

Computação Gráfica

2023/2024 – 4º Período

LEIC-A, LEIC-T

Técnico Lisboa

Labs 1 & 2 - Sumário

informações gerais sobre o funcionamento

formação de grupos

prototipagem de baixa-fidelidade

introdução ao Three.js – Hello 3D World

apoio à execução dos trabalhos

informações gerais
sobre o funcionamento

informações gerais

Breve apresentação

Daniel Simões Lopes

email: daniel.s.lopes(at)tecnico.ulisboa.pt

página: web.tecnico.ulisboa.pt/daniel.s.lopes



informações gerais

Breve apresentação

Filipa Cotrim

email: filipa.cotrim(at)tecnico.ulisboa.pt



informações gerais

Breve apresentação

Inês Lúcio

email: ines.marques.lucio(at)tecnico.ulisboa.pt



informações gerais

Breve apresentação

Ivo Roupa

email: ivo.roupa@tecnico.ulisboa.pt



informações gerais

Breve apresentação

João Moreira

email: joaopedro.moreira(at)inesc-id.pt



informações gerais

Breve apresentação

Tomás Alves

email: tomas.alves(at)tecnico.ulisboa.pt

página: web.tecnico.ulisboa.pt/~tomas.alves



informações gerais

Breve apresentação

Anderson Maciel

email: anderson.maciel(at)tecnico.ulisboa.pt

página: web.tecnico.ulisboa.pt/anderson.maciel/



informações gerais

Breve apresentação

Francisco Fernandes

email: francisco.fernandes@tecnico.ulisboa.pt



informações gerais

Breve apresentação

Pedro Belchior

email: pedro.belchior(at)tecnico.ulisboa.pt



informações gerais

Horário de Dúvidas - Alameda

Professor Daniel Simões Lopes

6as-Feiras das 11:00 às 18:00, [Sala Zoom](#) (Pass: uFx4M0)

Assistente Convidada Filipa Cotrim

6as-Feiras das 09:00 às 10:30, Sala 0.09, Pav. Informática III

Assistente Convidada Inês Lúcio

5as-Feiras das 10:30 às 12:00, [Sala Zoom](#) (Pass: 977359)

Assistente Convidado Ivo Roupa

2as-Feiras das 14:00 às 15:30, Sala 0.09, Pav. Informática III

Assistente Convidado João Moreira

3as-Feiras das 16:30 às 18:00, Sala 532, INESC-ID

4as-Feiras das 13:00 às 14:30, Sala 532, INESC-ID

Assistente Convidado Tomás Alves

2as-Feiras das 14:00 às 15:30, Sala 532, INESC-ID

3as-Feiras das 09:30 às 11:00, Sala 532, INESC-ID

informações gerais

Horário de Dúvidas - Taguspark

Professor Anderson Maciel

4as-Feiras das 11:30 às 12:30

5as-Feiras das 10:00 às 11:30

5as-Feiras das 15:50 às 17:00, 2-N9.21 ou [Zoom](#)

Assistente Convidado Francisco Fernandes

3as-Feiras das 14:00 às 15:30, Sala 533, INESC-ID ou [Sala Zoom](#)

Assistente Convidado Pedro Belchior

4as-Feiras das 10:00 às 11:30

5as-Feiras das 09:30 às 11:00, [Sala Zoom](#)

informações gerais

Horário de Dúvidas

Regra 1: as sessões poderão ser feitas por Zoom ou presencialmente

Regra 2: enviar e-mail informando o interesse na sessão de dúvidas com uma antecedência de 24h

Regra 3: se ninguém comparecer após os primeiros 15 minutos a sessão fica sem efeito.

informações gerais

Consultar página do Fénix Componente Prática

LEIC-A

<https://fenix.tecnico.ulisboa.pt/disciplinas/CGrA23/2023-2024/2-semestre>

LEIC-T

<https://fenix.tecnico.ulisboa.pt/disciplinas/CGrA3/2023-2024/2-semestre>

informações gerais

Os trabalhos são discutidos no turno em que o grupo está inscrito.

