

Brush Clipping in SinEd - Part I

by Eutectic

1. Introduction

The purpose of this tutorial is to cover all the aspects of brush clipping: what it is, what it does and how to use it. In Part I, we will look at the basic principle behind brush clipping and learn how to use it at its simplest expression: **2-point clipping**. In Part II, we will use a practical example of brush clipping to create a complex polygon from a simple cube. Part III is dedicated exclusively to the most powerful form of this amazing, yet not well known feature: **3-point clipping**. It includes everything you need to know to understand it well and to use it efficiently. We will also complete the polygon started in Part II.

For this, you will, of course, need to have [SinEd](#) installed and running.

2. Background

In the early days of 3D map editing, map editors offered only one way to shape brushes: **brush subtraction**. It wasn't too long before map designers realized that not only was this technique slow, awkward and severely limited, it also caused many problems when used with complex shapes: typically, an excessive number of brushes was created by the operation and often, many of those brush's vertices wouldn't align on the smallest allowable grid location (1 map unit). This, in turn, would often result in tiny gaps between the brushes of complex structures like arches and in some cases, could even cause leaks. In a nutshell, it didn't do a very clean job and caused many design and compilation headaches.

Later on, with the evolution of map editors, 3 essential (and direly needed) brush shaping features were introduced: vertex manipulation, edge manipulation and **brush clipping**. Some editors even introduced extra's like face manipulation and face splitting. Experienced map designers immediately saw the huge advantages offered by these features and adopted them enthusiastically. Many map designers consider this has made brush subtraction completely obsolete. Well, almost. Brush subtraction can still be useful if used judiciously.

All good map editors have implemented **2-point clipping** and some among those (like the Qe4 family editors and BSP) also offer **3-point clipping**. The first, 2-point clipping, is straightforward and very easy to understand. The second, 3-point clipping, is not much more complicated but, let's be honest here, poorly documented and rarely explained in a satisfactory way. This is a rather sad situation considering the huge potential of this great feature and my goal is to de-mystify it once and for all in Part III of this tutorial.

3. The basic concept behind brush clipping

What is brush clipping?

It's simply a **cutting tool** used to **re-shape** a brush by adding extra faces to it. You can also use it to **split** a brush in 2 separate brushes. Is that's all there is to it? In practical terms, yes. In more accurate terms, your cutting tool is actually a **clipping plane**.

So what's a clipping plane then?

Well, it's plane you use to cut things! ...

Ok, let's first look at what a **plane** is. Since I don't want to get into elaborate mathematical concepts here, I will use a practical image to illustrate this. Just consider a plane to be a **perfectly flat, infinitely thin sheet of paper** that extends infinitely in all directions. IOW, your sheet of paper is also infinitely long and wide. Now, imagine you can grab that sheet of paper, move it around and rotate it in 3D space to set it in any position and orientation you want.

Let's take this idea a step further and imagine that this sheet is not made of paper but actually from indestructible steel that can't be twisted or deformed and can cut through anything. Sort of like a razor blade from Krypton of galactic proportions that... uh, never mind

How is it represented in SinEd?

This is where the necessity to visualize things comes in. Throughout this tutorial, I have represented the clipping plane with a thin, black textured, rectangular brush in my illustrations to help you visualize the concept of the clipping plane. This will prove most useful when we get to 3-point clipping in Part II. Just keep in mind that in **SinEd's 3D view, you won't see that black "plane"**. You will **see only the orange dots and the dashed line** drawn in between. Please take a look at **Figures 1a & 1b** below to get acquainted with my graphical representations of a clipping plane.

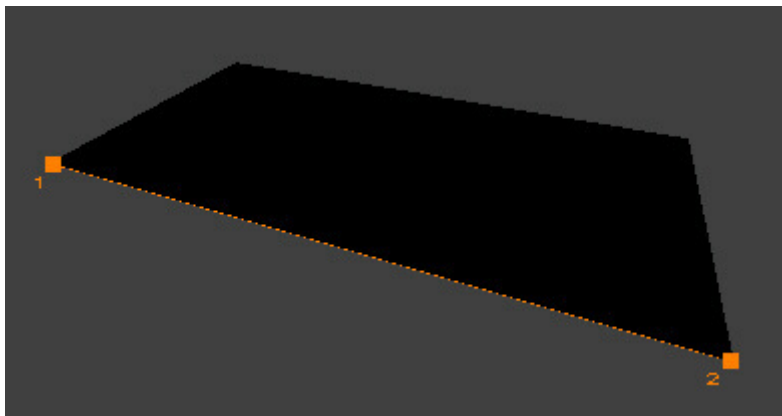


Figure 1a: Graphical representation of a horizontal clipping plane.

