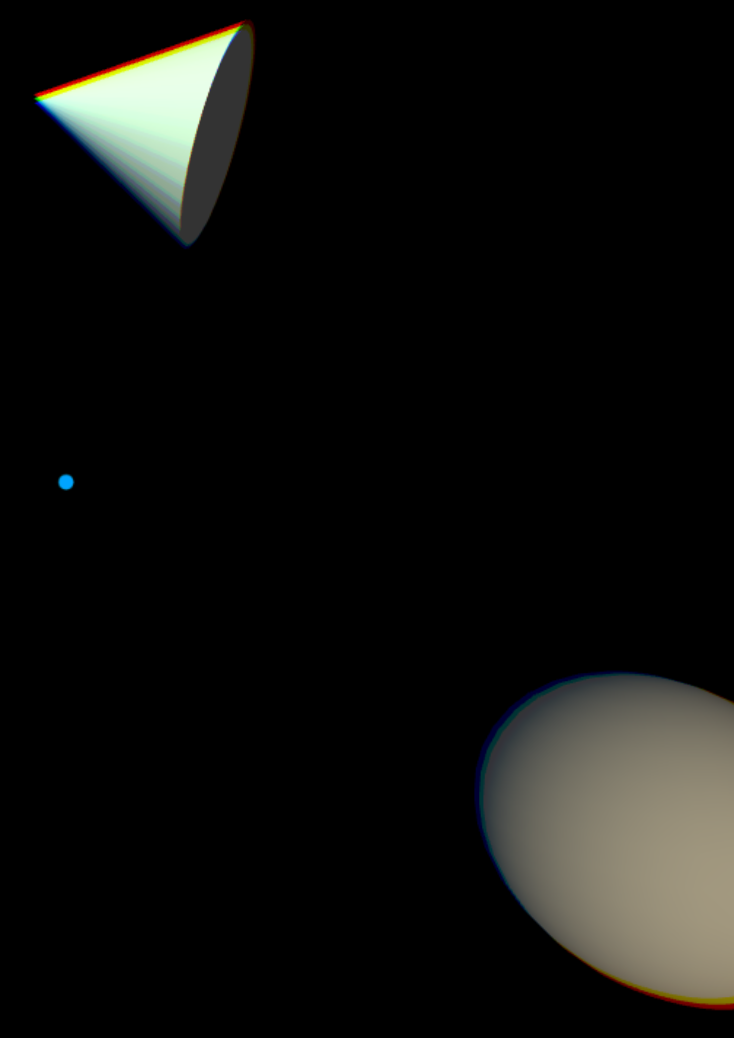
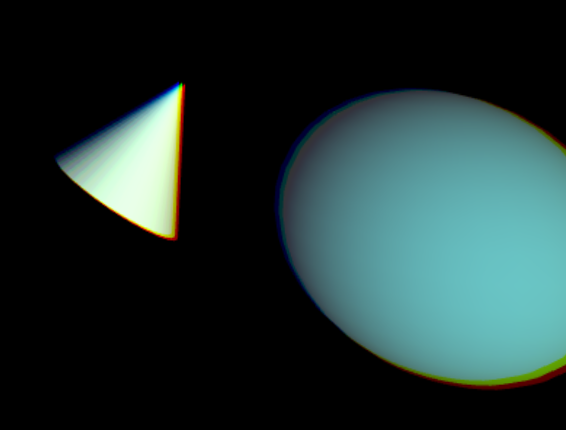
**Per Frame Scripting**

These are demonstrations with the <SCRIPT> node that contains JavaScript and is called for every frame. In each case the <TimeSensor>’s time value is routed to the <SCRIPT> node. The time will be a value between 0 and the cycleInterval. Thus if the cycleInterval is 10 seconds, then the value sent to the <SCRIPT> node will be between 0 and 10, and repeat if the <TimeSensor>’s ‘loop’ Boolean is set to ‘true’. Users are encouraged to modify the JavaScript code.

