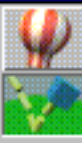


X-Ray © Matthew Bennett

This tutorial will cover how to have an object pass behind an 'X-ray' screen. All you will need is TS 4.1. I am not going to cover how to set up a sequence of numbered frames to use as a texture here, if you haven't done that, then please read the 'Animated Textures' tutorial listed on the right.

Overview: For this tutorial, I will use a bouncing ball that goes behind an X-ray screen. The camera will keep pace with the ball. It is critical to use a camera for this to work. The scene is going to be rendered three times. Once with solid rendered frames, one set of frames for masking, and the third time with the wire-rendered frames. Each rendering will require a different set up. So be sure to make copies of the scene files.

First, the bouncing ball. Create a sphere, not too dense a mesh, about 12x10 or so is sufficient. Raise it up a bit above the ground. Since I don't want the ball to slowly bounce to a stop, I set the resistance and friction to zero in the numerical physical attributes panel and the elasticity to 100. I used the generic Glass physical attributes for a mass setting, and added an initial motion vector of 2.62. (so that it doesn't just bounce in place)

numerical physical attributes		
Mass	31.885	
Elasticity	100	
Resistance	0	
Friction	0	

With that done, hit the start physical simulation button

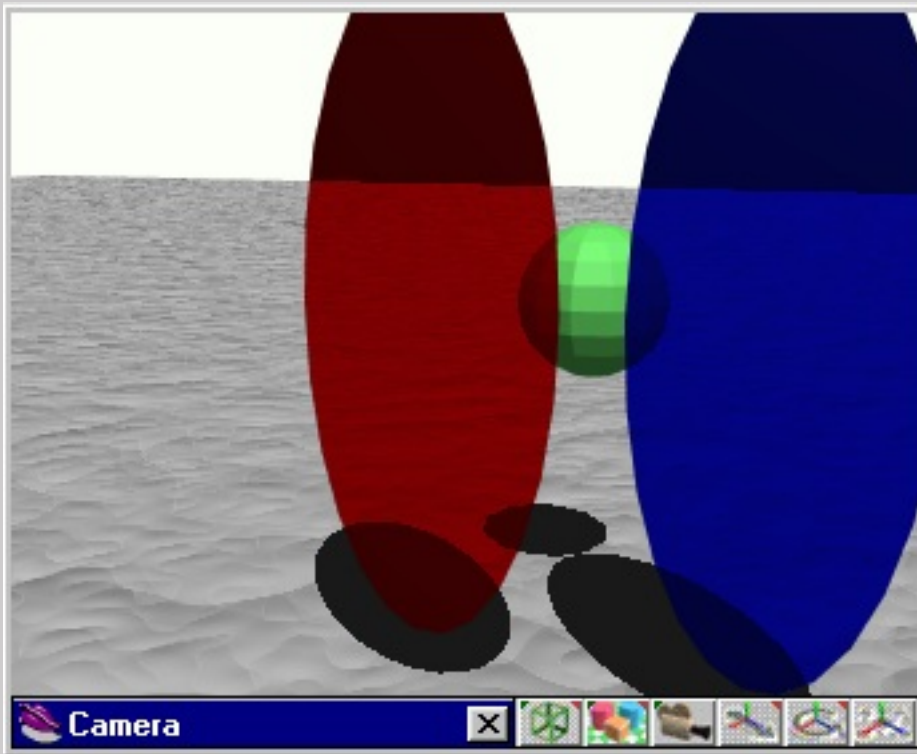


, and let it record about 10 seconds worth of frames (or however many you think you need). I usually let it calculate for longer than I need, just in case... (I ended up only using about 2.5 seconds worth.)

