

# Level Editor - Props

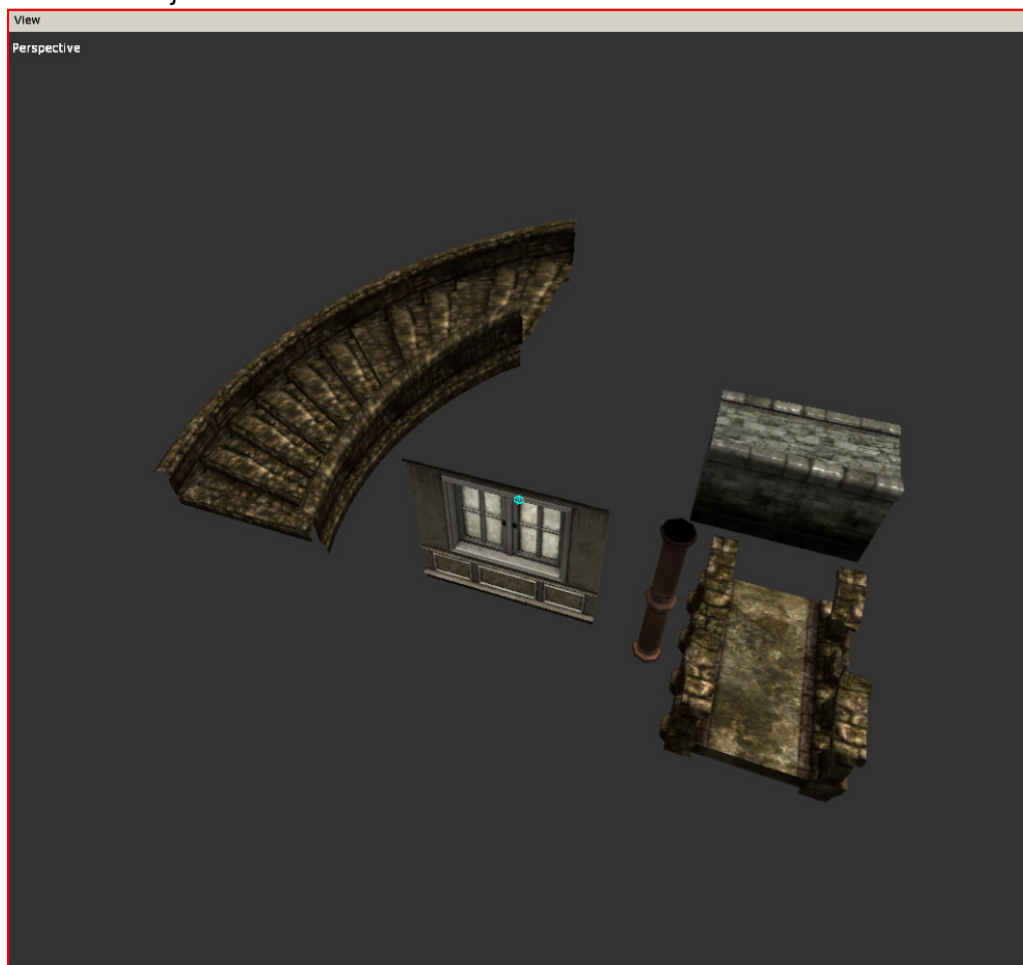
## Static Objects and Entities.

The HPL2 Level Editor is a prop heavy level designer. About 90% of your map will consist of prop **Walls, Ceilings, and Doorways. Props also make up details such as windows, furniture, debris, etc... I will talk about that later on.**

