Artist's Guide

Player Character Information

Information regarding the player controlled character to guide for appropriate dimensions.

Player Size

- Width: 70 cm
- Height standing: 190 cm
- Height crouching: 120 cm
- Eye (camera) position: 10 cm below height.

Level Information

Measurements

The following measurements are good if kept and all models created should assume them. However, they are not set it stone and can be broken if really needed. However, do not go crazy and use other measures for the fun of it, break them only if really needed. For example, if there is to be a hole in a wall for the player to crawl through it might look silly with a 0.25 meters thick wall, and you want to ramp up to 0.5 or even 0.75 to make it look good.

General

Wall thickness: 0.25m
Ceiling thickness 0.25m

Doorways

The size of the doorway (ie the hole in the wall that it is placed in) depend on what kind of door it is. For doors that are swingdoors (ie normal doors on hinges) and/or are meant be be used in a realistic real-life environment then a certain set of measurements. This type is called “realistic”. If the door is meant to be used in more sci-fi setting or that the doors themselves are not ordinary, then another set of messurements is used. This type is called “fantastic”.

Realistic: 1.5m x 2.5m
Fantastic: 1.75m x 2.75m

Doors & Frames

The doors should be smaller than the doorframe hole, but be careful not to overdo this. The rest of the hole space is filled up by the frame (see below).
Standard size: 1.3m x 2.45m (this is what looks best for a normal realistic door)
Minimum size: 1.1m x 2.3m (this is so that AI will always fit).

The frames of the doors can be whatever you like, the only thing necessary is that they fill up the hole so if the door were to removed the walls are not visible.

Characters

When creating a characters for depth there are some things that need to be kept in mind.

Basics

This are some general points that apply to any kind of character.

- Areas where the character will most likely bend should have extra polygons for smoother deformation. Example: Knees, elbows, wrists.
- Never do any textur mirroring or similar. Every pixel on the rendered character should be a unique texture pixel.
- Unlimited number of bones are supported. But it would be best to keep below 256

Humanlike

This include any character that has a human like appearance and features. Basically any character that might be in need of facial animations.

General

- Clothing and accessories like watches should be a part of the mesh. We dont want body parts sticking out of clothes.
- Spend polygon budget wisely. Example: If the character has a baseball cap make sure it is smooth as the silhouette will be very noticable. Also have more polygon density on the head then the body.
- All skin surfaces must be seperate texture

Face

- The face shall always be a separate texture/material.
- A good idea is to split the head texture where the neck meets the shirt line, to try and keep it as seamless as possible.
- If there is any amount of facial hair (very large beard or any haircut more than very bold), hair must have a separate texture.
- Eyes need to be separate submeshes and also have their own textures.
- Make sure there is enough mesh for the pupil to get to every corner of the eye.

- Eyelids should be able to open and close.
- Mouth need to be able to open, have teeth, toungue and uvula
- The facial rig should allow exaggerated poses (if we were to have animations to be seen at a distance)
- The inside of nostrils must be modeled

Entities

Entities are all objects that can be (but do not have to be) dynamic and will always have an .ent file. The ent-file always include a model file but can also include lights, physics, sounds, animations and more. These make up the foreground of the game.

What is an entity?

It should not hard to identify what should be an entity, but sometimes the object you have in mind is made up out of one or several entities. The easiest rule of thumb to have is that an entity only serves a single purpose and allows for a single coordinated movement. For example, if you have a door with a scanner, and lamp above, it and a cog wheels that turn when the door move, this is NOT a single entity instead this would be split into four different. These would be:

- Door (and frame)
- Scanner
- Lamp
- Cogwheel

Each of these serves a single purpose, the door can be opened, the scanner can be interacted with (and light up/down), the lamp can be turned on/off and the cogwheel can spin or stand still. All of these require different controls and thus need to be made into different entities. Another example (which is different from how Amnesia worked) would be a desk of drawers, here there would actually be two different entities. One for the frame and one for each drawer. The reason is that each drawer has its own movement to be controlled, and thus require a single entity for each drawer the desk contains. If the desk had only contained a single drawer the drawer and frame would be one single entity!

