're done.

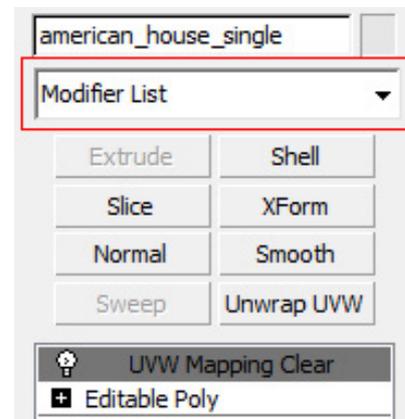
Material ID

Another thing to check is how many material IDs the model has. This is a 3DSMax internal structure which enables every polygon on the model to have a different material and therefore a different ID. Our editor and converter only supports 1 ID, so we must check if this is really so. This is more of an issue if the model is imported from other packages, so make sure it is correct. The way to do this easily is to select the model, go into polygon mode and with CTRL+A select every polygon on the model. The whole model turns red, which tells you that the polygons are selected. Scroll down to the model properties on the right side, go to the Polygon: Material IDs section and type in 1 in the Set ID: box. This turns all the selected polygons to the default ID of 1 which they should have.



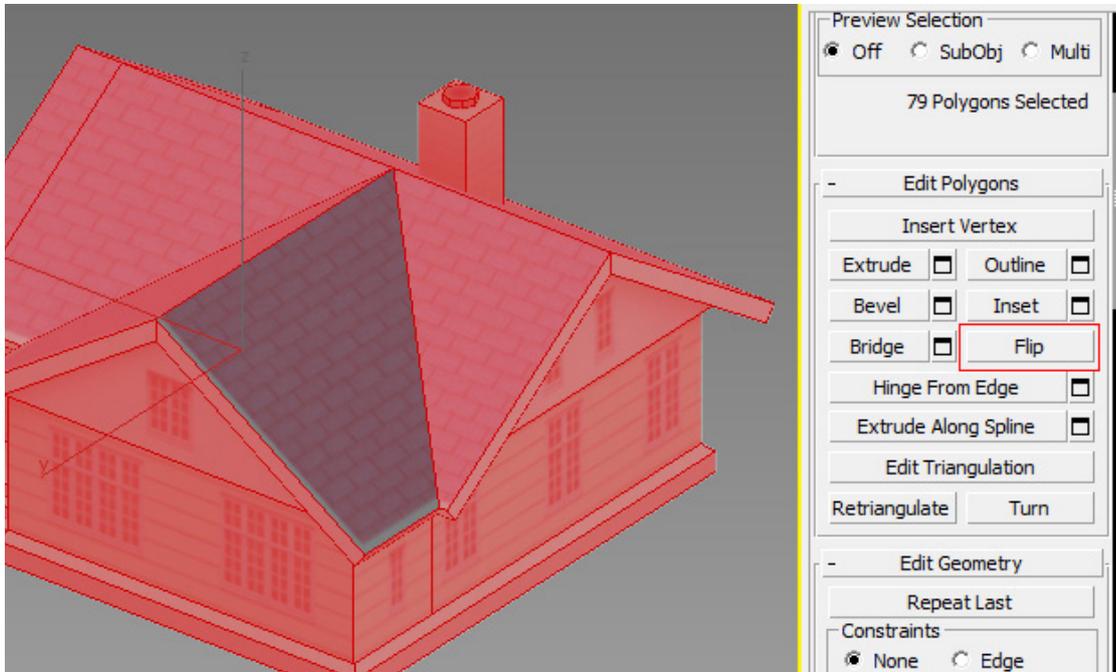
UV Channels

Another thing to make sure, is that there is only 1 UV Channel active on the model. Max supports multiple channels on a single model, but this will not work for our exporter. The way to remove these extra channels, is by adding a UVW Mapping Clear Modifier to the model and entering 2 or more in the Map Channel: box. Never type in 1, because that is your current channel which you are using. Our concern is to remove any extra channels. This is how you add the modifier: Select the model, and from the Modifier list dropdown in the upper right corner, select UVW Mapping Clear.