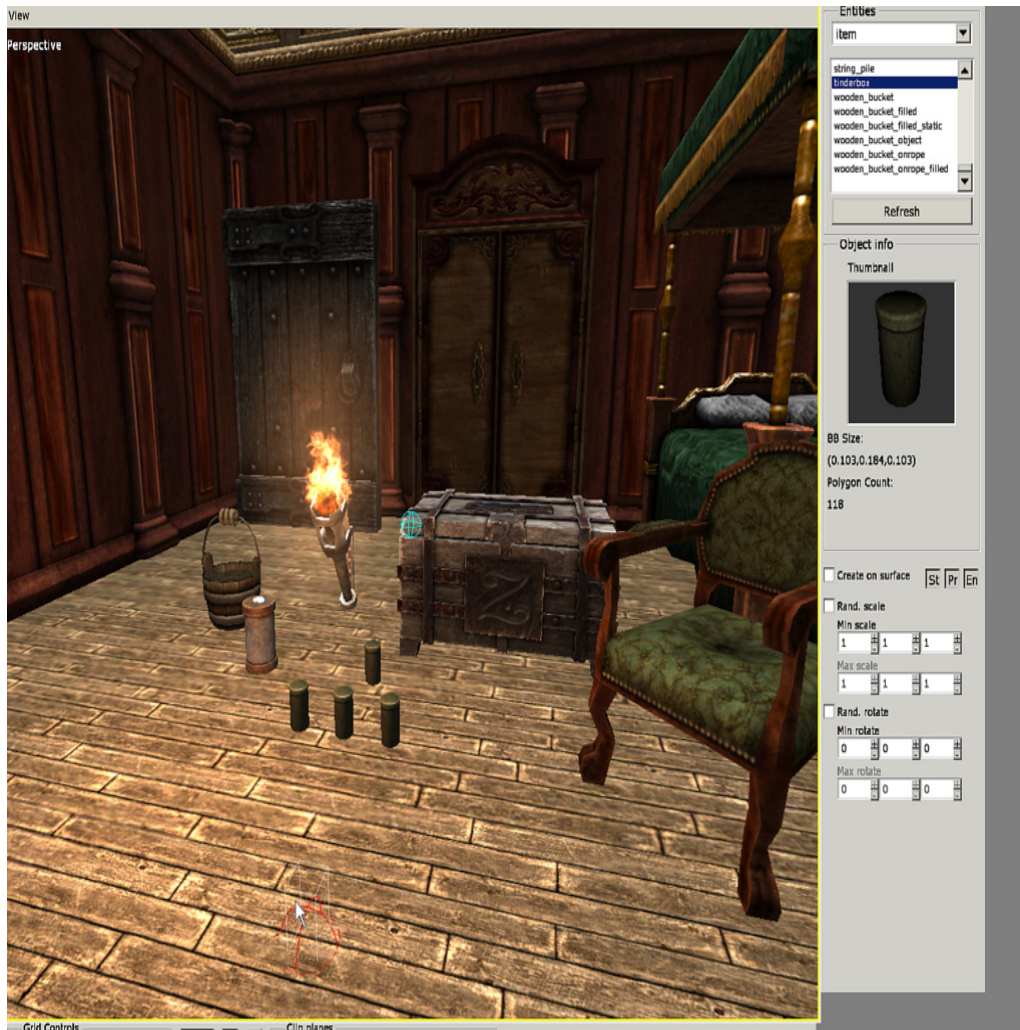
Props are created by selecting the *Static Objects* tool(6) and some with the *Entity Tool* (7). Both tools do not use the same props, Static Objects is used for props that are used to **create the map, such as Walls, Ceilings, Doorways(not doors), Debris with physics(Rocks, Carpets).**

The Entity Tool is used for props you can **interact and Script** with. **Props such as Doors, Closets and chests that open, tinderboxes/oil, NPC, interactive items, notes and diary entries.** Think **dynamic** with entities, they are the props that are used in scripting and events. Entities would be something like a key that opens a door, or a door that loads a new map. Entities are Props that can or do serve a purpose, **think physics, furniture, inventory items, quest items, levers, doors, torches, etc...**