Do NOT take this this thinking too far though. For example if a machine has lots of lights that light up when it starts working, and no other moving parts, then DO NOT make this into two entities, one lamp and one machine. Because the lamp and machine work in unison (lights light up when the machine start) there is no need to split them up! As always, if unsure, ask someone else (this makes it possible to put the blame on another person ;))

Categories

First of all, entities needs to be placed in the main category that they belong to. This is can be “creatures”, “organic” or a category for a specific type of environment in the game, e.g. “shipwreck”. All main categories then have sub categories that the individual entity folders are placed and for most categories these vary a lot. However for the environmental categories (like “shipwreck”) there is a certain basic system:

What follows is a list of the most common categories of entities:

“lamps”
Everything that emits light and can be turned on and off.

“furniture”
Stuff like sofa, tables, chairs, etc are placed here. Most stuff you see at Ikea is found here.

“ornaments”
This entails any kind of objects explicitly used to spice up the room. Vases, statues, books, etc belong here. This is NOT anything you think will spice up a room, but something the virtual/fictional owners have placed to do so.

“ornaments_wall”
Just like ornaments but everything that belongs on a wall. Posters, paintings, mirror, etc are in here.

“machines”
Anything that does some kind of mechanical process is in here. Diesel engine, pneumatic press, etc are here.

“tech”
All electronic gadgets go here: computers, speakers, radios, etc.

“panels”
Screens hat can be interacted with. E.g. hand scanners, eye scanners, touch screens, etc.

“gameplay”
Simple things that the player can physically interact with, which includes buttons, switches, levers, wheels, etc. This also any special object used for gameplay purposes, for instance woodboards the player can pry of a window or a special bridge that comes out of a wall. If it is used for gameplay, and do is not really fitting in another category it belongs here!

“storage”
Inventories you would find in a warehouse or storage room. Barrels, crates, boxes and shit.

“tools”
Any common (or perhaps uncommon) useful tools go here. Scissor, hammer, screwdriver, fire extinguisher, buzz saw, etc

“utility”
Things that are sort of “attached” to the environment and serve direct specfic physical usage. Examples: sprinkler, electrical outlet, ventilation, etc

“organic”
Flowers, corpses, gibs, etc. Anything like that.

“debris”
Non-useable static junk for the floor. stones, plaster, cloth, paper, etc.

“support”
Things that cannot really stand on their own and have many uses. Planks, girders, steel bars, beams, stretch of rope, etc. Anything that you use together with the environment and very seldom by itself. Important note: Much of this should probably be in static objects, so only use it here if it has lots of

polys or a need to by dynamic.

“edibles”
Anything that can be consumed, water bottle, food, medicine, etc.

“doodad”
complex stuff that does not serve any purpose other than spicing up the environment. Important is that it is complex enough that it should not be in static objects.

These are of course not the only categories available but should give a good start. If adding categories, do NOT add to specific ones like “red herrings” or have two categories that could contain the same entities, like “food” and “groceries”.

Naming

When naming entities the main rule is to make it easy for other people to easily find objects when editing. In order to do so make sure not to be to general and to name things in order of descending importance. Something like this:

[type of object]_[object name]_[attributes]

Examples: “computer_laptop_broken”, “table_dining_room_wood_large”

Make sure to be consistent in this type of naming, so that you do not have several different names for the same type of object. For example, do NOT begin some names with “pc”, some with “computer”, some with “terminal”, etc. Instead begin all entity names with a single name, like “computer_” instead!

You do not need to be all to specific in this kind of naming, so do NOT have names like:
“computer_linux_red_hat_networked_600gb_ram_white”!!!
But do NOT be too unspecfic either and name something “computer_small” either!

Type Specifics

Slide Door

When making a slider door make sure that the opening is always along the Z-axis (that is any character would go through the door along the z-axis).