Normals

Another thing to check, is if all the normals are facing the correct way. This can be easily checked by selecting the model and going into polygon mode again. CTRL+A alike before to select every polygon. You can spot the flipped normals by looking for darker ones as seen below

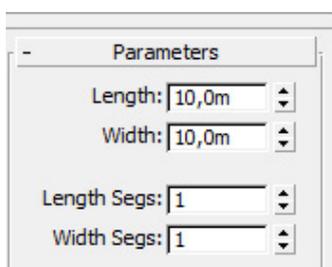
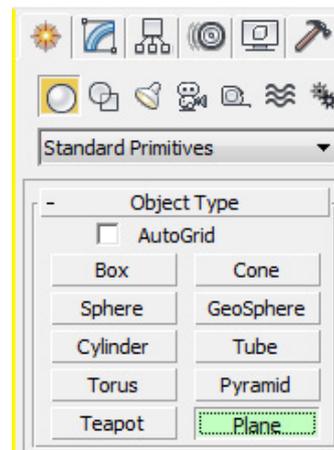


It is fairly obvious which polygon is flipped here. Select the bad polygon and press the Flip button on the right side. It is located under the Edit Polygons section.

Model Scale

Scale is always an issue when a model gets imported from different packages. Different programmes use different systems, so we have to make sure that our model will end up the correct size. The first thing to setup is the units in 3DSMax. Go to Customize -> Units Setup.

This will open a dialog. Check under Display Unit Scale the Metric checkbox. Make sure the drop-down is set to Metres. That's it, you're done. Make

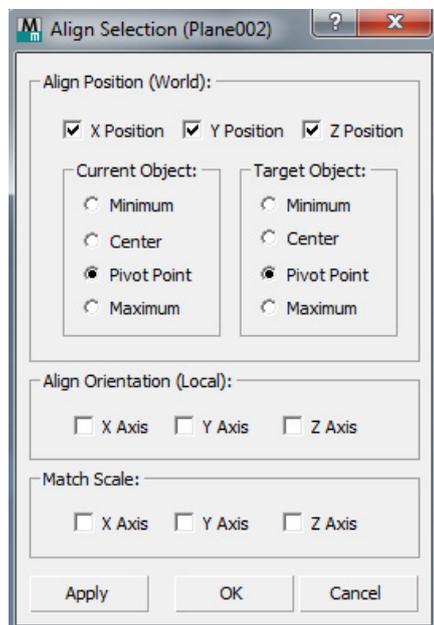
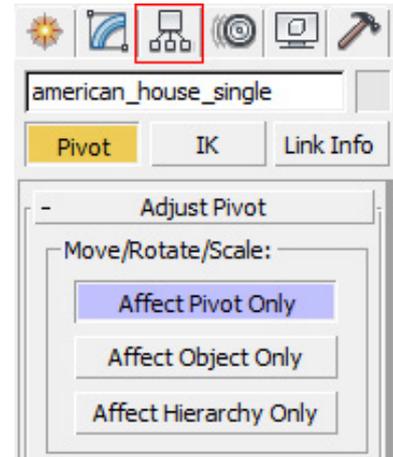


a primitive plane, and make sure it is 10x10 metres by entering 10 into the length and width parameters. Put your model over it, and make sure it fits inside the tile. If it's too small or too big, then scale it to the correct size with the scale tool.

Pivot

Align the pivot to the tiles pivot, so that our model will fit properly onto our editor grid. It is always centered onto the tile. You can change an objects pivot by doing the following:

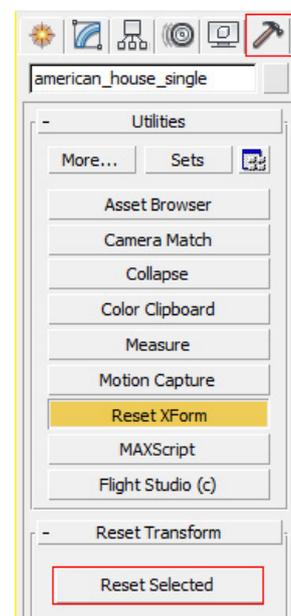
Select the model, and go to the Hierarchy tab in the upper right corner. Select Affect Pivot Only. This enables you to move the pivot without the object. Press ALT+A (which activates the Align tool). Select the Plane object and a dialog will appear. Check the following boxes:



This will make sure that your model's pivot is centered correctly for export. Exit the Affect Pivot Only mode.