- \*Example of Static Objects



Example of Entities\*\*



# Prop Dimensions(height, width, positions)

## The importance.

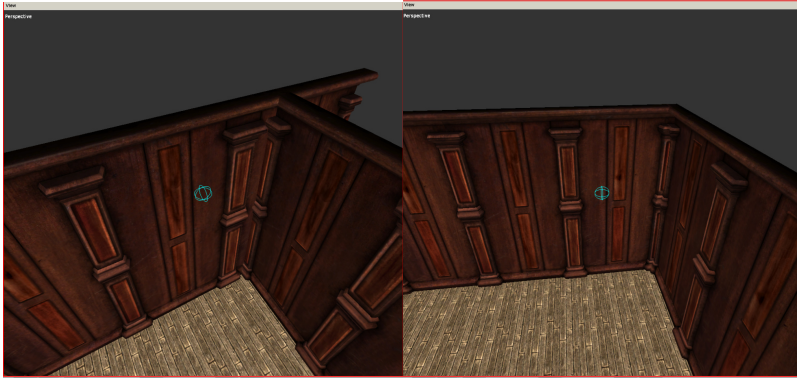
Dimensions for props are incredibly important because knowing them makes mapping incredibly faster, more enjoyable, and less tedious.

**Now this guide will be focused mainly on Static Objects because they need the correct sizes to look and fit right. They are like legos, all have set sizes to work with one another.**

You might be asking "Why not entities?" the reason being for not going over entities is because they don't make up the surroundings of a level, such as walls, they are interactive or have scripting tied into them, you place them where you want and are mainly used for detail so correct placement doesn't fall under entities nearly as much as static objects.

Now Knowing the correct sizes and sizes that work all around well together with props will allow you to **quickly map an area without trying to get size right or messing around with sizes to get things to fit right.**

For Example, notice how this doesn't fit right. **(Left)** And how it does fit **(Right)** **CLICK TO ENLARGE!!!!**



## What are dimensions?

IMPORTANT: 99% OF WHAT THIS GOES OVER IS SIZE AND PLACEMENT, BOTH ARE IMPORTANT BUT CAN BE CONFUSING SO ANYTIME I TALK ABOUT SIZE OR PLACEMENT I WILL SAY SO BECAUSE THEY ARE VERY DIFFERENT. ROTATION WILL BE EXPLAINED BRIEFLY.

Dimensions fall under x/y/z and make up the placement, rotation, and size for any primitive, prop, and entity. They are what you base the Scale, depth(**depth is not used in the editor**), area, rotation, and position when using something as a 3D level editor using a grid axis(**X/Y/Z are the axis**)

Now dimensions can be confusing in level editors because they change what they do whenever the rotation is changed, im not gonna bullshit anyone here, i dont know why that happens, im guessing, that when you rotate a flat plane to a curve instead of being perfectly straight wall or floor, is that then it starts using all 3 axis(X/Y/Z) instead of 2. **I will NOT BE GOING OVER THIS**

Regardless, understanding the x/y/z when it comes to prop placement will make your mapping experience a WHOLE lot better.

## Dimensions with Scale and Placement - The most important part of this guide.

The easiest way I think of them when using **SCALE** is...

1. When dealing with **walls** of any kind, X is the **Width** and Y is the **Height**, **Z is not used**.
2. When dealing with **floors and/or ceilings** X is the **Width OR Height** and **Z is the Width OR Height**, Y is not used.

Now after reading about floor and ceilings(#2), you might be wondering how X and Z could be either Width Or Height. Well its simple and not a big deal, it all depends on what way you viewing through the perspective camera.

A good example for this is to create a floor with an X of 5 and a Z of 3 then look at a floor by placing your camera above it facing down. Then change the X size. Then Rotate your camera so you are looking down still but facing another side then change the X again and it should look like it changed its Height or Width depending on how you view it.

