CSC 240
Computer Graphics

Sara Mathieson
Fall 2016
Smith College
Outline: 12/7

- Solar System demos
- Collision Detection
- Introduction to Animation
- Blender Lab (Bezier curves)

- **Final Project**: due Thurs Dec 14
- **3D printing**: only a few spots left!
- **Office Hours**: Mon 4-5pm (015 Ford)
  Tues 4-5pm (346 Ford)
  can also come: Thurs 4-5pm
  TA hours end when classes end
Solar System Demos
Solar System Images

Perla

Prayasha

Sam

Anjali
Collision Detection

Modified from demo: http://stemkoski.github.io/Three.js/Collision-Detection.html
Setup up collidable objects

```javascript
var movingCube;
var collidableMeshList = []; // objects the movingCube can collide with

// first purple box
var wall = new THREE.Mesh(wallGeometry, wallMaterial);
wall.position.set(100, 50, -100);
scene.add(wall);
collidableMeshList.push(wall);
var wall = new THREE.Mesh(wallGeometry, wireMaterial); // wireframe (not necessary)
wall.position.set(100, 50, -100);
scene.add(wall);
```

Modified from demo: http://stemkoski.github.io/Three.js/Collision-Detection.html
Collision Detection Code

```javascript
var originPoint = movingCube.position.clone();

for (var vi = 0; vi < movingCube.geometry.vertices.length; vi++) {
    var localVertex = movingCube.geometry.vertices[vi].clone();  // get vertex coordinates relative to the object
    var globalVertex = movingCube.localToWorld( localVertex );  // convert to world coordinates
    var directionVector = globalVertex.sub( originPoint );  // vertex - origin (vector subtraction)
}
```

Modified from demo: http://stemkoski.github.io/Three.js/Collision-Detection.html
Collision Detection Code

```javascript
var originPoint = movingCube.position.clone();

for (var vi = 0; vi < movingCube.geometry.vertices.length; vi++) {
    var localVertex = movingCube.geometry.vertices[vi].clone(); // get vertex coordinates relative to the object
    var globalVertex = movingCube.localToWorld( localVertex ); // convert to world coordinates
    var directionVector = globalVertex.sub( originPoint ); // vertex - origin (vector subtraction)

    // cast a ray from the center of the object through the vertex
    var ray = new THREE.Raycaster( originPoint, directionVector.clone().normalize() ); // normalize to unit vector
    var collisionResults = ray.intersectObjects( collidableMeshList );
}
```

Modified from demo: http://stemkoski.github.io/Three.js/Collision-Detection.html
Collision Detection Code

```javascript
var originPoint = movingCube.position.clone();

for (var vi = 0; vi < movingCube.geometry.vertices.length; vi++) {
    var localVertex = movingCube.geometry.vertices[vi].clone(); // get vertex coordinates relative to the object
    var globalVertex = movingCube.localToWorld(localVertex); // convert to world coordinates
    var directionVector = globalVertex.sub(originPoint); // vertex - origin (vector subtraction)

    // cast a ray from the center of the object through the vertex
    var ray = new THREE.Raycaster(originPoint, directionVector.clone().normalize()); // normalize to unit vector
    var collisionResults = ray.intersectObjects(collidableMeshList);

    // if we have at least one collision result, and the collision vector is less than the direction vector, HIT
    // note: collisionResults[0].distance is like our "t" value
    if (collisionResults.length > 0 && collisionResults[0].distance < directionVector.length()) {
        console.log('HIT'); // or do something else like move the object back to where it was, or delete the collided object
    }
}
```

Modified from demo: http://stemkoski.github.io/Three.js/Collision-Detection.html
Final Project
Photo Examples
Final project examples
Final project examples
Final project examples
Final project examples