Swing Door

Make sure that the door opens in the postive Z direction. That is when the door swings open it will travel down a postive z axis.

Shapes, bodies & Collisions

Shapes are used to create bodies which in turn are used to give entities all their physical properties.

Shapes

Shapes can consist of the following shapes:

- Box
- Capsule
- Cylinder
- Sphere

Sphere is the fastest (as in takes the least amount of processing power), followed by capsule, cylinder and finally the box is the slowest.

Bodies

Bodies are created from one or several shapes. Bodies contain the actual properties for weight, collisions and so forth. It is to bodies that you attach models, lights, sound etc which results in the final entity.

For all bodies make sure to, AT THE VERY LEAST, configure:

- Body Material, this sets what sound, particle and physical behavior the body will have. For example, set it to Metal, and it will clank and show a spark on impact, and the entitiy will have less friction than if it were set to for example dirt.
- Mass, set this to something realistic (in kg), setting it to 0 will make any type of entity static.
- Max Linear & Angular Speed, a value of 8 to 16 seems to let entities move fast, but not get out of hand. Tip of the day: This can be used to fake air resistance by using a very low value.

Creating Bodies for Collision

When configuring the bodies to be proper colliders for entities there is basically one goal: **To use as few and simple shapes as possible to create an as accurate as possible body (or bodies) for collsions.**

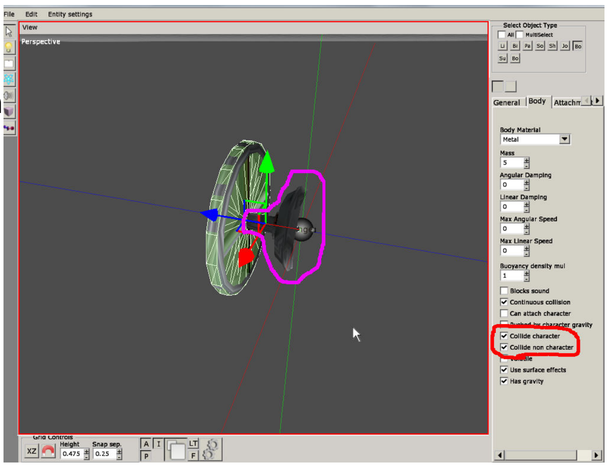
Some general rules:

- Don't create a body that has too great of a difference for the values of x, y, z. For example creating a very thin box to be a collider for a piece of paper is a bad idea if the paper is going to be dynamic, it will behave strangely and can even fall through the floor.
- Use a mix of bodies that have character collide, none character collide and both. For example a table, should have character collide turned off on all bodies that does the actual colliding for the table shape against other entities. Then one single body should enclose the whole table, with character collide turned on but none character collide turned off. This will make the player character behave more appropriate when in contact with the table.
- Test collisions for your entities in-game. This is best done by taking a simple entity, like a small box and then push it against the newly created entity to see how well the collisions work. This is mainly important in the beginning, to see just how simple type of shapes you can get away

with, or to quickly notice that you can't be too sloppy either.

Collider Examples - Wheel

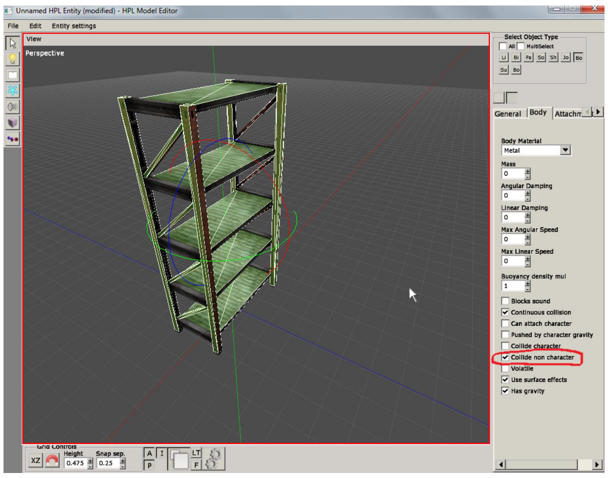
The body of the wheel is only one cylinder, there is no need to make shapes for all the small bars in the wheel. As this is the only body for this entity it collides with both the character and other entities, no need to have different bodies with different collider types. Also see how the purple marked models (the mount and bar attaching to the wheel) have no bodies at all, this because nothing is small enough to actually get in between the wheel and the wall it is mounted to. There is a joint for this entity, and the joint will make sure that the wheel is stuck in mid-air and can be rotated.



