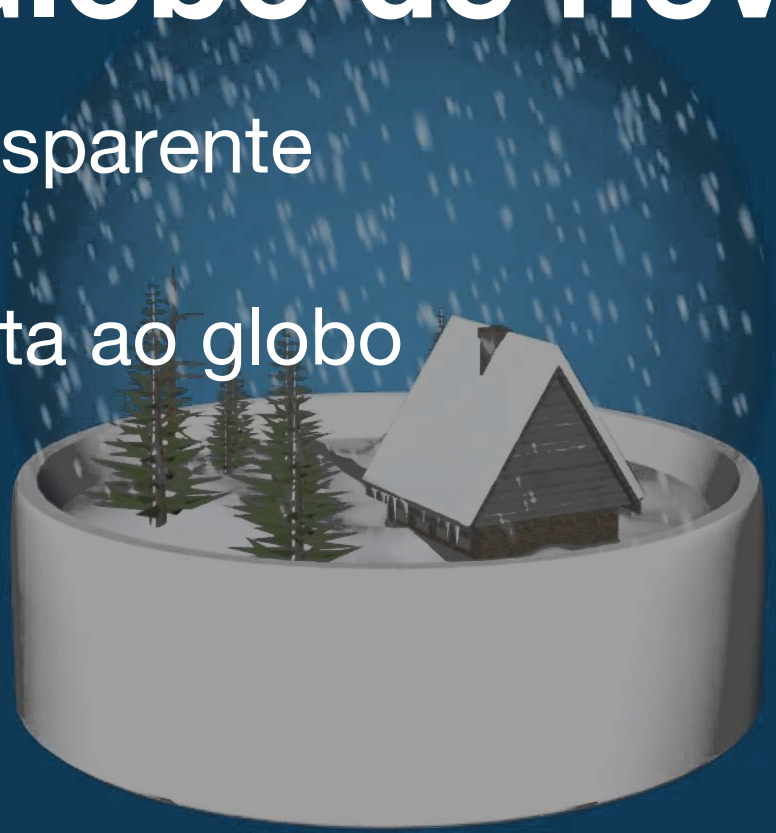
Globo de neve

Globo de neve

- Esfera transparente

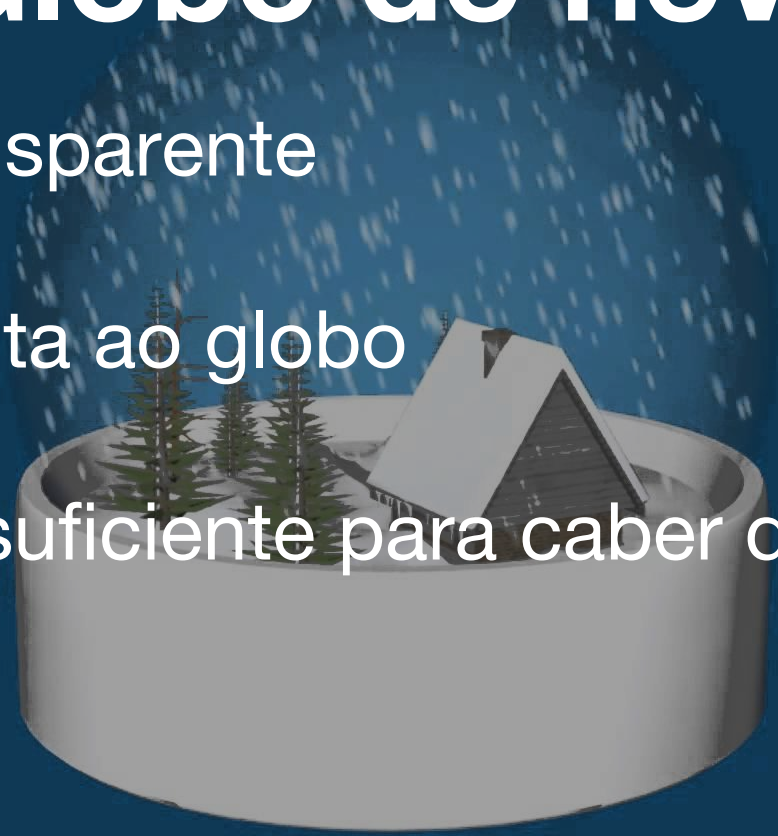
Globo de neve

- Esfera transparente
- Neve restrita ao globo



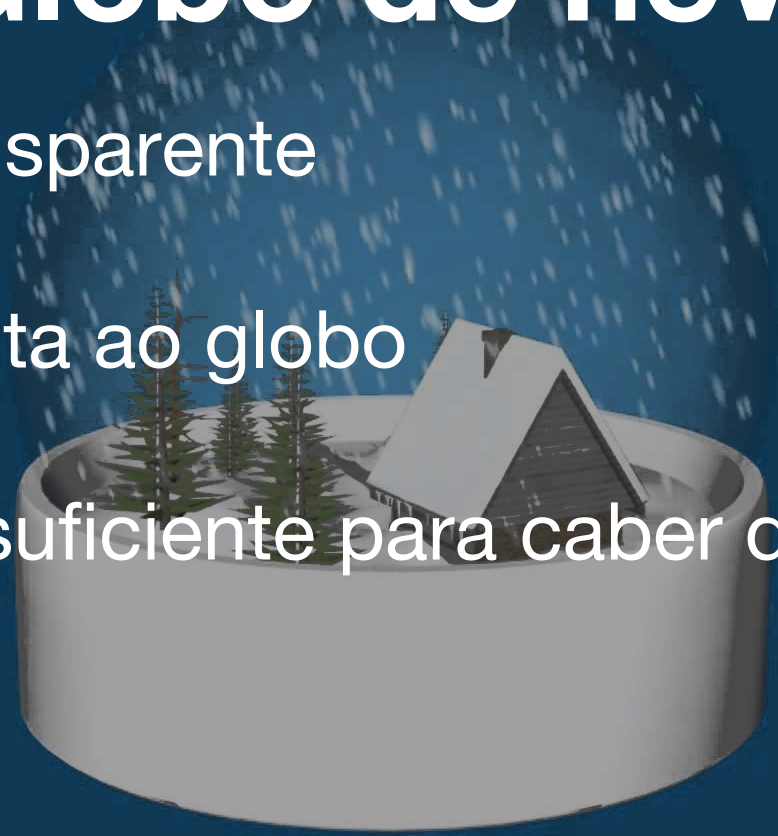