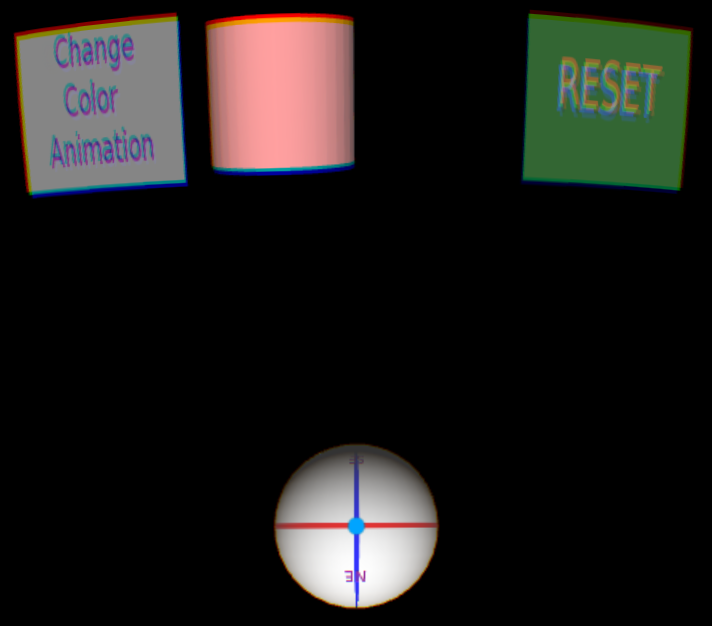
**JavaScript\_PerFrame\_01\_ProceduralAnim.x3d**

There are two objects a cone and a sphere, both animated by JavaScript. The cone follows a sine curve and the sphere moves from left to right to left. The sphere also changes color controlled inside the <SCRIPT> node’s JavaScript. There is no interactivity.

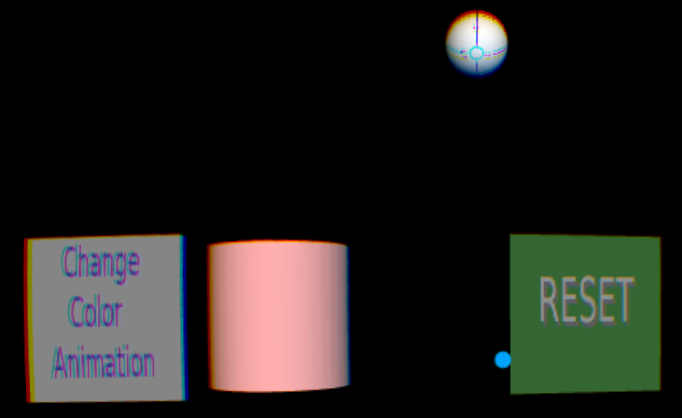


**JavaScript\_PerFrame\_02\_LaunchSphere.x3d**

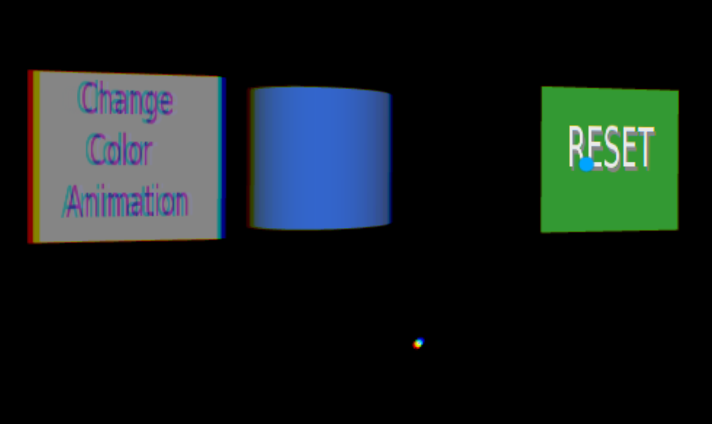
The demo begins with a sphere remaining in front of the viewer. The Cylinder color animates from red to white.