Body Material: Metal
Mass: 5
Angular Damping: 0
Linear Damping: 0
Max Angular Speed: 0
Max Linear Speed: 0
Buoyancy density mul: 1
☐ Blocks sound
☑ Continuous collision
☐ Can attach character
☐ Pushed by character gravity
☑ Collide character
☑ Collide non character
☐ Volatile
☑ Use surface effects
☑ Has gravity
Height: 0.475
Snap sep.: 0.25

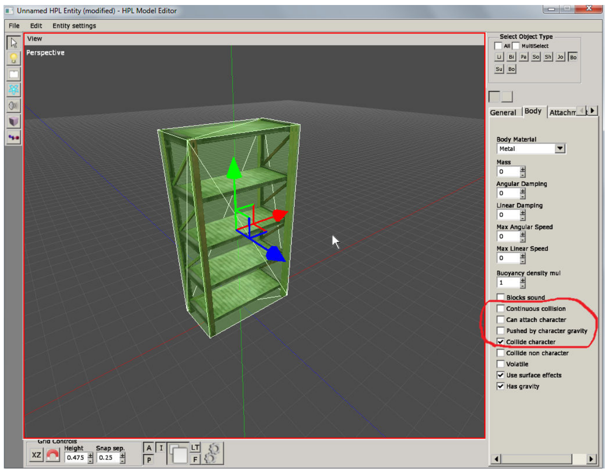
Collider Examples - Shelf

The body for the collision against other objects. The body is quite similar to the actual graphical object, mainly lacks the small differences. Depending on game and what type of other objects that exists, the sides could be changed to one single boxed shape instead of the current six box shapes. But if you have small objects like cans they could potentially collide in mid-air if cheating too much with the collision details.



Body Material: Metal
Mass: 0
Angular Damping: 0
Linear Damping: 0
Max Angular Speed: 0
Max Linear Speed: 0
Buoyancy density mul: 1
☐ Blocks sound
☑ Continuous collision
☐ Can attach character
☐ Pushed by character gravity
☐ Collide character
☑ Collide non character
☐ Volatile
☑ Use surface effects
☑ Has gravity
Height: 0.475
Snap sep.: 0.25

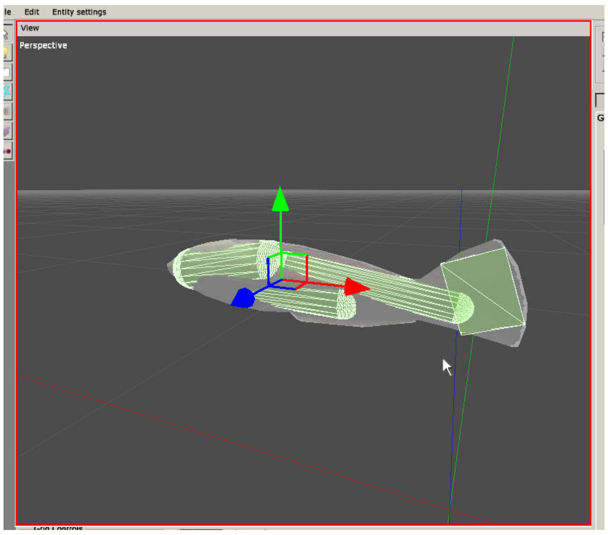
The box for colliding with the character. One single large box covering the whole model, this will make the character collision work faster and there is no risk of small weirds bumps and similar if the player walks against the model.