Example of when X is the Height and Z is the Width:

<http://www.flickr.com/photos/65508834@N03/5965071992/sizes/l/in/photostream/>

Example of X being the Width and Z being the Height:

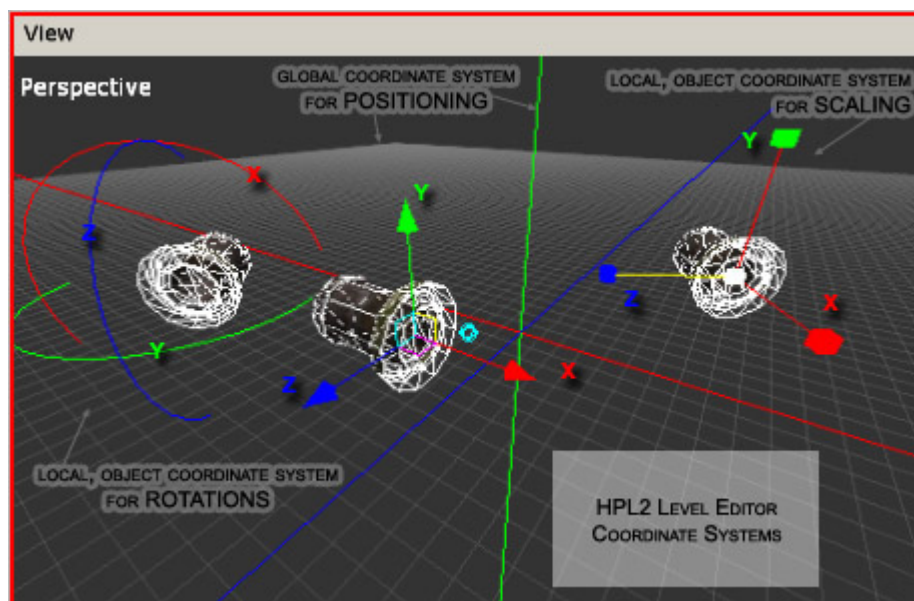
<http://www.flickr.com/photos/65508834@N03/5965072076/sizes/l/in/photostream/>

To be more precise, what is considered width or height is relative. And what would you do if the width and the height are the same? Instead, think in terms of X-size and Z-size.

In HPL2 engine, and thus in the level editor, the X-axis & the Z-axis lie in the ground plane, while the positive Y-axis points up. The editor displays each of these axes in a different color, allowing you to orient yourself. **X is RED**, **Y is GREEN**, and **Z is BLUE**. This is easy to remember - those are the main components of the RGB color model.

In short:

X	Y	Z
R	G	B



This color scheme is followed both for the global (world) coordinate system, as well as for the local (object-space) coordinate system.

The translation, rotation and scale indicators/manipulators seen in the image are called gizmos. To make object positioning easier, the level editor does not rotate the translation gizmo along with your object, so it is always aligned to the world coordinate axes. The rotation and scale gizmos are, however, rotated with the object, so that the object can be properly manipulated. The scale gizmo is similar to the translation gizmo. The rotation gizmo is a bit different, but the color scheme makes it simple to understand and use. Each of the circles rotates like a wheel, and each rotates along a single

*local* axis. However, note that the corresponding values that appear on the General panel are *global*, and that, after a few complicated rotations, will not reflect the behavior of the gizmo. This is done on purpose, so that the engine can track the rotations properly, while allowing for a more intuitive control over rotations for humans.

## Mathematics of Scale and Placement.(Primitives and Static Objects only)

For this sub-section, **SCALE** is bolded when specifically talking about Size and **PLACEMENT** when about placement, they are different with numbers.

Now onto the mathematics part of Prop Scale and placement and how to always know that walls/floors/ceilings will fit or make fit.

Every Static Object, walls, floors, ceilings, bridges, anything big. Everything uses multiples for there **SCALE** and **PLACEMENT**. **Multiples Are numbers that can be divided by another number without a remainder.**

**An Example of Multiples:**  $2+2+2+2+2+2=12$ , **2 goes into 12 6 times which is also  $2 \times 6$** , 12 divided by 2 is 6. (2, 3, 4, 6 are all multiples of 12 because 2,3,4,6 can all be multiplied into 12,  $2 \times 6=12$ ,  $4 \times 3=12$  and can be divided from 12.)