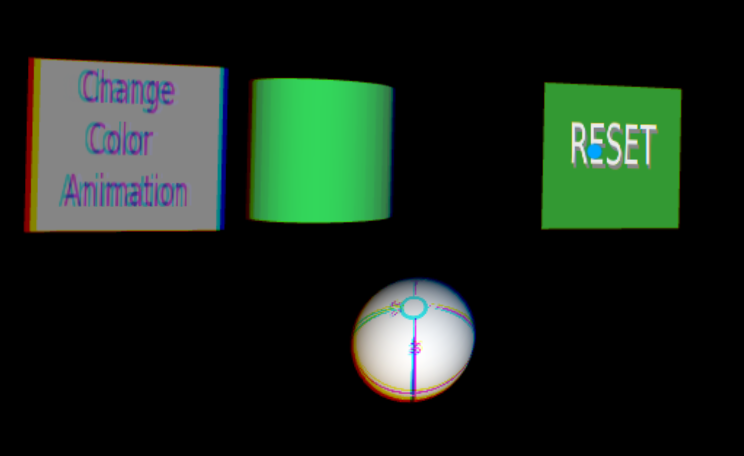
Tapping the GearVR Touch Pad launches the sphere as it is no longer directly in front of the user. Tapping the Touch Pad again when the user view (blue dot) is over the sphere ‘captures’ the sphere and it is now in front of the viewer, but further out.



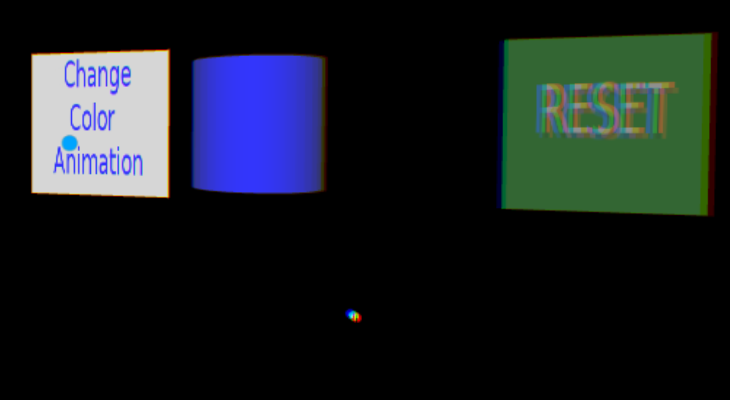
Rolling over “Reset” highlights the button as a visual cue to the user that it’s interactive.



Clicking on “Reset” brings the sphere back.

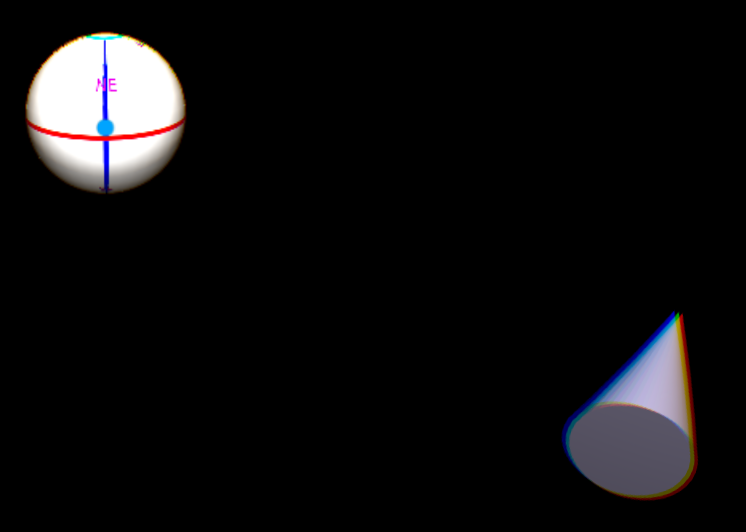


Rolling over the “Change Color Animation” button highlight it as a visual cue. Clicking it changes the animated colors over the sphere to be from green to blue. Clicking it again changes it back to animate it between red and white.