Globo de neve

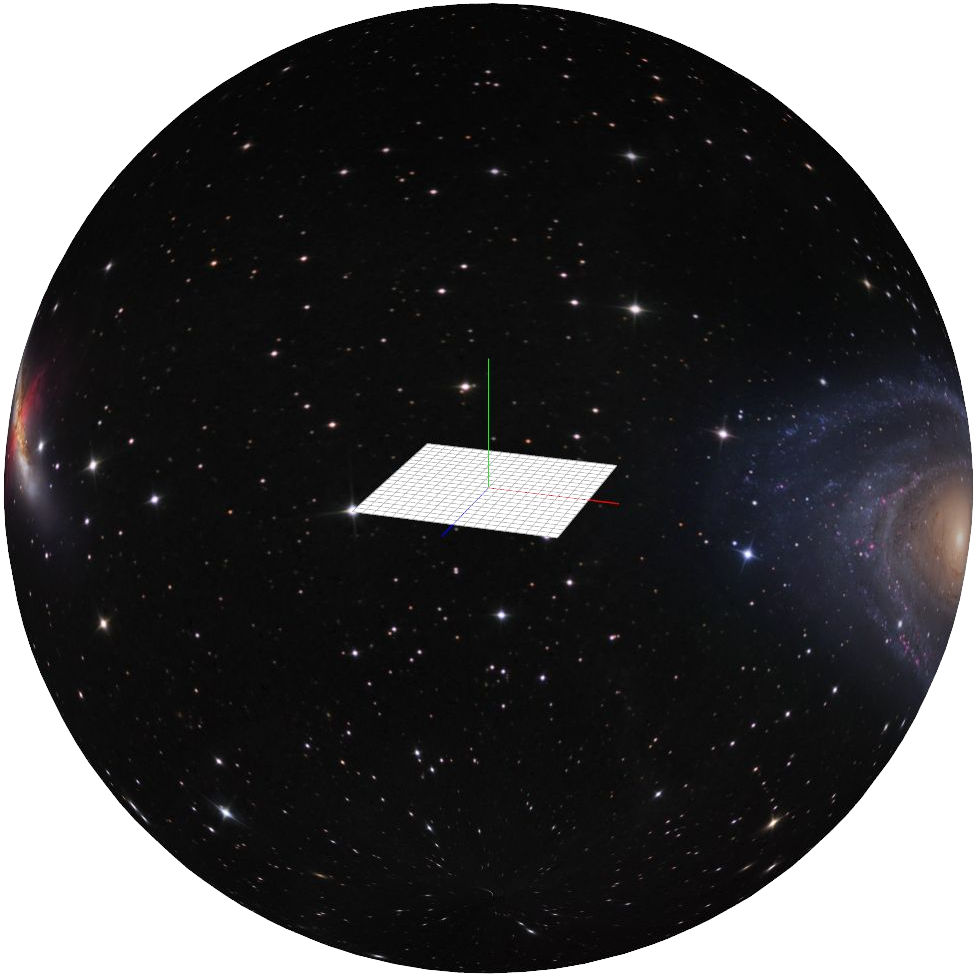
- Esfera transparente
- Neve restrita ao globo
- Grande o suficiente para caber dentro



Globo de neve

- Esfera transparente
- Neve restrita ao globo
- Grande o suficiente para caber dentro
- ???