Freezing Transformations

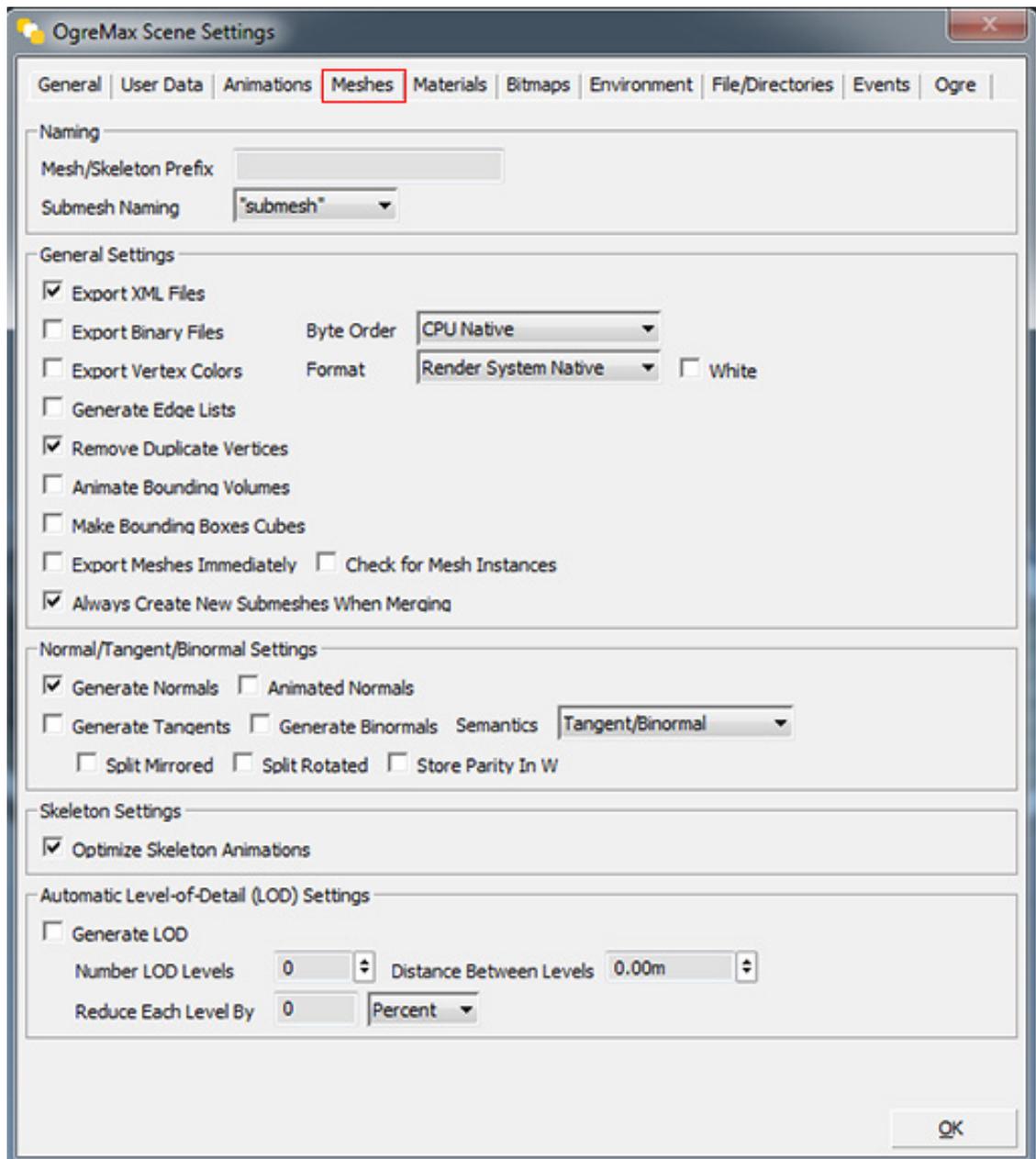
One final thing must be done so that the model exports correctly. We must freeze all the transform, which we have done to the model. This can be done by the Reset Xform Command. Go to the Utilities Tab and select Reset Xform. Click on Reset Selected which appears at the bottom. This will create an Xform modifier, and thereby freezing every transform to it. Convert the model to Editable Poly like previously so that the Xform is collapsed into the model. We are now ready to export the model with the OGREMAX plugin.



Exporting with OGREMAX.

So now we are finally ready to start exporting our Asset to an OGREMAX XML file, so that it can be converted with our OGREMODELCONVERTER. After you have installed the OGREMAX plugin for 3DSMax (Most Max versions at Tygron have this plugin already.), the dropdown menu for the OGREMAX exporter appears in the menu bar at the top. It is usually located between Maxscript and the Help menus. Click on OGREMAX, and a dropdown menu appears. There is a whole list of options, but we will be only concerned with 2 of them. These are Scene Settings and Export.