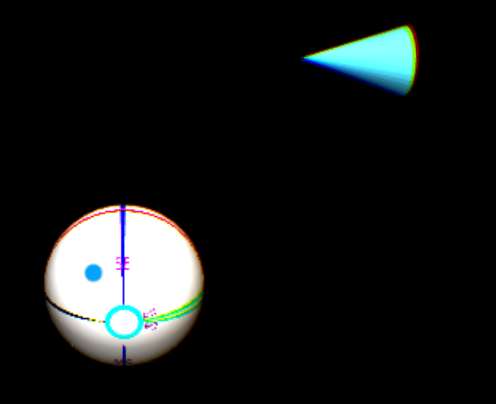
**JavaScript\_PerFrame\_03\_TimeStamp.x3d**

The Cone navigates a rectangular pattern using Interpolators (key frame animation), not the <SCRIPT> node. This helps test that previous functionality still works.



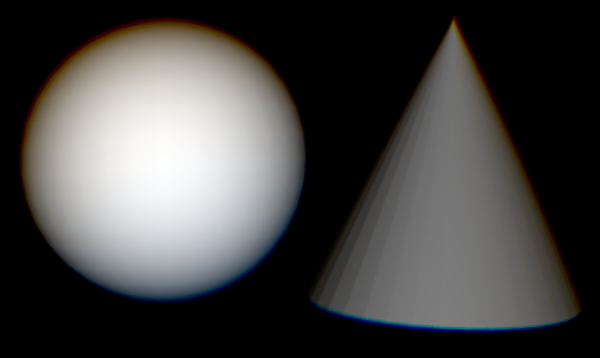
The color of the cone is however controlled by the <SCRIPT> node using JavaScript to cycle through various colors.

This demo also uses the ‘timestamp’ parameter that can be optionally passed to the <SCRIPT> node with the current time. [Normally, only one parameter is passed, either a true/false value, or the <TimeSensor> time value]. The timestamp value controls the sphere’s rotation around the x-axis.

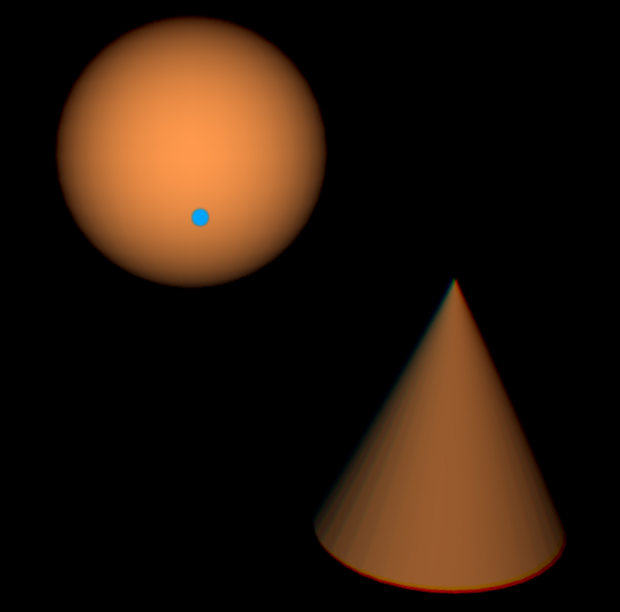
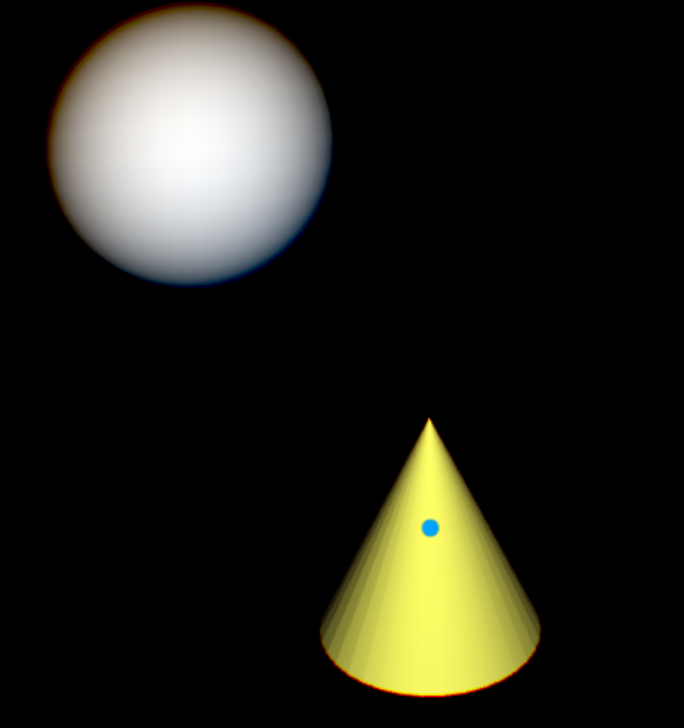