Now Amnesia also uses these and reading this guide will hopefully give you an easy understanding of how they are used when creating the right **SCALE** and **PLACEMENT** with planes and props. Learning this will let you almost never have to bother with resizing unless it is on purpose for a custom fit.

When dealing with anything that lets you adjust the scale/position it will go up or down by 0.25 when you pressed the - or +. From what ive noticed, going up or down by 0.25 does not let things fit 90%. It either makes it a bit to small or a bit to big. Dont fear, you are not stuck to using multiples of 0.25 and there are better multiples to go by that work perfectly with props and primitives.

## Multiples when changing Scale, how they work, and Why correct Scale is perfect.

**As I said earlier this guide only goes over Static Objects that include walls/doorways/ceilings/floors and Primitive planes.**

Also, I do not know every single prop the multiples for them. I do know however, from the amount i tested and seeing how similar Props Scale with one another, i feel safe to say that everything said in this sub-section can be used and will work.

So far using multiples of 2 and 4 work perfectly for Walls. You should never go wrong with walls using multiples of 2 and 4.

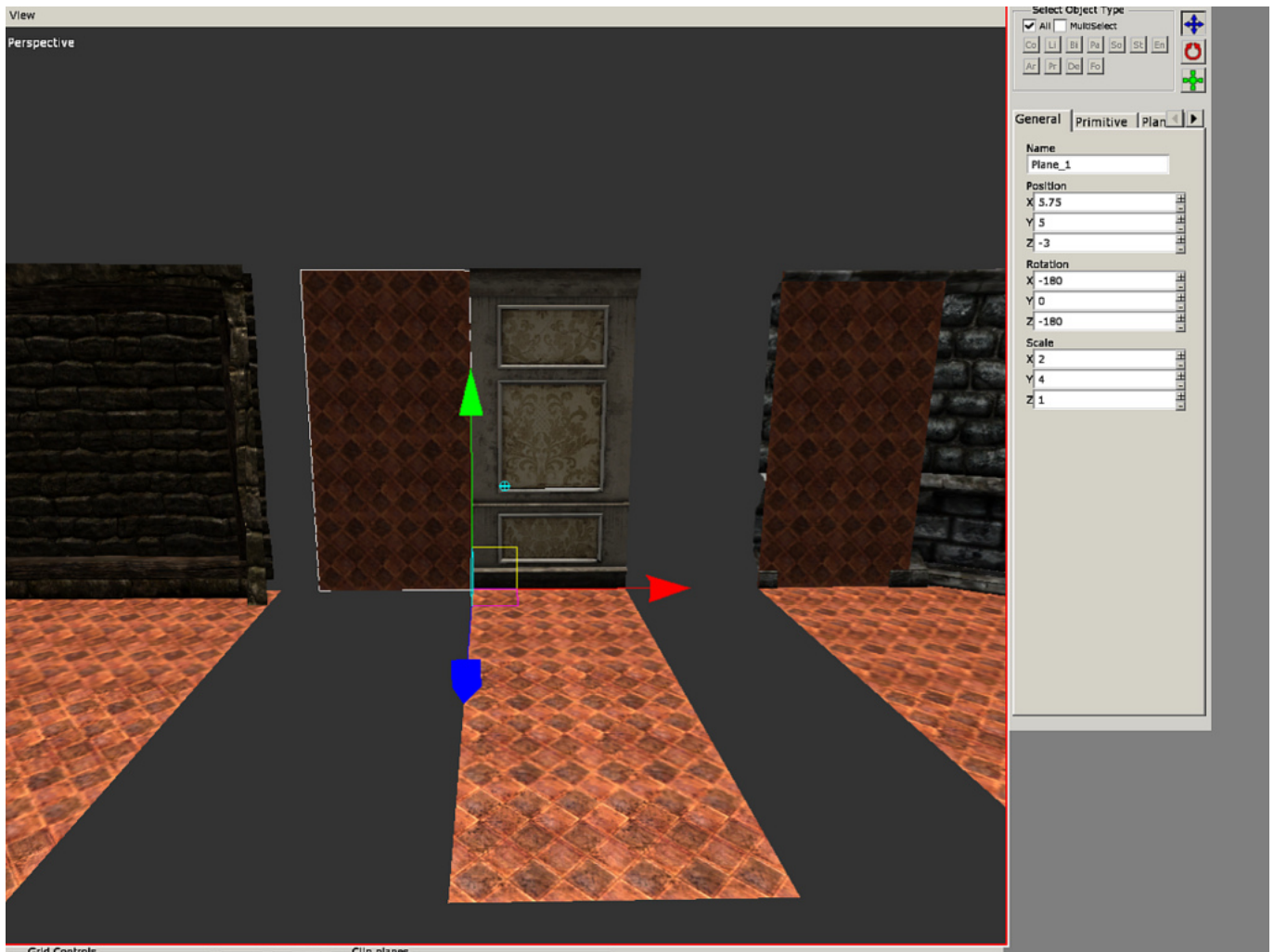
**Examples(look at the scale, not position)**

