

### 1./ axis-angle

Bài tập trực/góc: thêm ba hình trụ nữa.

viết Function to add a cylinder to the scene -> function addCylinder(x, y, z, rotAxis, angle) xong sau đó thêm vào dàn và điều chỉnh các góc quay tương ứng

### 2./ Cylinder Positioning

### 3./ Capsule (hình viên nang)

Yêu cầu: hoàn thành hình viên nang hoàn chỉnh bằng cách thêm khối cầu (Sphere)

//Solution: replace sphGeom on down in createCapsule():

```
var capsule = new THREE.Object3D();
capsule.add( cyl );
if ( !openTop || !openBottom ) {
    // instance geometry
    var sphGeom = new THREE.SphereGeometry( radius, segmentsWidth, segmentsWidth/2
);
    if ( !openTop ) {
        var sphTop = new THREE.Mesh( sphGeom, material );
        sphTop.position.set( top.x, top.y, top.z );
        capsule.add( sphTop );
    }
    if ( !openBottom ) {
        var sphBottom = new THREE.Mesh( sphGeom, material );
        sphBottom.position.set( bottom.x, bottom.y, bottom.z );
        capsule.add( sphBottom );
    }
}
return capsule;
```

Explain:

I implemented the solution by creating an object in which to put my cylinder and two spheres. At the top there's a little logic here to determine whether any sphere is needed at all. If so, I create the sphere's geometry. Then, each end is checked to see if it's used. If it is, a sphere object is created with the geometry, positioned and added to the capsule, which is then returned. That's it.

- dòng code `if ( !openTop || !openBottom )` dùng để kiểm tra xem phần đầu và phần cuối của vật thể có đang trống không, nếu có thì tạo ra các hình cầu, sau đó thêm vào chỗ còn thiếu ( phần đầu cuối). kết thúc sẽ trả về hình viên nang tương ứng.

#### 4./ Helices (Hình xoắn ốc)

yêu cầu: thay thế các khối cầu bằng các hình viên nang. Điều chỉnh function `createHelix` để làm điều này

Solution: replace helix code in `createHelix()`:

```
var helix = new THREE.Object3D();
var bottom = new THREE.Vector3();
var top = new THREE.Vector3();
var openBottom = false;
var openTop = false;
var sine_sign = clockwise ? 1 : -1;
bottom.set( radius, -height/2, 0 );
for ( var i = 1; i <= arc*radialSegments ; i++ )
{
    // going from X to Z axis
    top.set( radius *Math.cos( i* 2*Math.PI / radialSegments ),
            height *(i/(arc*radialSegments)) - height/2,
            sine_sign *radius* Math.sin( i *2*Math.PI / radialSegments ) );
    var capsule = createCapsule( material, tube, top, bottom, tubularSegments, openTop,
openBottom );
    helix.add( capsule );
    // after first capsule is laid down, don't need to draw sphere for bottom.
    openBottom = true;
    // make top of previous capsule the bottom of the next one
    bottom.copy( top );
}
```

```
}
```

```
return helix;
```

Giải thích thuật toán:

Thuật toán:

Hàm khởi tạo một đối tượng mới THREE.Object3D để chứa các viên nang.

Nó đặt điểm dưới cùng của viên nang đầu tiên tại điểm bắt đầu của helix.

Một vòng lặp tạo từng đoạn của helix:

Nó tính toán vị trí của đỉnh viên nang tiếp theo trong helix dựa trên góc hiện tại xung quanh trục trung tâm của helix (radialSegments) và chiều cao dọc theo trục Y của helix.

Nó tạo ra một viên nang (createCapsule) từ điểm dưới cùng hiện tại đến điểm đỉnh được tính toán.

Nó thêm viên nang vào đối tượng helix.

Điểm dưới cùng của viên nang tiếp theo được thiết lập là đỉnh của viên nang hiện tại.

openBottom được thiết lập thành true sau khi viên nang đầu tiên được tạo ra để ngăn không vẽ phần hình cầu dưới cùng của các viên nang tiếp theo, vì chúng sẽ được kết nối với viên nang trước đó.

Kết quả:

Hàm trả về một đối tượng THREE.Object3D (helix) chứa một chuỗi các viên nang được kết nối tạo thành cấu trúc helix.

điểm bắt đầu của helix (đường xoắn ốc) được xác định bởi biến bottom

## 5./ Drinking Bird

yêu cầu :thêm mắt, mũi và thanh ngang. Nhiệm vụ của bạn là chỉnh sửa hàm createDrinkingBird khoảng dòng 163.

Solution: add code at end of createDrinkingBird():

```
//crossbar
```

```
cylinder = new THREE.Mesh( new THREE.CylinderGeometry( 5, 5, 200, 32 ), crossbarMaterial );
```

```
cylinder.position.set( 0, 360, 0 );
```

```
cylinder.rotation.x = 90 * Math.PI / 180.0; scene.add( cylinder );
```

```
//nose
```

```
cylinder = new THREE.Mesh( new THREE.CylinderGeometry( 6, 14, 70, 32 ), headMaterial );
```

```
cylinder.position.set( -70, 530, 0 );
```

```
cylinder.rotation.z = 90 * Math.PI / 180.0; scene.add( cylinder );  
  
//eyes  
  
var sphGeom = new THREE.SphereGeometry( 10, 32, 16 );  
  
//left eye  
  
sphere = new THREE.Mesh( sphGeom, eyeMaterial );  
sphere.position.set( -48, 560, 0 );  
var eye = new THREE.Object3D();  
eye.add( sphere );  
eye.rotation.y = 20 * Math.PI / 180.0;  
scene.add( eye );  
  
//right eye  
  
sphere = new THREE.Mesh( sphGeom, eyeMaterial );  
sphere.position.set( -48, 560, 0 );  
eye = new THREE.Object3D(); eye.add( sphere );  
eye.rotation.y = -20 * Math.PI / 180.0;  
scene.add( eye );
```