Body Material: Metal
Mass: 0
Angular Damping: 0
Linear Damping: 0
Max Angular Speed: 0
Max Linear Speed: 0
Buoyancy density mul: 1
☐ Blocks sound
☐ Continuous collision
☐ Can attach character
☐ Pushed by character gravity
☑ Collide character
☐ Collide non character
☐ Volatile
☑ Use surface effects
☑ Has gravity

Collider Examples - Organic

Normally for something as small as a fish you only use one simple shape, as the entity is so small that you can't really tell how well it collides with other entities. But the fish gives a nice organic, complicated shape that is easy to show how to simplify.



Using three capsules and one box you can create a very detailed collision model for the fish, it is likely even only two capsules would work just as well, but it would have to be tested and tweaked. All the parts of the model not covered by a shape is not a problem, because the outtermost points of the shapes will create areas inbetween them, where nothing is small enough to get into. Or it would be such a small part of the entity that it would be difficult to see on screen when moving if it clips into other entities on collision.

Terrain

Settings

All maps shall have the following settings:
unit size: 1 m
geometry patch size: 16
texture patch size: 16
Max height map size to use is 1024×1024. Keep this lower if possible (but still make sure there is ample of room around the area of gameplay focus).

The sizes of the blend textures are a blir more flexible. Having the default of texture size == heightmap size should be good enough, but it is okay to crank the size up to 2048×2048 for a SINGLE layer if more fine details are needed. This is also true for the diffuse color blend texture.

Static Objects

Static objects make up the background of the game and are always immovable. The include things like walls, pillar, pipes, and more.

The most important dimensions to learn in the game is the sizes of the different static objects. The reason that is so important is that it must be possible to use assets from several different sets together. By being able to to do this it is easy to piece together unique looking environments by combining objects from different sets. Another important aspects is that it is much easier to make generic entities and other set pieces that can easily fit together.

So make sure to keep the directions given below as much as possible. There might be occasions when it is necessary to break, but do not do this without think about it and asking somebody else.

The names that are given to the various objects are also important to keep the same, but it is necessary then it is of course possible to deivate from this standard as well.

General

Origo
The Y-axis-origo of most static objects shall be at the lower part of the model. This so that it is easy to place them in the editor. However this is not optimal for all objects, for example ceiling welders.
The X/Z-axis-origo depends a lot on what type of object. For walls, it should be at the front of the object where (from where the floor is visible), for pillars at the center, etc. The important thing is that the origo placement makes the object easy to rotate into place in the level editor.
If unsure, try and figure what is the the default relative postion you want a certain object to be in compared to other objects, and put the origo there. Make sure to also ask another person for advice! This is a vital part as bad origos will make mapping more cumbersome and will probably end up wasting a lot of time.

Naming
The models(.dae) should be organized in a folder structure as follows:
..*category/typeofobject/nameofobject.dae*
..example: “station/wall/default.dae”

If the texture(s) for pieces are shared with other types(i.e. ceiling, floor)they should be place in the graphic set's root folder:
..*category/nameoftexture.dds*
..example: "station/wall/default.dds"

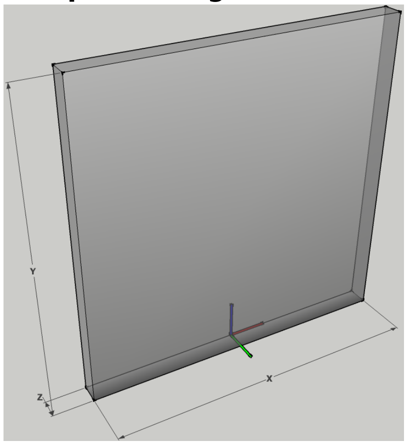
Or if they are specific for the type of object they should be placed in wall folder instead:
..*category/station/nameoftexture.dds*
..example: "station/infirmary_base.dds"

Walls

Default

Dimensions:
X: 4.0m Y: 4.0m Z: 0.0m-0.5m (or more)