First, we have to setup the scene, so that it exports normally. Go to OGREMAX -> Scene Settings and a big dialog box will appear. At the top, you can see a bunch of tabs for various settings. We will only be focusing on the Environment and Meshes tabs. Click on the Environment tab, and UNCHECK Export Environment Settings at the top. If this is turned on, your model will include the environment settings of your Max scene and conversely your Asset will render black inside our engine. Leave the other settings as is. After this, go to the Meshes tab and make sure that all the boxes are checked like the example below:



That's it for the Settings. Close this dialog and select your model.

Go to OgreMax -> Export -> Export Selected Objects.

A Windows browser will appear, asking you to save your file. Go to your OGREMAX folder which you made previously and where your 2 textures are located.. Use as your file name the same name as your material which you made previously. In this case "american_house" (without the quotes of course).

This is very important! Your exported files should always have the name of your material, so that they can recognise each other. Under the File name box there is a Save as type: box. Make sure it is set to OgreMax Scene (*.scene)

Click on Save.

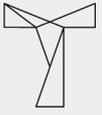
Go to your OGREMAX folder you created, and now you can see that the exporter has created 3 files. A mesh.xml file, a .material file and a .scene file. These all reference each other.

The file, which contains our model, is the mesh.xml file. This file contains all the models dimensions and vertex coordinates. This file will be used to convert our model to a .J3O file.

.J3O is a native mesh format, which is used by our jMonkey engine platform. In order for any model to be rendered properly, it must be converted to .j3O. Make sure that the 2 textures are still in the same folder, because they will be needed as reference when converting the model.

There is one thing you should probably check before moving on. Open the .material file by right-clicking on it and choose open with Wordpad. You will see a block of code. This is the material definition that has been created when exporting the model. Go to the line between texture_unit where your texture is being referenced. Make sure that the .jpg extension is lower case. If not, correct it. This happens when your original texture, which you have created is saved with an upper case.

This completes the OGREMAX part of the process. You should have 5 files in your OGREMAX folder now. The 2 textures and the 3 OGREMAX reference files.



TYGRON

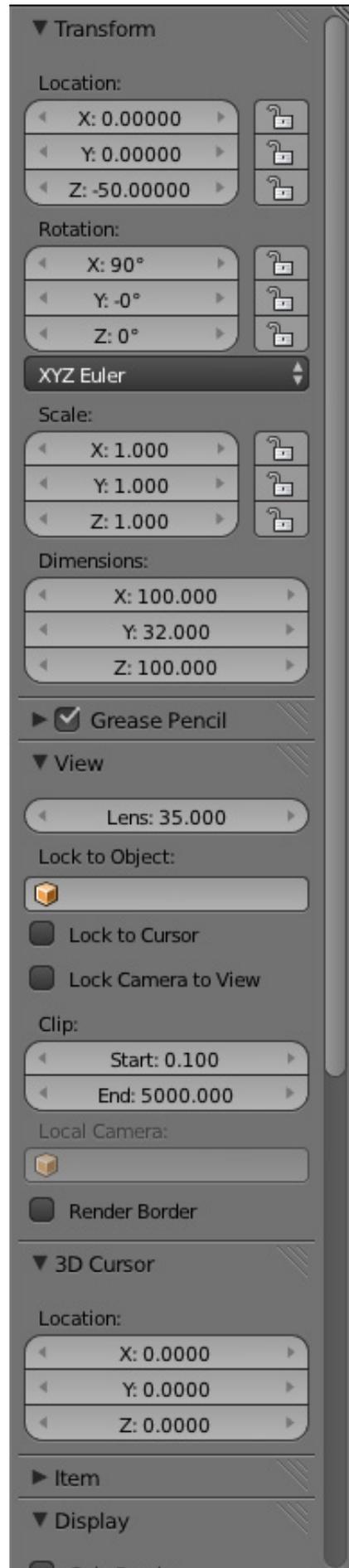
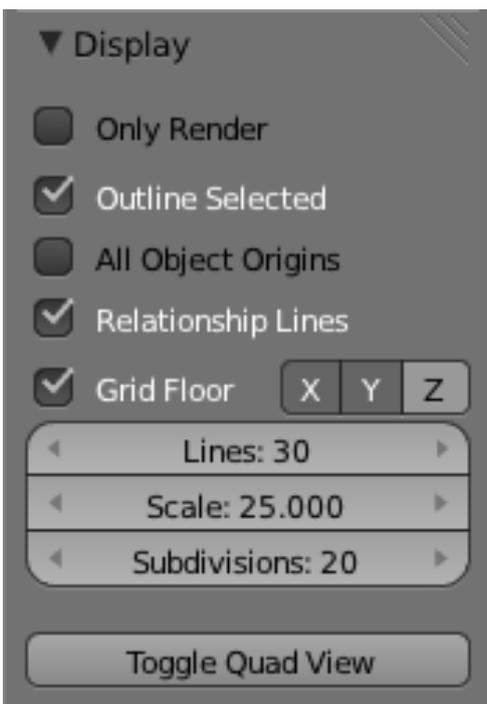
Blender Quick Start