Cada trabalho de laboratório é entregue na mesma data e hora para todos os alunos às 23:59h da Sexta-Feira que antecede a semana da avaliação (o código será avaliado fora do período de aulas mas nas semanas das avaliações há discussão com cada grupo).

informações gerais

As notas de laboratório obtidas no ano anterior podem ser aproveitadas no ano corrente, desde que o aluno não se inscreva em nenhum turno de laboratório.

Caso o aluno se inscreva num turno do laboratório, as notas do ano anterior serão anuladas

informações gerais

mapa das aulas laboratoriais

week (semester)	week (course)	course interruptions	calendar	school calendar	lab evaluations	scores	deadlines	lab class topics
9	0	-	08 Apr - 12 Apr	no classes				All Assignments Made Available
10	1		15 Apr - 19 Apr	classes				Lab Structure, Lab Rules, and Lab Assessments Group matching Intro do Three.js - Hello 3D World! Project Support
11	2	Public Holiday on Thursday	22 Apr - 26 Apr	classes	Evaluation of Assignment A	2	during the week	Project Support Project Discussions
12	3	Public Holiday on Wednesday	29 Apr - 03 May	classes				Project Support
13	4		06 May - 10 May	classes			10 May, 23h59	Project Support
14	5		13 May - 17 May	classes	Evaluation of Assignment B	4		Project Support Project Discussions
15	6	Dia do Técnico on Thursday	20 May - 24 May	classes			24 May, 23h59	Project Support
16	7	Public Holiday on Thursday	27 May - 31 May	classes	Evaluation of Assignment C	4		Project Support Project Discussions

informações gerais

abordagem pedagógica das aulas

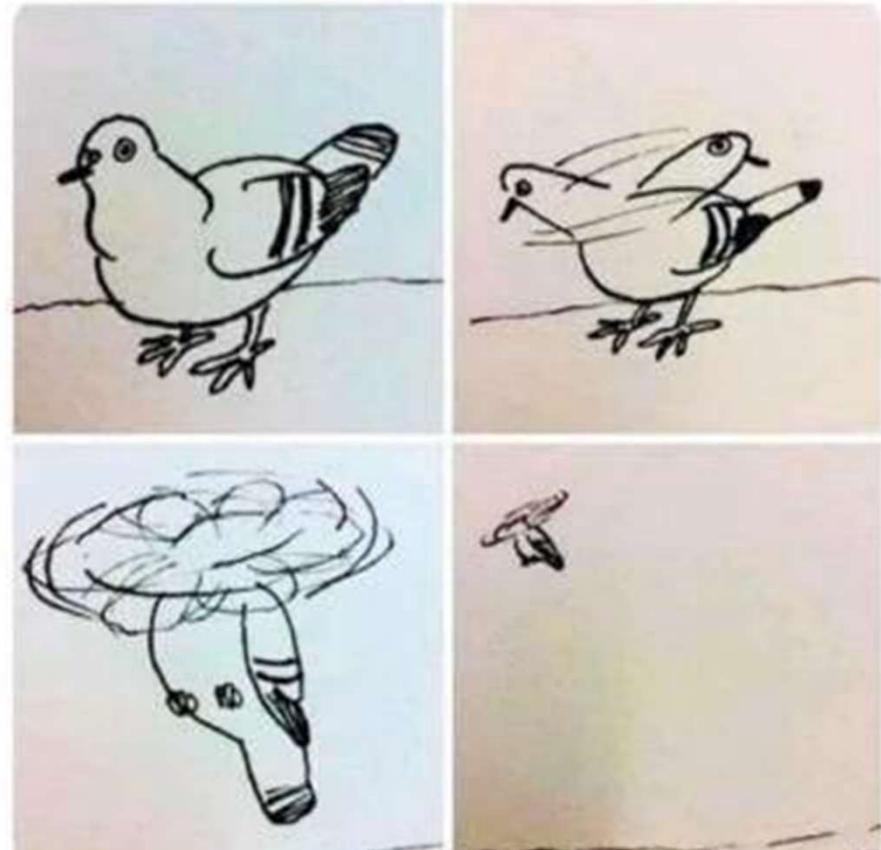
CG teóricas → *well-defined assignments*

CG labs → *open-ended project requirements*

informações gerais

“lá funcionar funciona, mas ...”