**Notice the image below, a Plane wall is matched up against half a prop wall. The scale is  $x=2$  and  $y=4$  the  $x$  is the width while the height is the  $y$ . This is good to know because**

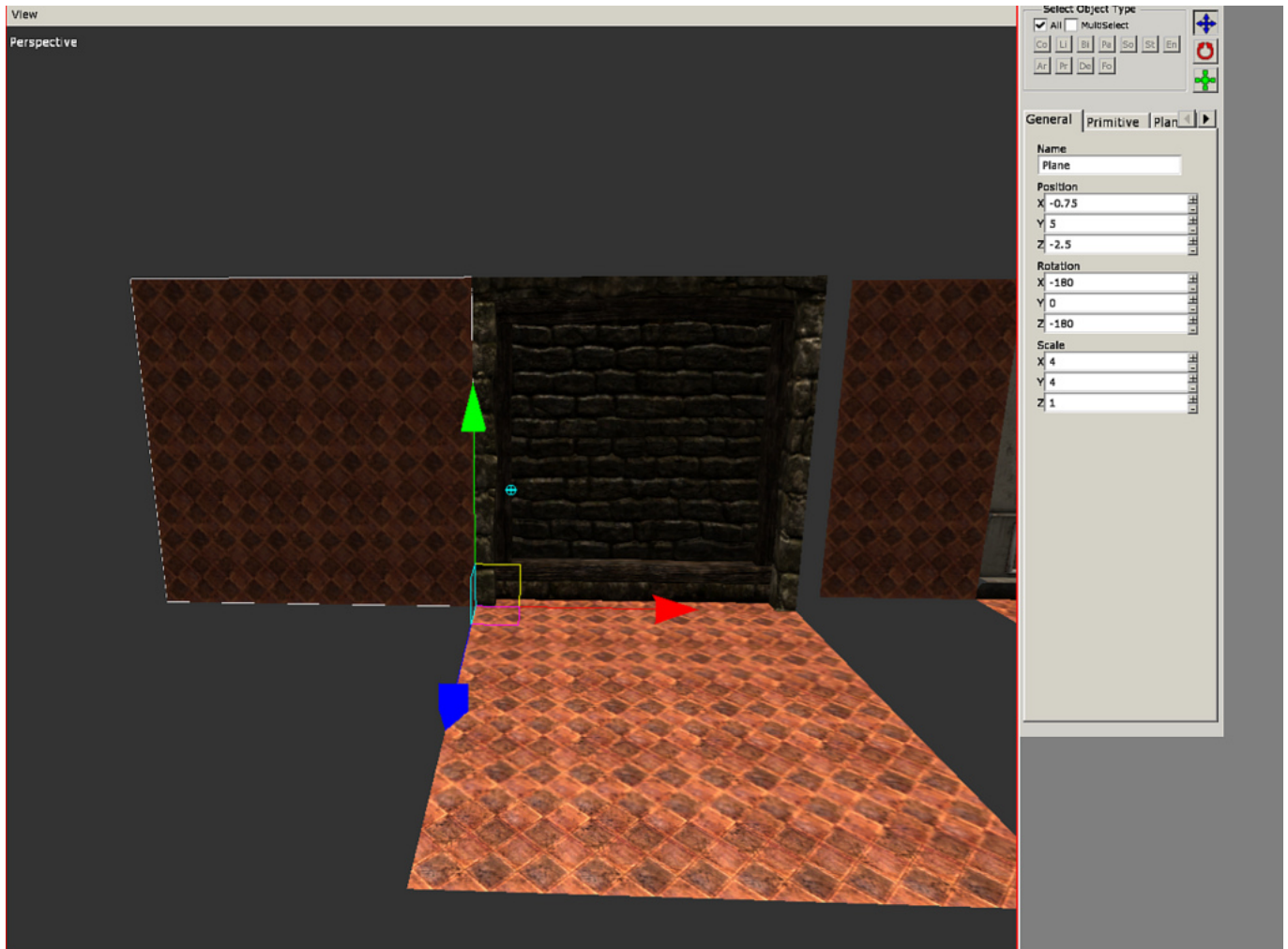


**when using short walls(usually named default\_short) you know that the floor will have to be a multiple of 2 and the height should always be 4 unless you change it.**

Now this is where knowing dimensions can be helpful. Say you use 4×4 walls(thats next up) and want to end on a short wall(what is shown below) you cannot fit a small wall in a room that X=20 and Z=20 when 4×4 walls are being used and only one 2×4 wall. You might be asking why? They are both multiples of 2 and 4 and thats the issue, a 2×4 wall is half a 4×4 wall in width so you would need to change the X to become only a multiple of 2 and not a multiple of both 2 and 4. **Think of it as you added half a 4×4 wall so you have to make up for the missing half.**