What follows here is a short introduction to Blender showing only the basic essentials and how it should be set up for working with the Tygron engine. Especially when dealing with the grid and exporting this is quite important.

Scene Setup

Before starting one of the most important things to do is setting up the grid properly so that one can create assets for the Tygron engine at the correct scale. The grid settings can be found right-hand side of the viewport.

When scrolling down in this tab, the section for Display can be found. Make sure the grid floor is turned on. The Scale of the grid floor is most important. Scale needs to be set to 25. For Subdivisions it's nice to have a round number, 20 will do. Lines are less important, but it's best to have at least a fairly high number, around 30 should be fine.

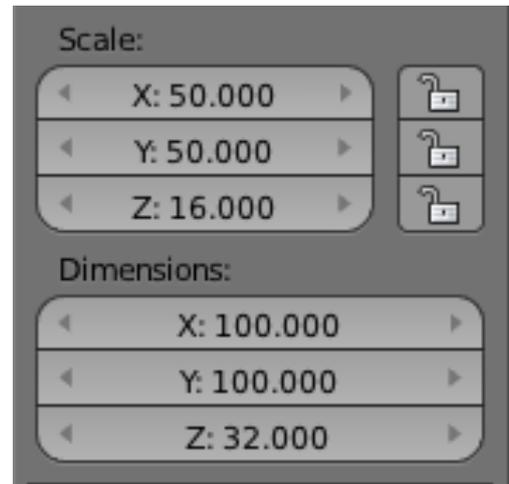


For scene navigation Blender has some nice presets for users that have experience with either 3Dsmax or Maya. Go to *File > User Preferences*. In the Input tab there is a dropdown menu in the top middle. This has Blender (Default), 3Dsmax and Maya options. This will change the navigation keys so that it's much easier to use for people who are used to either 3Dsmax or Maya.

Modeling Basics

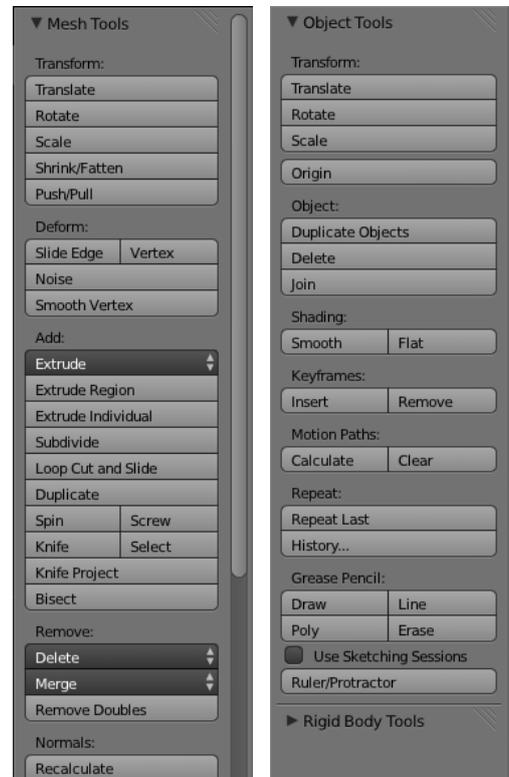
In the menu bar is the Add menu for creation of new objects. Create a cube and set the X and Y dimensions to 100. Which is also equal to 10 metres. When making a stackable building for the Tygron engine, the height that is most commonly used is 3.2 metres. Which is 32 on the Z axis.

With these dimensions you have the basic size of a building block asset.



When an object is selected, in object mode it can only be moved, rotated and scaled. When you want to edit individual faces, vertices or edges you need to enter Edit Mode, which can be activated with the TAB key. Press TAB again to exit Edit Mode.

Most of the tools needed for editing a model can be found on the left-hand side menu and in the bottom. The left menu changes depending on whether you are working in the Object Mode or Edit Mode. By far the most important tool here is Extrude, which can be used in the Edit Mode.