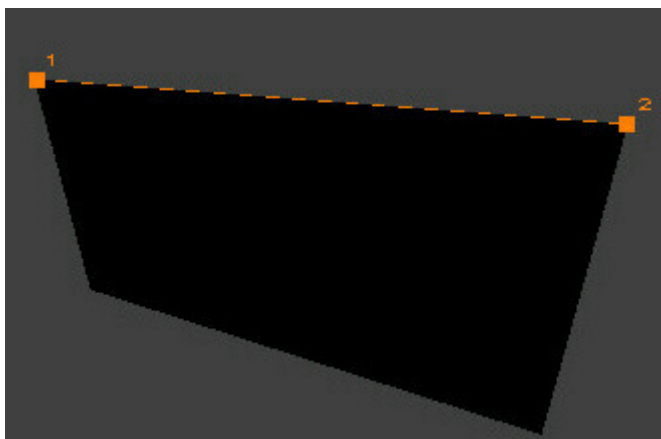


Figure 2a: Graphical representation of a vertical clipping plane.

What happens in reality is that SinEd uses the line traced between the orange numbered dots to define the location and angle of the clipping plane. But the plane itself is not displayed in the views, only its resulting effect: the shape and size to which the selected brush(es) will be clipped or split.

4. How to use 2-point brush clipping in SinEd

Preliminaries

Before we get started, we need to make a little modification to our SinEd project file: **sample.sed**. A little mistake in it slipped right by me and it's important to correct it to benefit from a very useful brush clipping feature called **flip-clip** which we'll get to a little later on. So open sample.sed (\tools\sin\base\scripts folder) with a text editor and scroll down to this line:

```
"KEY_CLIPPER_FLIP" "~"
```

and change the ~ character between the quotes to **kpadslash**. This is what it should look like:

```
"KEY_CLIPPER_FLIP" "kpadslash"
```

Now save your file. This part of your project file sets what keys are assigned to which commands in SinEd and for some reason, the ~ key is not recognized thus denying you of the flip-clip feature. What we have just done is to reassigning it to the **keypad slash key**. I chose this key because it wasn't assigned to anything else and it's easily accessible.

Let's start chopping dem brushes!

Ok, open SinEd, bring the 2D view to the foreground, set your grid to 16 units (5 key) and draw a 128 x 128 x 128 brush. In my examples, I used a clip texture for looks but texture doesn't matter at all here. Once you're done, make sure you go back to the **XY 2D view (top view)** using your NUM LOCK key. This is what you should have:

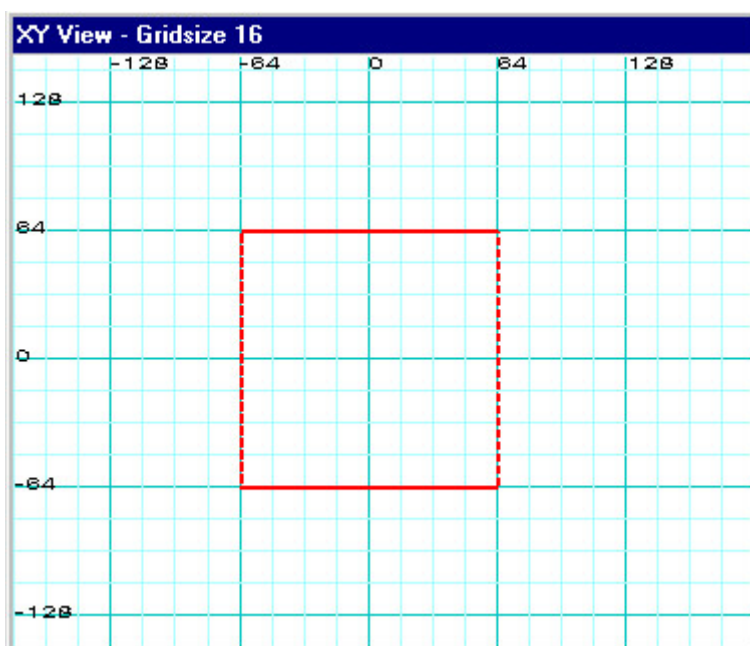


Figure 3a: Your basic cubic brush - 2D XY (top) view.