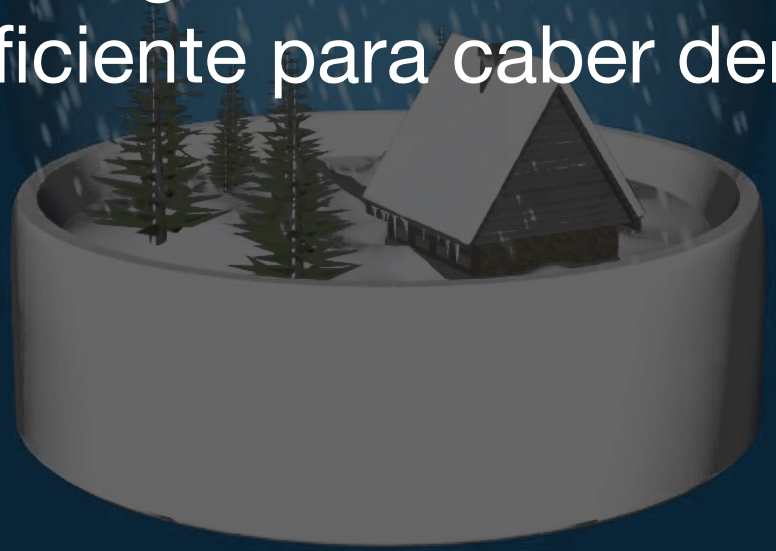




SkyGlobe Alienígena

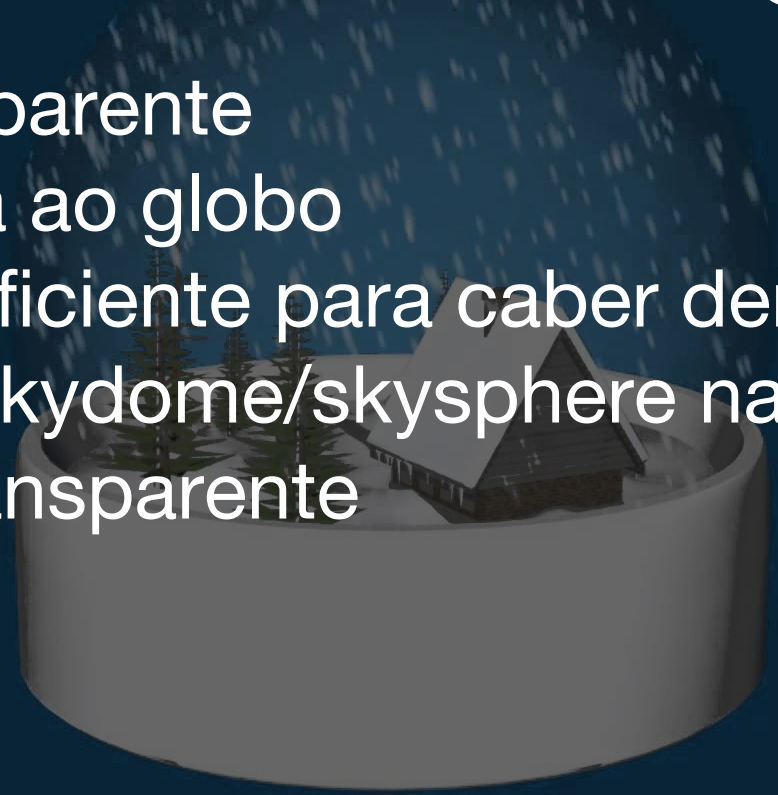
SkyGlobe Alienígena

- Esfera transparente
- Neve restrita ao globo
- Grande o suficiente para caber dentro