Naming examples:
default (can be used on the most generic type of wall, avoid using though!)
default_window (default with a window)
rock (material differnt)
decayed01, decayed02 (two versions of a decayed wall)

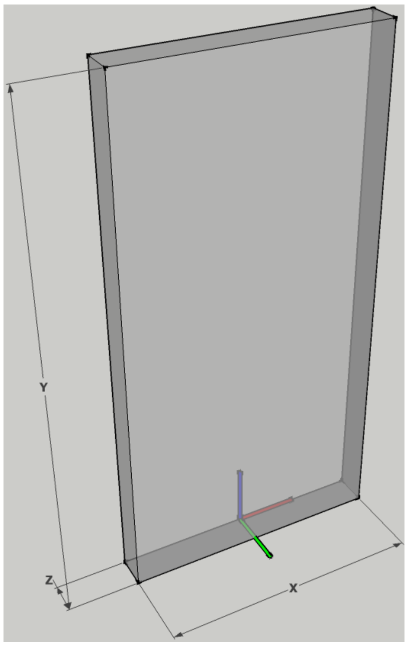
Template image:


Short

Dimensions:
X: 2.0m Y: 4.0m Z: 0.0m-0.5m (or more)

Naming examples:
default_short (can be used on the most generic type of wall, avoid using though!)
default_window_short (it has a window)
rock_short (material different)
decayed_short01, decayed_short02 (two versions of a decayed wall)

Template image:

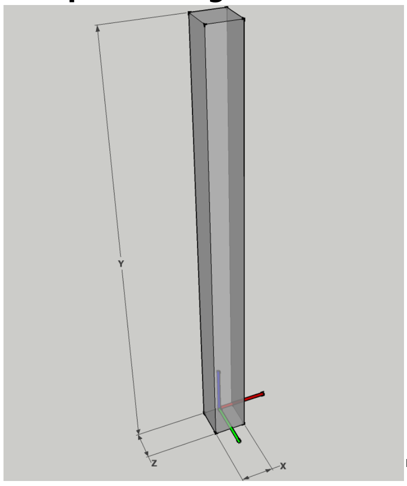


Welders

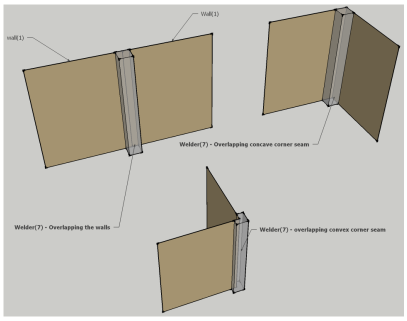
Since it is very hard (at times impossible) to two pieces of wall to seamlessly tile, it is almost always necessary to have an object between two wall pieces, in order to hide any seams. This object is called the welder. The welders can have a very varied ranged of looks, so the measurements here are very flexible. The only really important thing is that the welder is just as high as the wall.

Dimensions:
X: 0.25 -0.5m Y: 4.0m Z: 0.25m-0.5m

Naming examples:
welder_default (the most common welder)
welder_stone (used to weld stone walls)
welder_default_concave (to be positioned over corner seam, see right-most example image.)
welder_default_convex (to be positioned over corner seam, see bottom example image.)

Template image:


Examples of welder usage


wall(1)
Wall(1)
Welder(7) - Overlapping concave corner seam
Welder(7) - Overlapping the walls
Welder(7) - overlapping convex corner seam

Doorway

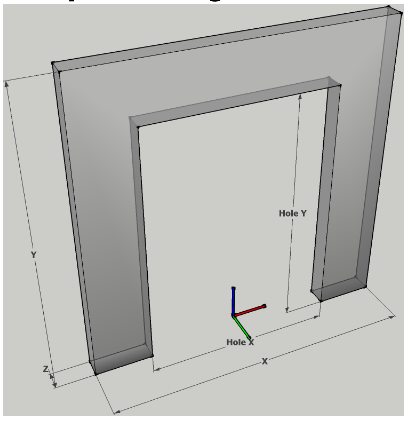
Door ways are holes in walls that are either meant to place a door entity (contain both frame and door) or a static doorframe (see below) in.
For more discussion this see messurements above

