

Übung 9: Modellierung von Polygonnetzen

1. Warum wird WebGL als Zustandsmaschine bezeichnet?
2. Nach welchem Prinzip arbeitet Three.js?
3. In welcher Reihenfolge müssen die Eckpunkte für die Vorderseite eines Polygons angegeben werden?

Praktischer Teil: Schreiben Sie ein Programm, das eine Pyramide in Parallelprojektion (`THREE.OrthographicCamera(xmin, xmax, ymax, ymin, zmin, zmax)`) darstellt!

Die Pyramide ist ein Objekt der Klasse `geometry` und enthält `vertices` und `faces` (siehe <http://definitelytyped.org/docs/threejs--three/classes/three.geometry.html>)

Zeichnen Sie alle Deckflächen in einer Farbe und variieren Sie die zu zeichnenden Seiten mit dem Parameter `side`!

Rotieren Sie die Szene zur besseren Ansicht, um nicht nur auf eine Fläche zu sehen!

z.B. `scene.rotation.y = 0.5;`

Erzeugen Sie eine Animation, indem Sie den Rotationswinkel regelmäßig verändern und mit `requestAnimationFrame()`; immer wieder ihre Zeichenfunktion aufrufen!

Zur besseren Betrachtung überlagern Sie das Ganze mit einer Kantendarstellung!

Hier können Sie die Methode `THREE.SceneUtils.createMultiMaterialObject()` nutzen;

Sie können sich an den Beispielen aus der Vorlesung orientieren!

Für Motivierte:

Zeichnen Sie zusätzlich die Grundfläche und alle Flächen in unterschiedlichen Farben!

Hierzu müssen Sie den `faces` noch andere Eigenschaften zuordnen (siehe <http://definitelytyped.org/docs/threejs--three/classes/three.face3.html>)

Bei der Materialeinstellung müssen Sie dann statt `colors` `vertexColors` spezifizieren!