Next, go to the 3D view and move your camera around using your arrow, <, >, D, C, A and Z keys or if you're the stuntman type, navigate by using your mouse's Right Button

Place the camera in a convenient location to see your brush as shown below:

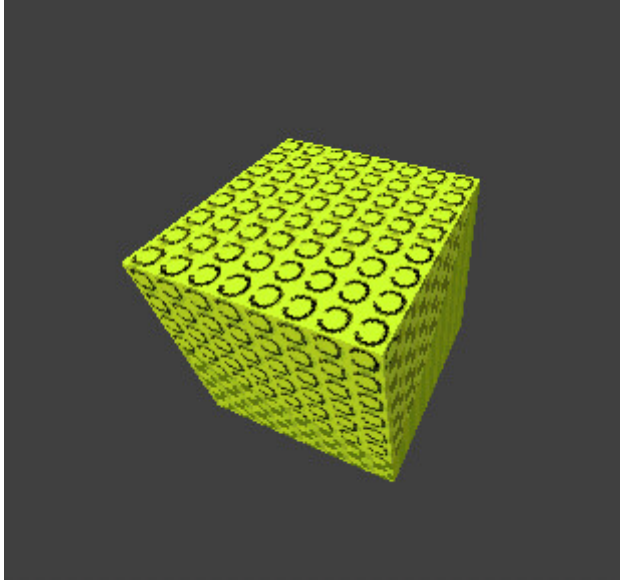


Figure 4a: Your basic cubic brush - 3D view.

Go back to the 2D view and make sure your brush is still selected. Place your cursor at the XY coordinate 128,0 (to the right of the brush) and do a **SHIFT-RightClick**. The first orange clipping dot will appear with a small 1 next to it. Then, place your cursor at the XY coordinate -128,0 (to the left of the brush) and do another **SHIFT-RightClick**. The second orange dot with a 2 next to it will appear and the lower half of your brush will disappear. But the brush is not clipped yet, it's just **ready** to be clipped at this point. This is what you should get:

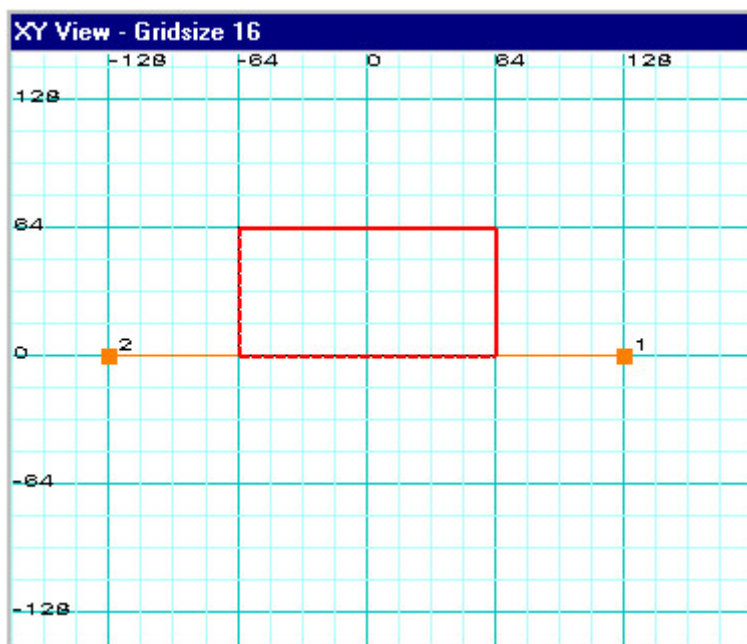


Figure 5a: The clipping plane is set - 2D top view.

And below is a 3D graphical representation of the clipping plane at work: in this case, a vertical plane since the 2 points were drawn in the XY (top) view. The outlined brush portion is the one that would remain and the other half is the part that would be clipped away if we went ahead with the actual clipping operation. This, I remind you in **not exactly** what you would see in SinEd's 3D view.