The next step is to place a camera that will keep pace with the bouncing ball. Since we took care of the slowing down of the ball, it should pretty closely move at constant speed. Position the camera at frame zero to be looking at the ball. If you used the 'look at' button, be sure to turn it off. Having the camera looking at a bouncing ball with the look-at feature on will cause a rather nauseating effect. Advance to the desired end frame, and move the camera in a parallel path so that it is again looking at the ball. Having a window open set to 'camera view' is handy for positioning the camera.

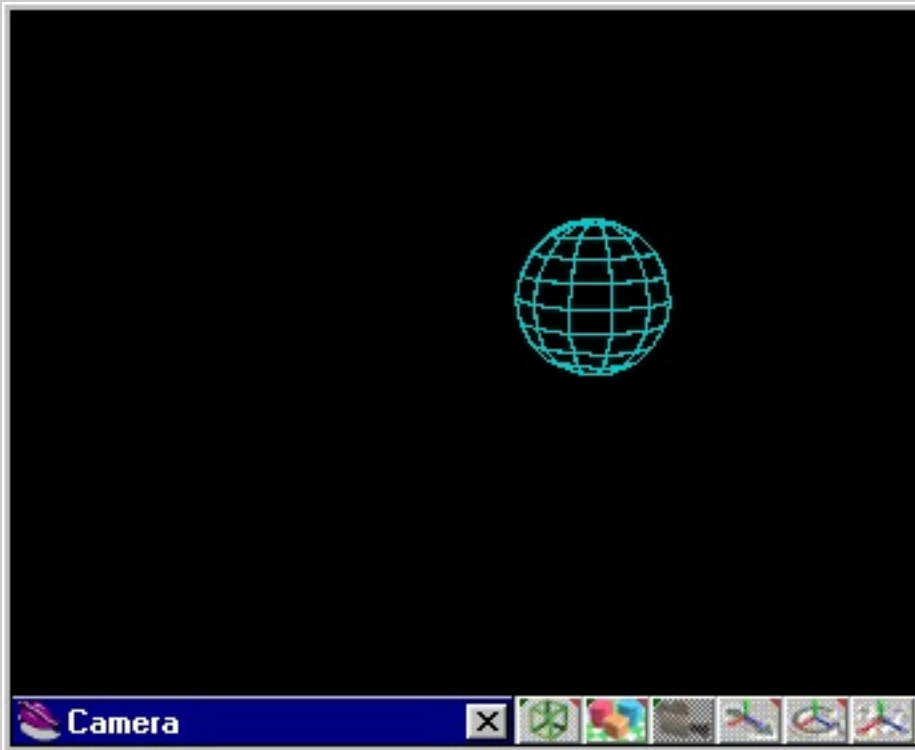
The last piece to put in is the x-ray screen. Just for fun, I am going to use a pair of ellipsoid screens.

The first scene set up is now complete. Below is a screen shot of frame 43. Nothing too fancy, but good for a tutorial.



