BEUTH HOCHSCHULE
FÜR TECHNIK
BERLIN
University of Applied Sciences

Prof. Dr.-Ing. Kristian Hildebrand

Computergrafik 2 / Aufgabe 2.3
Transformationen und Szenengraph

WiSe 2015/2016

In dieser Übung machen Sie sich mit der Transformation von Modellen und Szenen vertraut und modellieren und animieren eine Three.js-Szene hierarchisch.

Abbildung 1: Beispielhafte Ergebnisausgabe

Aufgabe 1: Implementierung eines animierten Roboters

Planen Sie einen Fantasie-Roboter, der mindestens einen Oberkörper, einen Kopf, Arme mit Händen und Beine mit Füßen haben sollte. Der Roboter soll sich mindestens aus Würfeln, Cylindern, Kugeln (für die Gelenke) und parametrischen Flächen Ihrer Wahl zusammensetzen. Lassen Sie Ihrer Kreativität freien Lauf. **Für eine sehr gute Note sollte Ihr Roboter möglichst nicht-trivial und sehr systematisch konstruiert sein.**

Legen Sie für die Implementierung ein neues Modul models/robot.js an, in welchem ein Robotermodell definiert wird.

Aufgabe 1.1: Konstruktionsplanung

Fertigen Sie für die Bearbeitung und Demonstration der Aufgabe einen Konstruktionsplan Ihres Roboters an und planen Sie darin, welches Körperteil seinen Koordinatenursprung wo haben wird. Aus der Zeichnung soll hervorgehen, welche Einzelteile der Roboter besitzt und wie diese Einzelteile relativ zu Ihren Abmessungen zueinander transformiert sind (Skelett).

Benennen Sie zunächst die Einzelteile des Roboters und geben Sie den wichtigen Größen symbolische Bezeichner (z.B. OAL = Oberarm-Länge). Später in Ihrem Code sollen sich bei den Transformationen im

Skelett möglichst wenige absolute Zahlenwerte, sondern Formeln unter Verwendung dieser Bezeichner wiederfinden (z.B. "transformiere um die halbe Oberarmlänge", nicht "transformiere um 2.374"). **Wird im Code der Bezeichner auf einen anderen Wert gesetzt, soll das Modell sich entsprechend konsistent anpassen.**

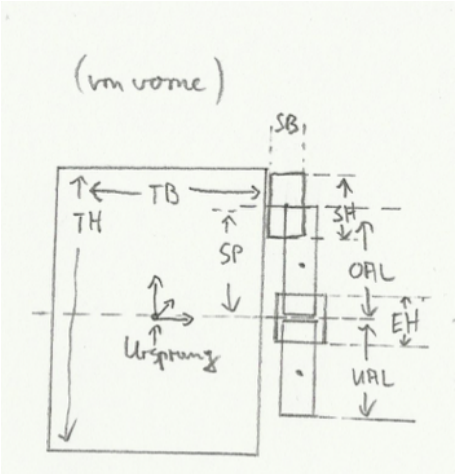

Abbildung 2: Konstruktionsplan

Bitte beachten: eine elektronische Kopie dieses Plans gehört in Ihre .zip-Datei zur formalen Abgabe via Moodle. Anhand des Plans sollen Sie bei der Abnahme den Aufbau des Roboters / Szenengraphen erläutern.

Aufgabe 1.2: Hierarchischer Szenenaufbau mit Skelett und Haut

Fügen Sie in Ihrer Szene einen neuen Button "Create Robotïn *index.html* hinzu, der den Roboter erzeugt und in der Szene zeichnet. Setzen Sie Ihren Konstruktionsplan im Roboter-Modul in Code um. Erstellen Sie mit Hilfe von den Three.js Modulen *THREE.Object3D* und *THREE.Mesh* Skelett und Haut des Modells getrennt, so dass z.B. Skalierungen der Haut sich nicht auf das Skelett und die übrigen Haut-Teile auswirken.