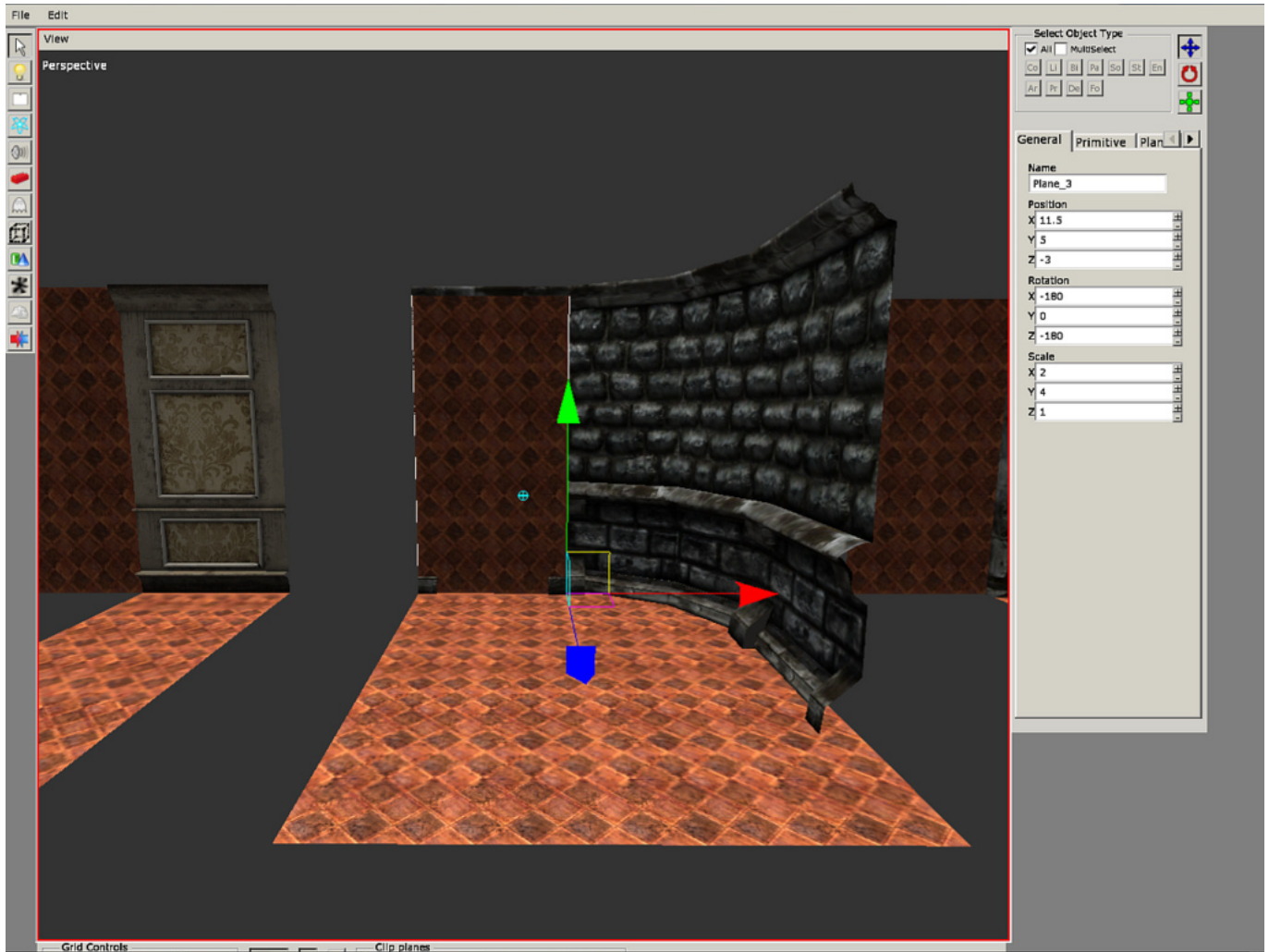


Heres an example of a 4×4 wall, the most basic and probably most used wall size in game. Rooms need to be sized as a multiple of 4, such as X=16 and Z=8 **You wouldnt use any other multiple EXCEPT 4 when getting 4×4 walls to fit, that is if you dont rescale them(I dont advise this at all for walls)**



Now this is when walls can get confusing when trying to get a perfect fit. Curved walls are actually made up of four 2x4 walls. You can see in the image below that one of the four sections to fill is 2x4. Now what does this mean? It means the wall, if made flat, is actually 8x4(because four 2x4 = 4 x 2 = 8x4, Do not worry about the height which is 4)

Now that means your floor has to have a Scale that makes  $X+Z=7.75$  for a perfect fit(it would equal 8 but since it curves it doesnt use part of a corner, saving 0.25 of space) That is because its essentially two 4x4 walls that are just curved together so it needs the same space to fit as if it was a 8x4 wall.**(Note, the floor is not matched for a perfect fit with wall because i do not have  $X+Z=7.75$ )**



Now those three types of walls are in every set and follow the same type of scales and dimensions, so when you use walls like these, remember multiples of 2 and 4.

That being said, not everything under walls uses multiples of 2s and 4s, but will use multiples of 2,4,5,10. Walls like these tend to be bigger or unique,

## Props sets and bases. THIS IS A WORK IN PROGRESS BY REL BELOW THIS LINE.

From what I have learned is that every set(masionbase/dungeonbase/etc...) work perfectly inside there own sets. Think of a set of pieces that are made to work with each other, this is on purpose because of how the devs built the game, they created a buch of props and put them into catagorys that **will always work when used in there respective sets.**

Props in the same set/base(mansionbase/cisternbase/dungeonbase/etc...) **WILL ALWAYS WORK AND LOOK GOOD**

someone has to finish this tutorial



From:

<https://wiki.frictionalgames.com/> - **Frictional Game Wiki**

Permanent link:

[https://wiki.frictionalgames.com/hpl2/tutorials/level\\_editor/tutorial\\_5](https://wiki.frictionalgames.com/hpl2/tutorials/level_editor/tutorial_5)

Last update: **2012/01/05 04:25**