Dimensions:
Outside: X: 4.0m Y: 4.0m Z: 0.05m-0.5m
Hole *(realistic type)*: X: 1.5m Y: 2.5m
Hole *(fanastic type)*: X: 1.75m Y: 2.75m

Dimensions (double):
(simply double with of above)

Naming examples:
doorway_default (doorway for default wall type) doorway_default_double (double doorway for default wall type)

Template image:



Doorframe

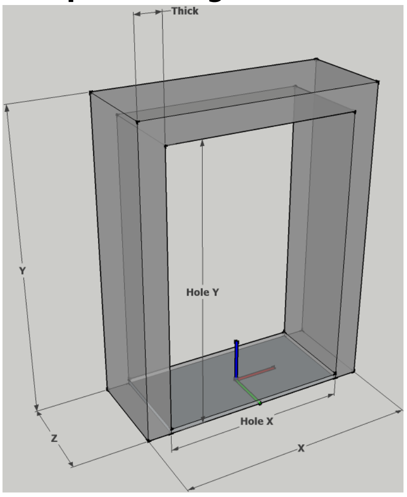
A piece that is a doorframe, the piece will be positioned in the hole in Doorways, so that it creates a nice frame for a hole when there is no door to fill it out. The dimensions are not exact as it depends on the look of the frame.

When a doorframe is placed a door entity is NOT meant to be placed inside it!

Naming examples:
doorframe_default (doorframe for default wall type)

doorframe_stone_double (double doorway for stone wall)

Template image:



Notes

The above are just the most general types of walls and should serve as general guide for the kind of sizes that are needed. There are tons of other wall types that are possible. Here are some examples that have been used in the past:

corner_concave
A special piece for round corners.

corner_convex
Another special piece for round corners, but this timed beveled in the opposite direction.

doorway_arched
The same as the doorway above, but with an arched top. The arched bit should start at (or close to) 3 meters.

doorframe_arched
The arched version of the door way.

wall_extension
Placed above walls in order to make the room higher. Normally these are 2 meters in y-size.

The above are just examples so do not let this stop you from doing cylindric corridors, non-euclidean geometry or whatever madness your deranged mind can make up! **Wall extension**

Stairs

Default

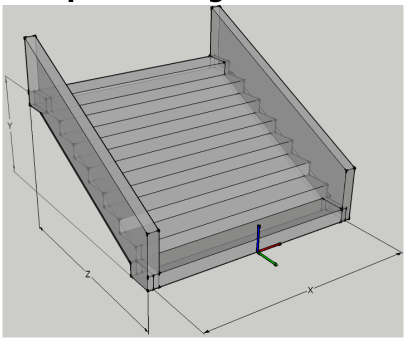
A piece of stair that can be “tiled” endlessly to make as long of a stair as needed. If the model can’t be created to be “tile’able”, make one additional stair model (see stairs Long). Also see example below of how the stair models should be “tiled” and also how they use the welder.

Dimensions:
Total: X: 4.0m Y: 2.0m Z: 4.0m
Step height: ~0.25m
Railing Height: About the same as “welder” (see below)
Notes:

- The actual model with steps, a bottom and railings on the side can be larger than the step measures. The bottom and rail could expand beyond the measures, as well as the railing go into the model. See measures in sketch.
- It’s important that the steps are 4m wide(X) and that the Y distance is from the bottom of the first step to the top of the last step.

Naming Examples: default (the default stairs for the set) metal (metallic stairs)

Template Image:



Welder

These welders are to be used to give stairs a nice look at the beginning and end. They are also be used for making railings look good, see examples. If not possible to use the same welder for the railing as for the stair, make an additional welder for railings.