Sie können alle Knoten in beliebiger Reihenfolge erzeugen und dann einen Knoten einem anderen mittels der Three.js *add()* Funktion als Kind zuweisen (z.B. um das Skelett zu konstruieren, oder um die Skins an das Skelett zu hängen).

Bringen Sie selbstständig in Erfahrung (über die Three.js Dokumentation) wie Sie die Transformation jedes Knotens manipulieren können. Diese Transformation sollten in einem Szenengraphen immer relativ zum Elternknoten erfolgen.

Folgender Beispielcode erzeugt einen Roboter aus zwei Quadern (Torso und Kopf) mit Skelett und Haut:

```
1
2   var headSize  = [130,130,130];
3   var torsoSize = [250,400,150];
4
5   this.root = new THREE.Object3D();
6   // skeleton
7   this.head = new THREE.Object3D();
8   this.head.name = "head";
9   this.head.translateY(torsoSize[1]/2 + headSize[1]/2);
10  this.torso = new THREE.Object3D();
11
12  this.torso.add(this.head);
13
14  // skin
15  this.headSkin = new THREE.Mesh( new THREE.CubeGeometry( headSize[0], headSize[1],
        headSize[2]), new THREE.MeshNormalMaterial() );
16  this.headSkin.rotateY( Math.PI/4 );
17
```

```
18   this.torsoSkin = new THREE.Mesh( new THREE.CubeGeometry( torsoSize[0], torsoSize
       [1], torsoSize[2] ), new THREE.MeshNormalMaterial() );
19   this.torso.add(this.torsoSkin);
20   this.head.add(this.headSkin);
21
22   this.root.add(this.torso);
```

Beachten Sie, dass der *this.root*-Knoten (Object3D ohne Transformationen) zu unserer Szene hinzugefügt werden muss. Zum Beispiel so:

```
1
2 var robot = new Robot();
3 scene.addMesh(robot.getMesh());
```

Aufgabe 1.3: Bewegung der Gelenke

Wie Sie schon bemerkt haben, kann die gesamte Szene mittels der Pfeil-Tasten rotiert werden. In der Szene wir bei jedem Tastendruck der *onDocumentKeyDown*-Callback aufgerufen und dabei das aktuelle Mesh rotiert.

Erweitern Sie den Code in *scene.js* für alle Gelenke Ihres Roboters. **Hinweis:** Der folgende Beispielcode sucht aus dem Three.js Szenengraphen das Objekt mit dem Namen 'head' und rotiert dieses um die x-Achse um PI.

```
1 var nodeHead = scope.scene.getObjectByName("head", true);
2 if( nodeHead ) {
3   nodeHead.rotateX(Math.PI);
4 }
```

Aufgabe 1.4: Animation (optional, nur für eine sehr gute Note)

Bauen Sie eine automatische ablaufende Animation ein, die Teile des Roboters bewegt. Kombinieren Sie dabei unterschiedliche Transformationen: Rotationen, Skalierungen und Translationen.

Abgabe Dies ist *Teilaufgabe 2.3*; die Bearbeitungszeit der Teilaufgabe ist für ca. eine Woche ausgelegt. Die Abgabe der gesamten Aufgabe 2 soll via Moodle bis zu dem dort angegebenen Termin erfolgen. Verspätete Abgaben werden mit einem Abschlag von 2/3-Note je angefangener Woche Verspätung belegt. Geben Sie bitte pro Gruppe jeweils nur eine einzige .zip-Datei mit den Quellen Ihrer Lösung ab.

Demonstrieren und erläutern Sie Ihre Lösung in der nächsten Übung nach dem Abgabetag. Die Qualität Ihrer Demonstration ist, neben dem abgegebenen Code, ausschlaggebend für die Bewertung! Es wird erwartet, dass alle Mitglieder einer Gruppe anwesend sind und Fragen beantworten können.