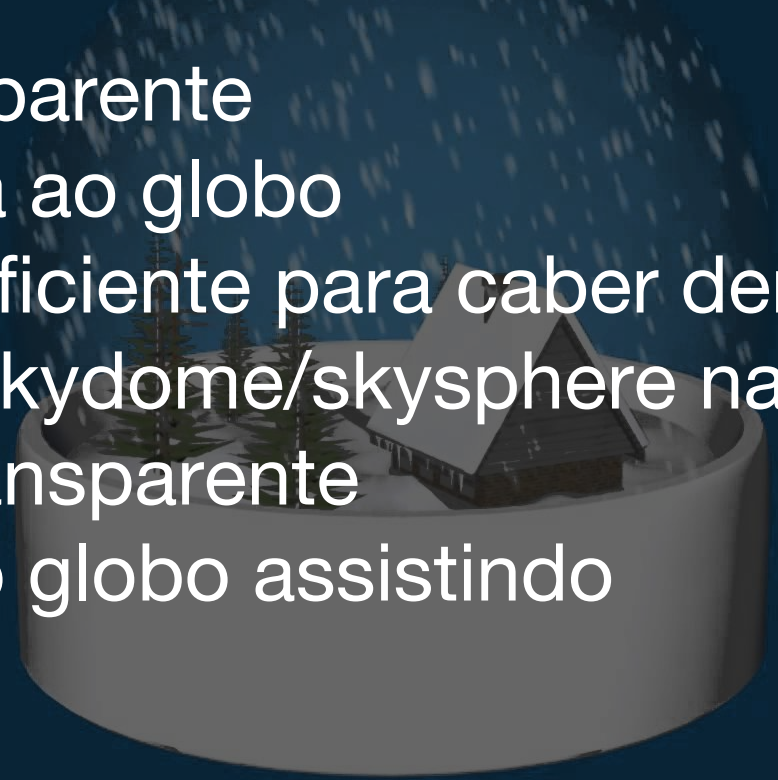
SkyGlobe Alienígena

- Esfera transparente
- Neve restrita ao globo
- Grande o suficiente para caber dentro
- Textura de skydome/skysphere na superficie da esfera transparente



SkyGlobe Alienígena

- Esfera transparente
- Neve restrita ao globo
- Grande o suficiente para caber dentro
- Textura de skydome/skysphere na superficie da esfera transparente
- Alien fora do globo assistindo



Criando a Neve

- Criando material e geometria

```
var particles = new THREE.Geometry();
var pMaterial = new THREE.PointsMaterial({
      color: 0xFFFFFF,
      size: particle_size,
      map: THREE.ImageUtils.loadTexture( "snowflake.png" ),
      blending: THREE.AdditiveBlending,
      transparent: true
    });
pMaterial.depthWrite = false;
```

Criando a Neve

- Criando cada particula

```
// Creating individual particles
for (var p = 0; p < particle_count; p++) {
  // create a particle with random position inside sphere
  var position = randomPointInsideSphere(radius - particle_size);
      particle = new THREE.Vector3(position.x, position.y, position.z);
  // add it to the geometry
  particles.vertices.push(particle);
}
```

Criando a Neve

- Gerando posição aleatória dentro da Esfera

```
function randomPointInsideSphere(radius){
  var theta = 2 * Math.PI * Math.random();
  var phi = Math.acos((2 * Math.random() - 1) * Math.PI/2);
  var u = Math.min(Math.random() * radius * 3, radius);
  var x = u * Math.sin(phi) * Math.cos(theta);
  var y = u * Math.sin(phi) * Math.sin(theta);
  var z = u * Math.cos(phi);
  return { x: x, y: y, z: z };
}
```

Criando a Neve

- Finalmente criando o sistema de particulas (Points)

```
// create the particle system
var particleSystem = new THREE.Points( particles, pMaterial);

return {
  particle_size: particle_size,
  particle_count: particle_count,
  radius: radius - particle_size,
  particleSystem: particleSystem
};
```

Movimentando a Neve

```
function render_snow(snow) {
  var pCount = 0;
  while(pCount < snow.particle_count) {
    var particle = snow.particleSystem.geometry.vertices[pCount];

    square_d = (particle.x*particle.x + particle.y*particle.y + particle.z*particle.z);
    if(square_d > snow.radius*snow.radius) {
      var new_point = randomPointOnTopOfSphere(snow.radius);
      particle.set(new_point.x, new_point.y, new_point.z);
    }

    particle.y -= .1;
    pCount++;
  }

  snow.particleSystem.geometry.verticesNeedUpdate = true;
}
```

Movimentando a Neve

- Gerando posilção apenas no topo.

```
function randomPointOnTopOfSphere(radius){
  var theta = Math.PI * Math.random();
  var phi = Math.acos((2 * Math.random() - 1) * Math.PI/2);
  var x = radius * Math.sin(phi) * Math.cos(theta);
  var y = radius * Math.sin(phi) * Math.sin(theta);
  var z = radius * Math.cos(phi);
  return { x: x, y: y, z: z };
}
```

:)