Edit Mode

Object Mode

The most important tools in the bottom bar are the transformation tools for Translating, Rotating and Scaling. These can also be set to Global or Local and other transformation types. It also contains three buttons for Vertex, Edge and Face Select when working in the Edit Mode. There are also settings for snapping here that can be useful for more exact modelling.

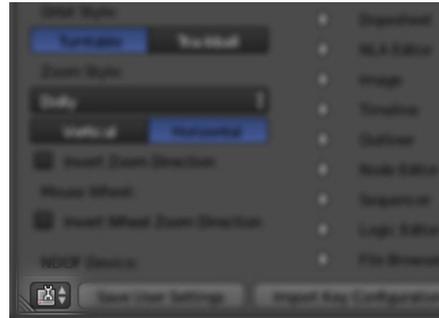


Transformation Tools | **Selection Tools** | **Snapping**

UV and Texturing

All windows are modular and have a button on the bottom left where you can change the window into something else. The UV/Image Editor is also located in this list.

A nice way to work is having the UV Editor in a separate window. A quick way to do this is to open the *File > User Preferences* menu again. This opens a new window. Then you can click the bottom left button and select the UV/Image Editor to bring it up.



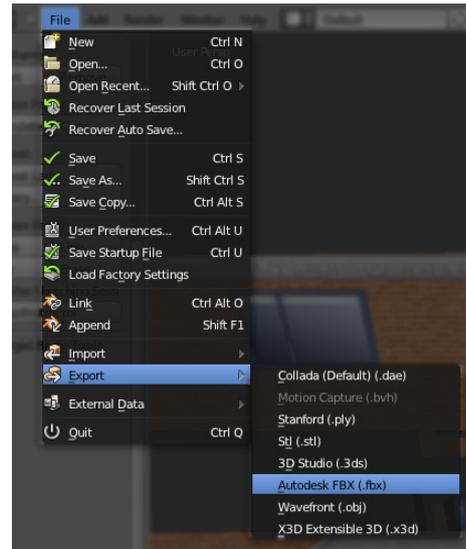
When model faces are selected in the Edit Mode in the viewport, an image can be applied to these selected faces in the UV/Image Editor. Click on *Image > Open Image* in the bottom left to open a new image and apply it to the selected faces.

UVs can only be edited when in the Edit Mode. Under the UVs menu in the bottom left there are a number of useful tools, such as Weld/Align and Mirror. The selection and snapping tools are located in the bottom just like in the viewport.

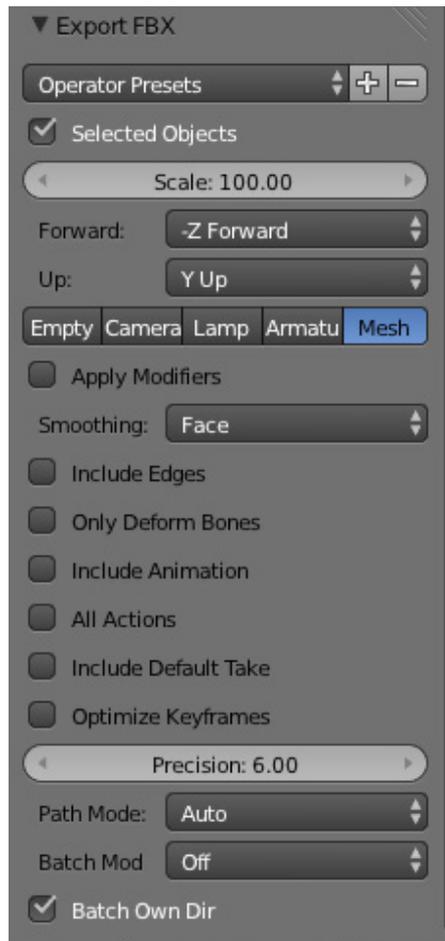


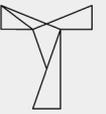
Exporting

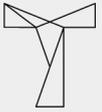
Make sure that the scale of all the models are correct (10m x 10m x 3.2m - Which means 100x100x32 in Blender XYZ). When you are sure your models are the right scale, please make sure that you select the objects you want to export. These objects can make a lot of problems for you later on if you forget to remove them before exporting.



When you are certain everything is correct and done as said, you can now start the exporting process. To export your model, select *File > Export > Autodesk FBX (.fbx)*. The settings for exporting is shown below, make sure you check / uncheck everything like in the example. Save the document where wanted and you are done exporting.







TYGRON