David Nnaji

January 27, 2025

CSC 322 Introduction to Computer Graphics Spring 2025

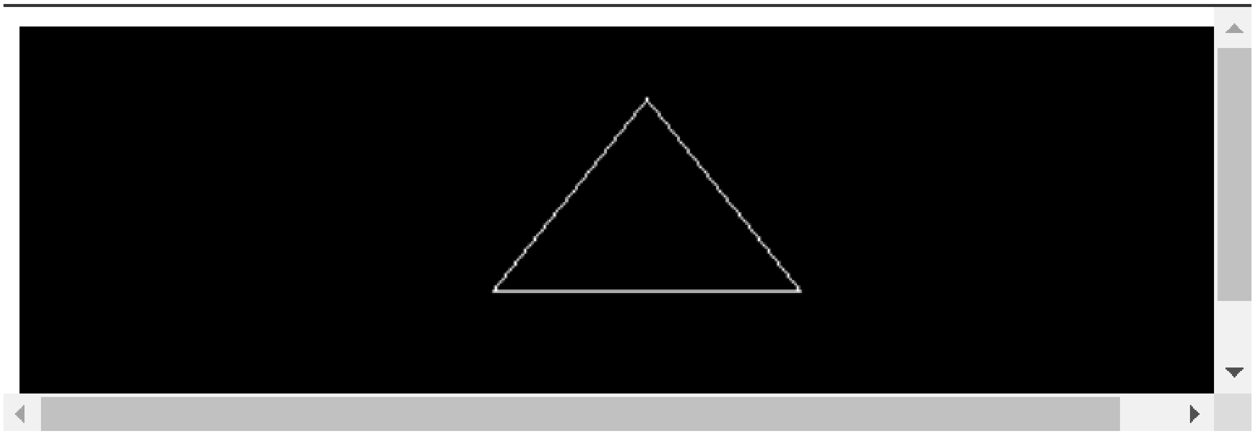
Homework #1

Due Date: 01/29/2025 before class

Submission Link: https://learn.cua.edu

Problem 1:

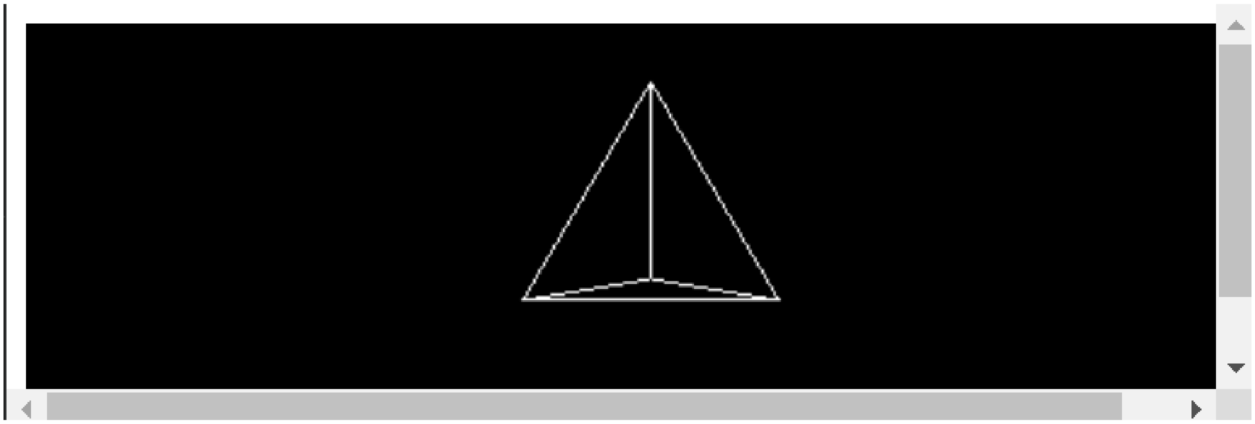
Task #1 -



```
const renderer = new THREE.WebGLRenderer();

renderer.setSize( window.innerWidth, window.innerHeight );

document.body.appendChild( renderer.domElement );


const camera = new THREE.PerspectiveCamera( 45, window.innerWidth /
window.innerHeight, 1, 500 );

camera.position.set( 0, 0, 100 );

camera.lookAt( 0, 0, 0 );


const scene = new THREE.Scene();


//create a blue LineBasicMaterial
```

```
const material = new THREE.LineBasicMaterial( { color: 0xFFFFFF } );

const points = [];
points.push( new THREE.Vector3( - 30, -10, 0 ) ); //left side
points.push( new THREE.Vector3( 0, 30, -10 ) ); //top
points.push( new THREE.Vector3( 30, -10, 0 ) ); //right
points.push(new THREE.Vector3(-30,-10,0)) //bottom

const geometry = new THREE.BufferGeometry().setFromPoints( points );

const line = new THREE.Line( geometry, material );

scene.add( line );
renderer.render( scene, camera );
```

Task #2 -



```
const renderer = new THREE.WebGLRenderer();
renderer.setSize(window.innerWidth, window.innerHeight);
document.body.appendChild(renderer.domElement);

const camera = new THREE.PerspectiveCamera(45, window.innerWidth /
window.innerHeight, 1, 500);
```

```
camera.position.set(0, 0, 100);
camera.lookAt(0, 0, 0);

const scene = new THREE.Scene();

//create a blue LineBasicMaterial
const material = new THREE.LineBasicMaterial({ color: 0xFFFFFF });

const points = [];
// vertices of the tetrahedron
points.push(new THREE.Vector3(0, 30, 0));  // top
points.push(new THREE.Vector3(-20, -10, 20));  // front-left
points.push(new THREE.Vector3(20, -10, 20));  // front-right vertex
points.push(new THREE.Vector3(0, -10, -20));  // back vertex

// edges of the tetrahedron
const edges =
[
   points[0], points[1], // top to front-left
   points[1], points[2], // frontleft to front-right
   points[2], points[0], // frontright to top
   points[0], points[3], // top to back
   points[1], points[3], // frontleft to back
   points[2], points[3], // frontright to back
];

const geometry = new THREE.BufferGeometry().setFromPoints(edges);
const line = new THREE.LineSegments(geometry, material);


scene.add(line);
renderer.render(scene, camera);
```