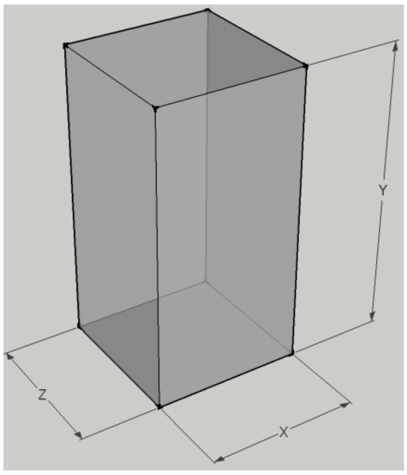
Measurements are very much estimates, so feel free to change as see fit.

Note: The model must be solid, as in all planes are modeled and textured.

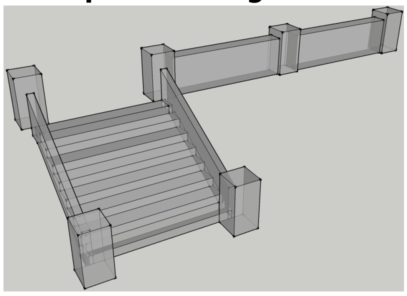
Dimensions:
X: 0.75m-1.25m Y: 1.5m-2.0m Z: 0.75m-1.25m

Naming examples:
welder_default (for the default staircase)

Template image:



Examples of usage



Railing

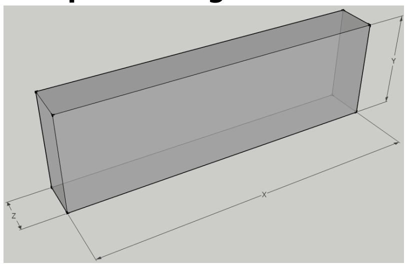
A railing that can be used to create catwalk like Stairs, imagine a stair with a stair that leads up to another stair in the same room and that stair has the railings along the steep to the lower stair.

This is not the railing that is part of the stairs!

Dimensions:
X: 4.0m Y: 1.0m-1.25m Z: 0.25m-0.75m
Note:
The width(X) must always be 4, but the rest of the measures can be quite freely as long as it works well with the stair and static object set in general.

Naming examples:
railing_default (railing used for the default staircase)

Template image:



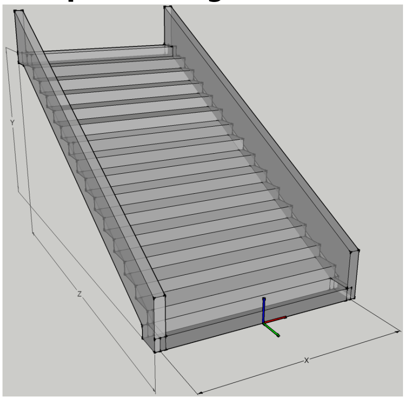
Long

Same as Stairs default, but does not need to be tileable and is only to be made if the default version is not tileable.

Dimensions: Step: X: 4.0m Y: 4.0m Z: 8.0m
Step height: ~0.25m
Railing Height: About the same as 3- Stairs Railings
Notes:

- The actual model with steps, a bottom and railings on the side can be larger than the step measures. The bottom and rail could expand beyond the measures, as well as the railing go into the model. See measures in sketch.
- It's important that the steps are 4m wide(X) and that the Y distance is from the bottom of the first step to the top of the last step.

Naming examples:
default_long (long version of the default staircase)

Template image:


Special

The special folder is only used for objects that are only meant to be used at very specifc places. Name these according tp the level where they are used. For example: "01_machine_room_giant_machine".

Misc

There are lots of other categories that can be added as well. As always let your imagination be your limit, but do not try and be too specific, because that might make it too hard to find certain pieces. Examples of other previously used categories include:

floor
Normally floor is made up textured planes but it might be necessary for more specific pieces at times. Place these here.

ceiling
Like floor, ceiling is normally planes. Place specifics needed here. Often used for pieces that makes the ceiling arched and so on.

pipes
You guessed it. Pipes and shit.

And so on…