**JavaScript\_PerFrame\_04\_GrabAnObject.x3d**

Initial scene has a white Sphere and gray Cone.

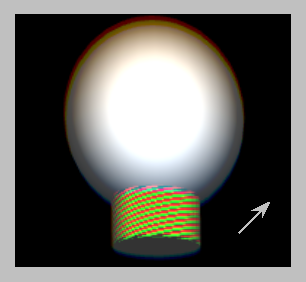
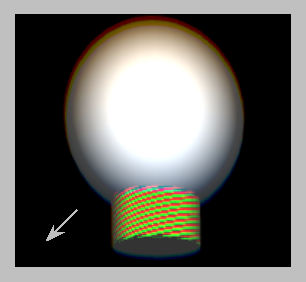
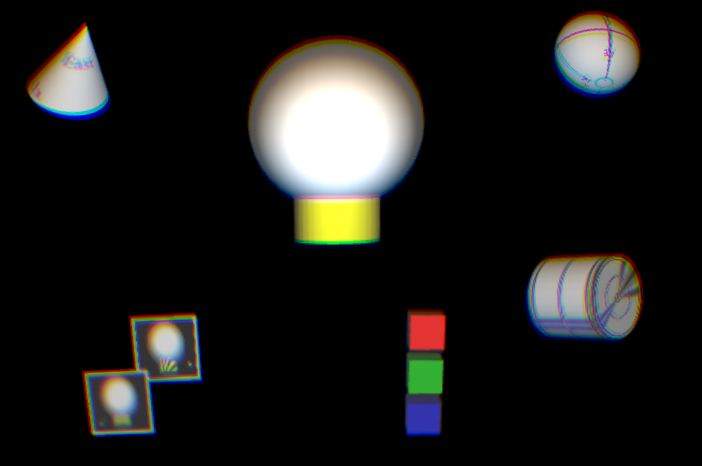


Clicking on the Cone turns it yellow and allows the user to drag it anywhere in the X-Y plane. Clicking on it again reverts it back to gray and ends the dragging. Clicking on the Sphere changes the white Point Light to orange. Clicking again reverts the change back to a white point light.