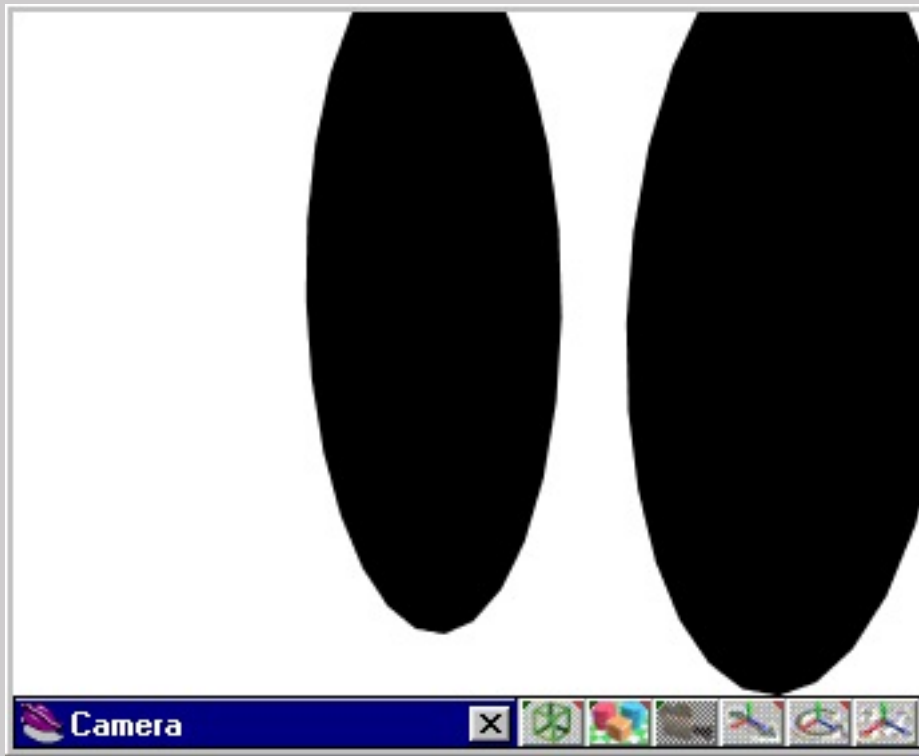
This scene should be rendered out to a set of frames. I usually try to keep each set of frames in an appropriately named folder (by the time we are done you should have about 4 folders full of frames).

The next step is to render the wire frame ball. Delete everything in the scene except the ball and the camera (you made a copy of the scene first right?). You can change the color of the wire frame by changing the 'Select' color in the display options panel. Set the background to black, and choose Hidden Line in the scene render options panel. Render out the frames to individually number .tga images. Below is what frame 43 looks like with just the wire frame ball. Not too exciting.



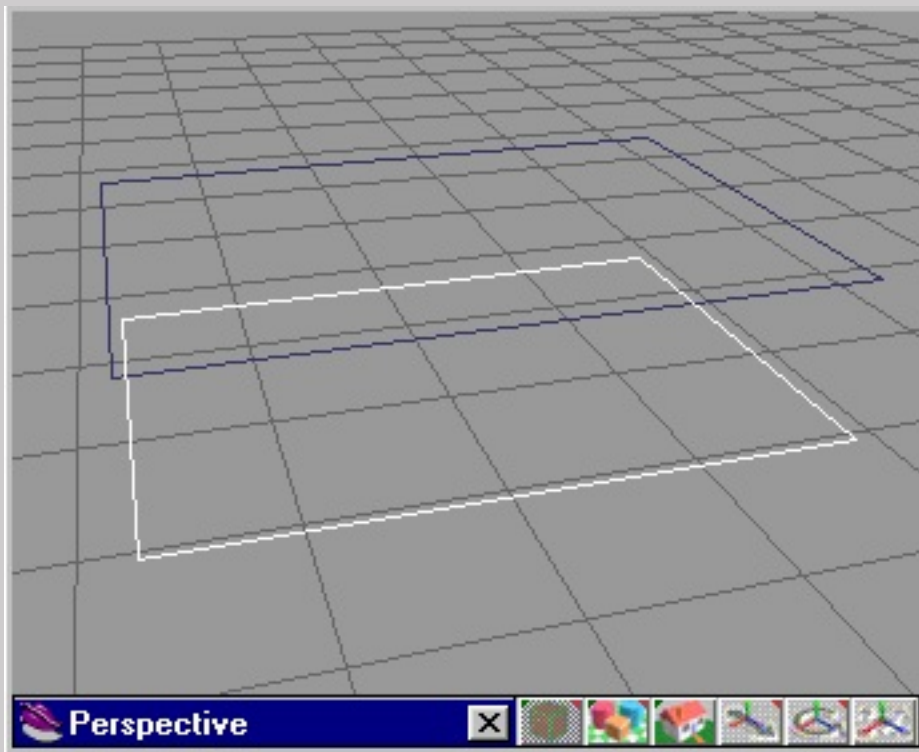
The next step is to make the masking images. These will allow the wire frame ball to only be seen when we want it. With the third copy of the full scene, delete everything but the screens and camera. Set the background to white, and change the color of the screens to flat black. Render out those frames to individual .tga images as well. Below is frame 43 again.