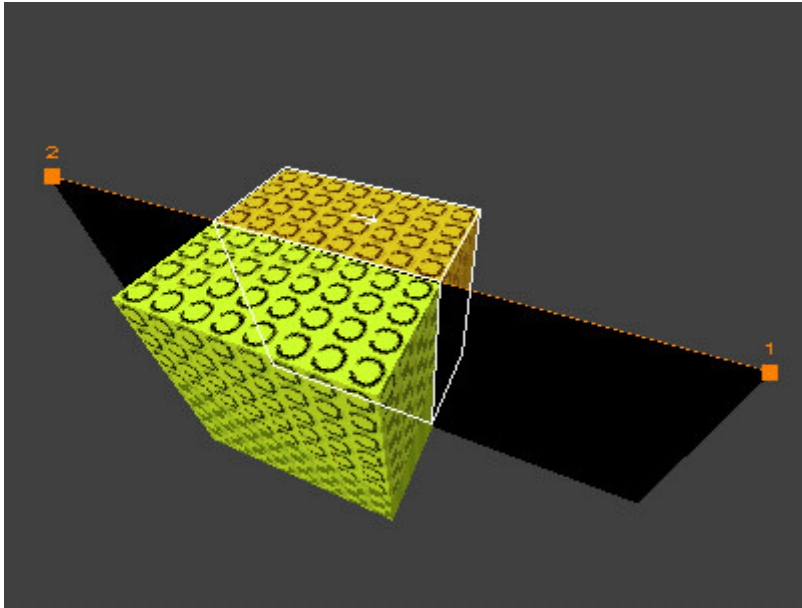


Figure 6a: The clipping plane at work - 3D view.

If for some reason, your points didn't end up in the same locations as **Figure 5a**, don't worry. SinEd lets you move those points as much as you need to before you perform the actual clipping. All you do is place your cursor over one of the dots and do a **SHIFT-RightClick & drag** while leaving the key-mouse combination pressed.

Let's try this right now. Drag point 2 to the XY coordinate 16,0 and then drag point 1 to the XY coordinate -16,0 which are both as close as we can get to the center of the brush at this grid setting. This is what you should have:

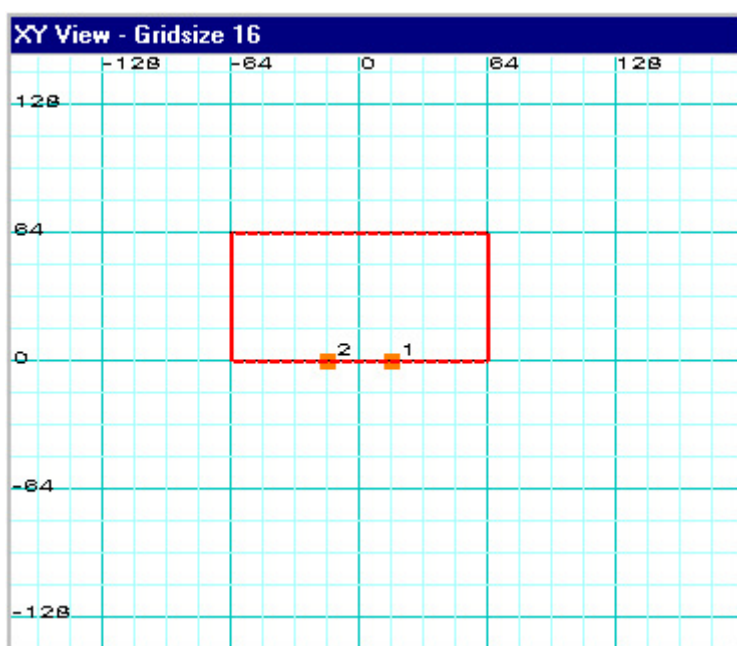


Figure 7a: Moving the points around - 2D top view.

Now if you observe closely, you will notice that even though we moved our points closer together, the size and shape of the brush that's ready to be clipped has not changed and is exactly the same in **Figure 5a and 7a**. This is an example of the principle I made a reference to in the **Basic concepts**: by definition, a clipping plane is infinite. Remember?... *A perfectly flat, infinitely thin sheet of paper that extends infinitely in all directions*. So it doesn't matter how close or how far apart your points are, what matters is the **position** and **angle** of the line **segment** formed between those points. The length of the line doesn't matter at all because the clipping plane it defines is infinite.