**JavaScript\_PerFrame\_05\_LightControls.x3d**

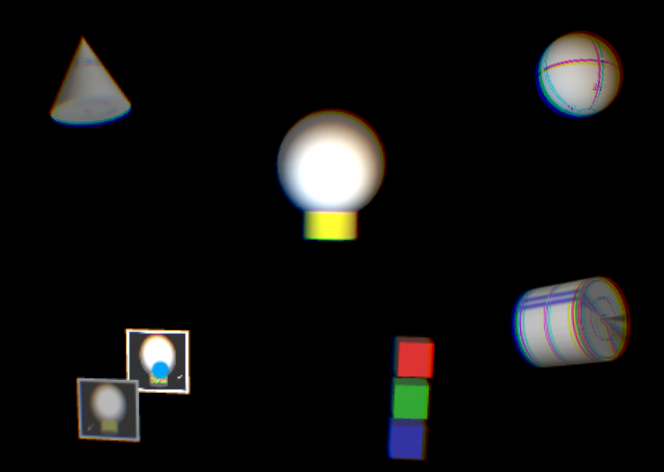
This demo enables controlling the location and color of a Point Light, represented by a light bulb. The scene has a Cone and Cylinder in perpetual animation. There are 3 Boxes toward the bottom – red, green, blue – which controls the Point Light color.



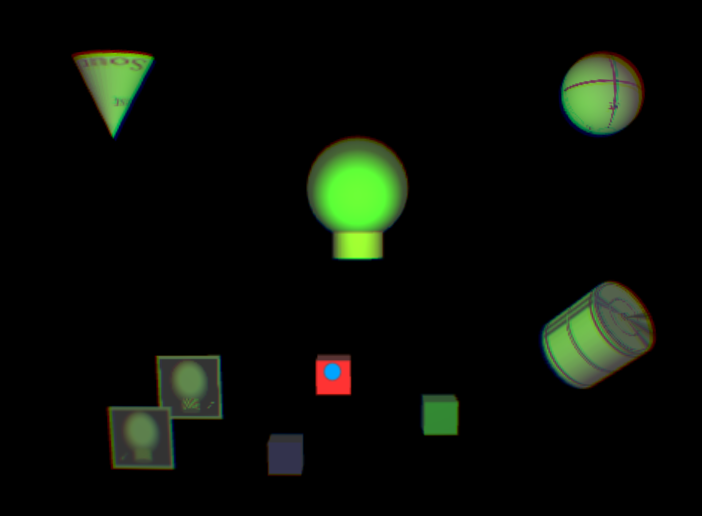
There are two textured planes in the lower left of the scene.

Being over the plane with the arrow pointing toward the lower left (shown on the left) will move the light bulb and Point Light forward (positive Z direction).

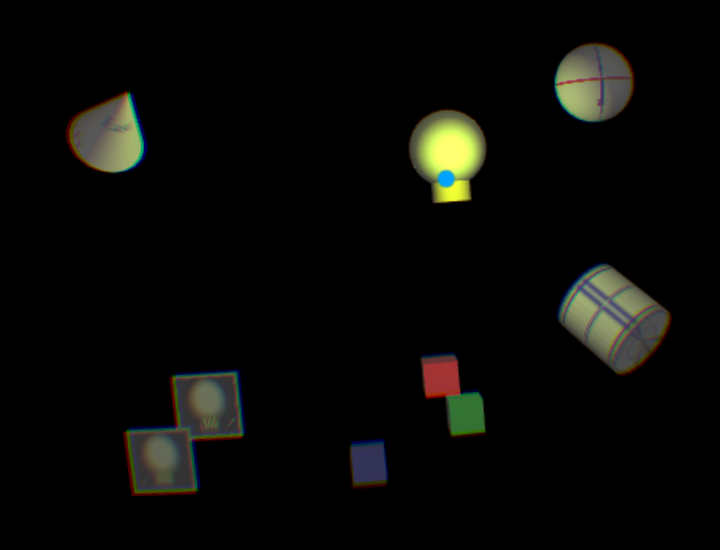
Being over the plane with the arrow pointing toward the upper right (shown on the right above and image below) will move the light bulb and Point Light back (negative Z direction).



Dragging the red, green and blue Boxes will adjust those colors of the Point Light and light bulb. In the image below, blue has already been dragged to 0 while red is reduced leaving a mostly green light.



Clicking on the light bulb will drag it and the Point Light left and right (X and Y axis). Must click again to stop dragging the light bulb.



The Sphere in the upper right will rotate the same amount as the camera.