Note: you could only render out the frames where the ball is passing behind the screens, however that would take a little more planning and thought when setting up the sequence of frames



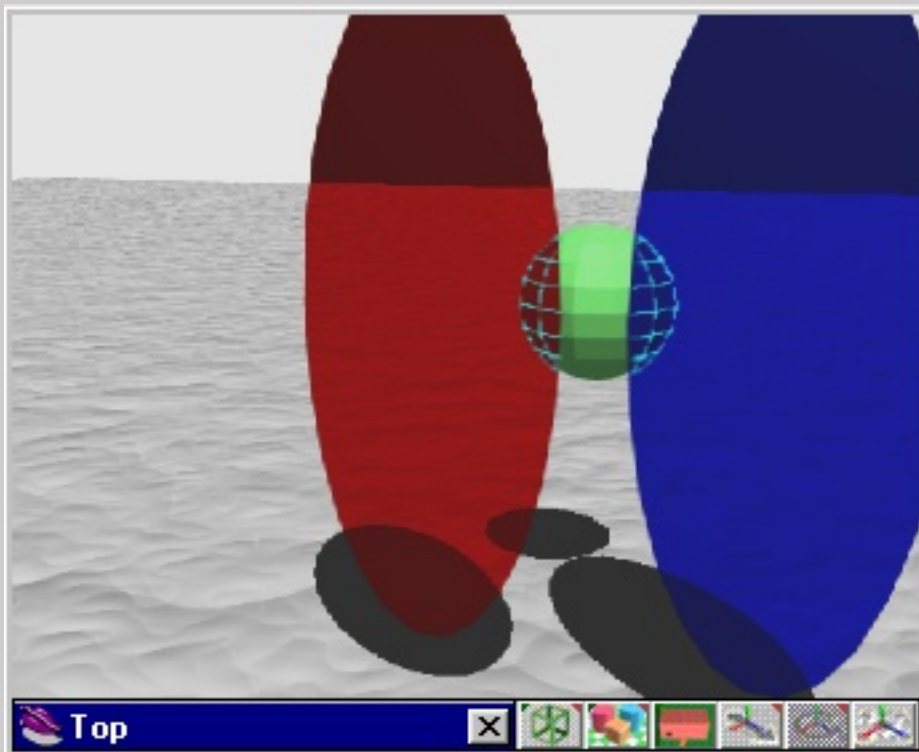
Now we will combine the wire frame image with the mask image to end up with a sequence of frames that shows the wire ball only when it is behind the screens.

With two planes sized to 3.42×2.56 , apply the sequence of wire frame images to the plane on the bottom. The top plane will need to have a plain color of black, and then apply the sequence of ellipse masks as a wrapped transparency filter. Setting a sequence of frames as a transparency mask works exactly the same as using them as a texture map. For more in depth coverage of this see the [Animated Textures](#) tutorial listed on the right. This scene will be rendered from a top view, zoomed in so that the planes take up exactly the size of the view. The first picture below is what the set up looks like, and after that is our friend, frame 43.



We are now ready to put the final pieces together. We will basically repeat the last step, but this time the bottom frame will have the solid rendered frames, and the top will have the resulting frames of the last render.

Start a new scene, put in two planes sized to 3.42 x 2.56. On the bottom frame apply the sequence of solid rendered frames as the texture map. For the top frame, we will be using the rendered frames from the last step. They will be used as the texture map, and as the wrapped transparency filter. Both need to be set at the same time when setting the series of frames. Below is the final version of frame 43.



I used a similar technique to create the Hopper Bot animation that won me a runner-up award in Caligari's March Animation contest. I won't go into the differences, but basically the technique described here is better.

[Close](#)