Figure 8a below is an attempt to illustrate this:

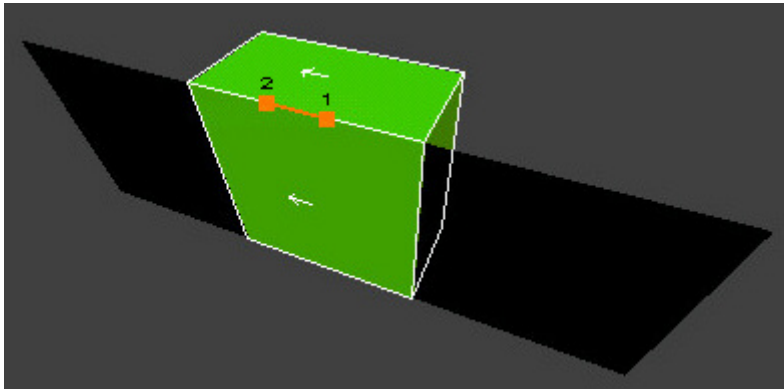


Figure 8a: The distance between the point is finite
but the clipping plane is infinite - 3D view.

Ok, now let's move our points back to their original position in **Figure 5a** before we move on to the next part.

Now, do I clip or split?

Well, that's up to you really but if you leave now, you'd be missing out on some really good stuff IMO

But seriously, it really IS up to you because at this point you can choose to clip the brush or split it in 2 separate brushes. If you want to **clip** away half of the brush, hit **ENTER**. But if you want to **split** it in 2, hit the **TAB** key. If you want to start over and draw new clipping points, just hit the **ESC** key once. This will clear your clipping points but your brush will remain selected so you're ready to draw your new points right away.

Why is this half clipped instead of the other half?

Brush clipping in SinEd uses the **clockwise rule**. This means that, by default, the part of the brush that is **clipped** away is always the one that's **above the line** traced between the clipping points when looking at their number sequence (1, 2) in a clockwise direction **in the 2D view**. I chose in the above example (**Figure 5a**) to place my points from right to left instead of from left right. I did this on purpose to illustrate this. Do you notice that the part that is clipped away in the 2D view is the bottom half? That's because that part of the brush is **effectively** above the line when following the points in a clockwise direction. Now hit the **ESC** key and draw new points at the same locations but this time, from left to right. What you should have is this:

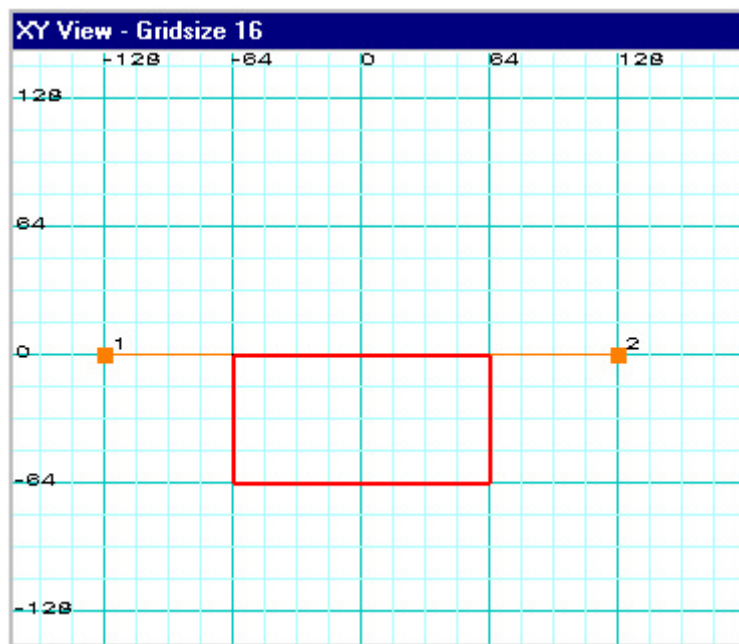


Figure 9a: The clockwise rule - 2D top view.

Now let's take a look at **Figure 5a** again and compare it with **Figure 9a** above. You can see that points 1 and 2 have been swapped between one and the other.

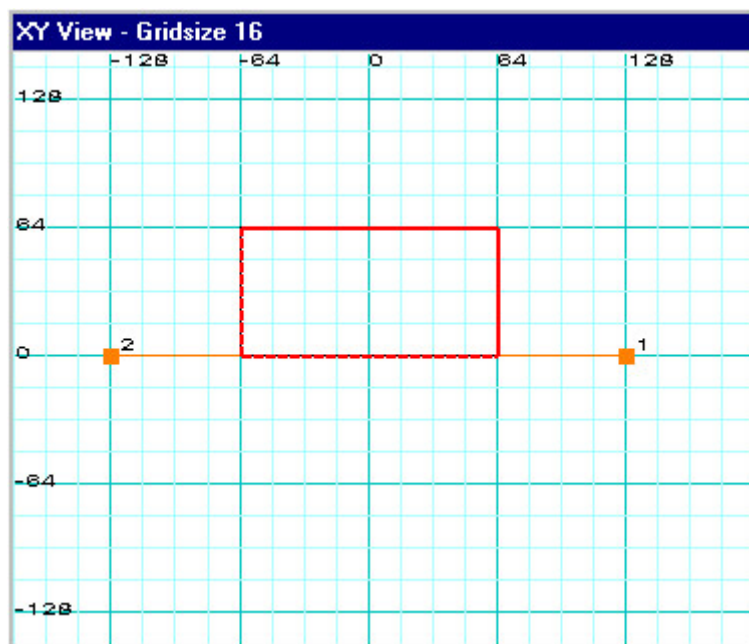


Figure 5a: The clockwise rule - 2D top view.

Clip flip

Ok, so you find this clockwise stuff confusing or you have a hard time remembering or visualizing it when working with your brushes and you often end up drawing your points in the wrong order. No sweat. Thanks to a great little feature called **clip flip**, you can easily change which part of the brush gets clipped away. If you modified your **sample.sed** project file as instructed above, you're all set. If

you didn't, go back and do it now. Once again, let's use the example in **Figure 5a** above. Now hit the **keypad slash** key and voila! Here's the result:

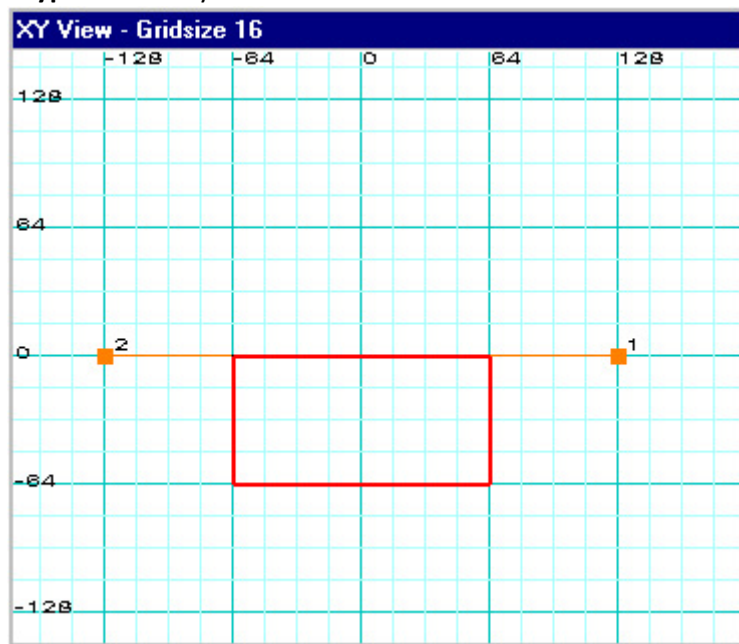


Figure 10a: Clip flip! - 2D top view.

You can see that points 1 and 2 are in the same order between **Figure 5a** and **Figure 10a** but the part of the brush that gets clipped away **flips** from one side to the other. Of course, this is a toggle type feature so if you hit **keypad slash** again, you will toggle back and forth between one side or the other of the brush. Once you made your choice, hit **ENTER** to clip.

5. Clipping brushes using horizontal planes

So far, in the examples above, we have always clipped brushes from the XY (top) view therefore using only a vertical clipping plane. But if you need to clip away the top or bottom part of the brush in 3D, you have to set a **horizontal** clipping plane. To do this, you have to draw your clip points in one of the 2 side views: **XZ** or **YZ**. If we take the same rectangular brush as in our first examples, hit the NUM LOCK key to switch to the **XZ 2D view** and draw the clipping points at the same locations as before:

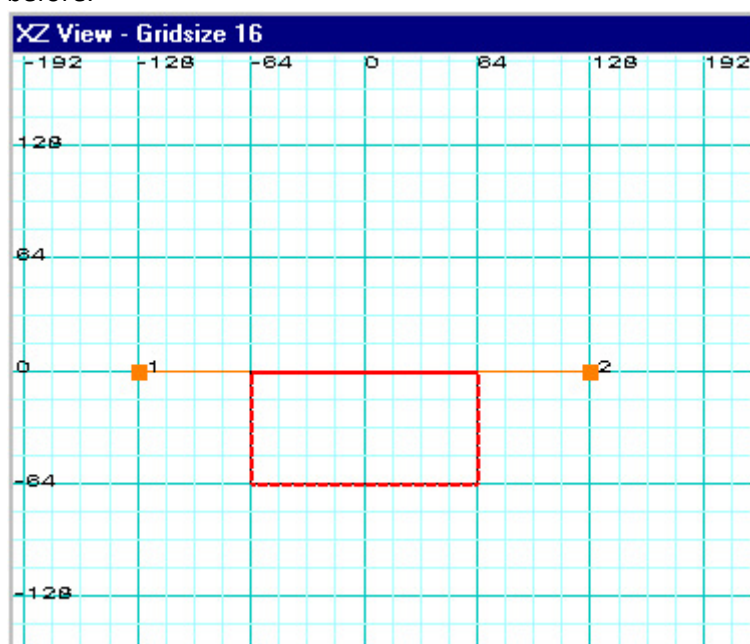


Figure 11a: Clipping with a horizontal plane - 2D XZ side view.

And below is a 3D graphical representation of the clipping plane at work: this time, a horizontal plane since the 2 points were drawn in the XZ (side) view. The outlined brush portion is the one that remains and the other half is the part that will be clipped away.

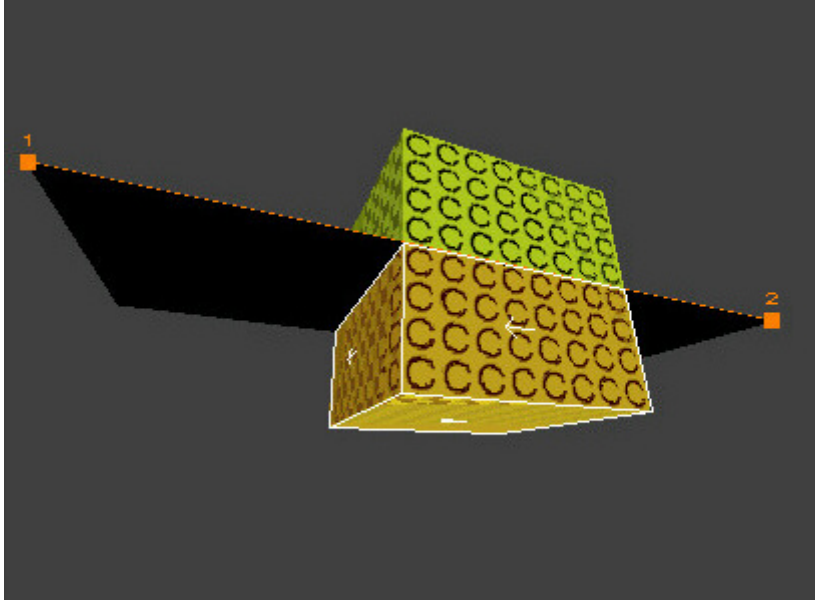


Figure 12a: The clipping plane at work - 3D view.

Of course, if you had placed the clipping points in the YZ view instead, the result would have been the same **in this case**. The main factor that determines whether to use the XZ view or the YZ view is when making angled cuts. Then, the result in the XZ view is entirely different from the result in the YZ view: a different part of the brush will be clipped away. This is covered in more detail in the next part of this tutorial.

If you're still having a hard time visualizing this, here's a **trick** to tell in what orientation your clipping plane will be defined in the **2D views**: *just imagine the line between your 2 clipping points is like the back edge of a thin blade and that the cutting edge is moving away from you and "into the screen".*

6. Clipping brushes using angled planes

At this point, you should be familiar with the basic concept of a clipping plane and 2-point clipping in SinEd. In the examples used above, the brush is always clipped using a straight cut (no angles) to make things simple to understand. But one of the most useful aspects of brush clipping is the ability to add new faces to a brush using angled cuts.

Part II of this tutorial includes an exercise showing how we can use 2-point clipping for simple angle cuts to make a complex solid like an octagon from a simple cube. So let's move on shall we?