When your program
is a complete mess,
but it does its job



informações gerais

“learning strategies”

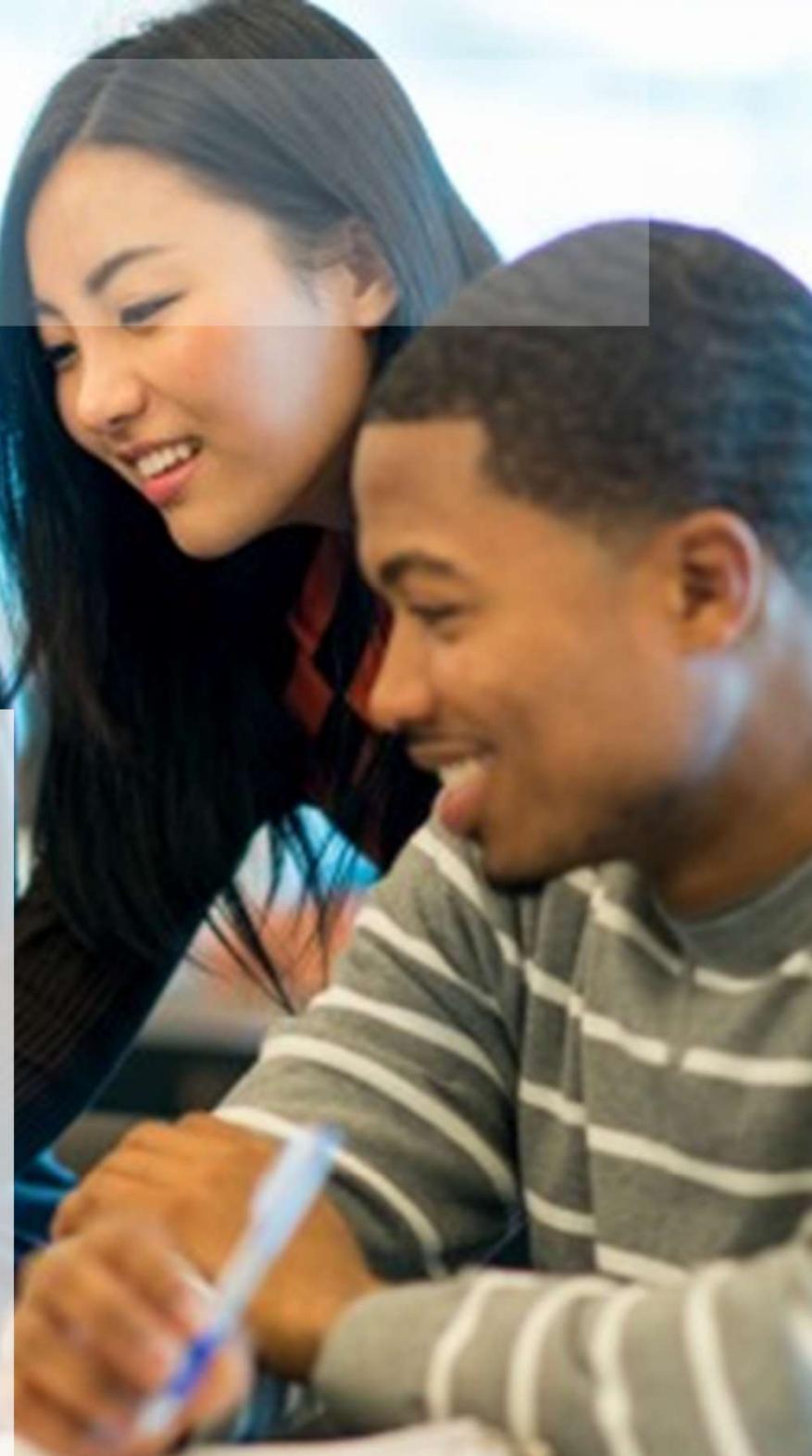
distribuem tarefas entre todos membros do grupo
(o mais equitativamente possível)

comecem a trabalhar a partir do “Dia Zero”

todos os dias existem oportunidades para progredir

compareçam nas aulas (labs + teóricas)

contribuam para um bom ambiente de trabalho
(socializem, colaborem, respeitem-se mutuamente)



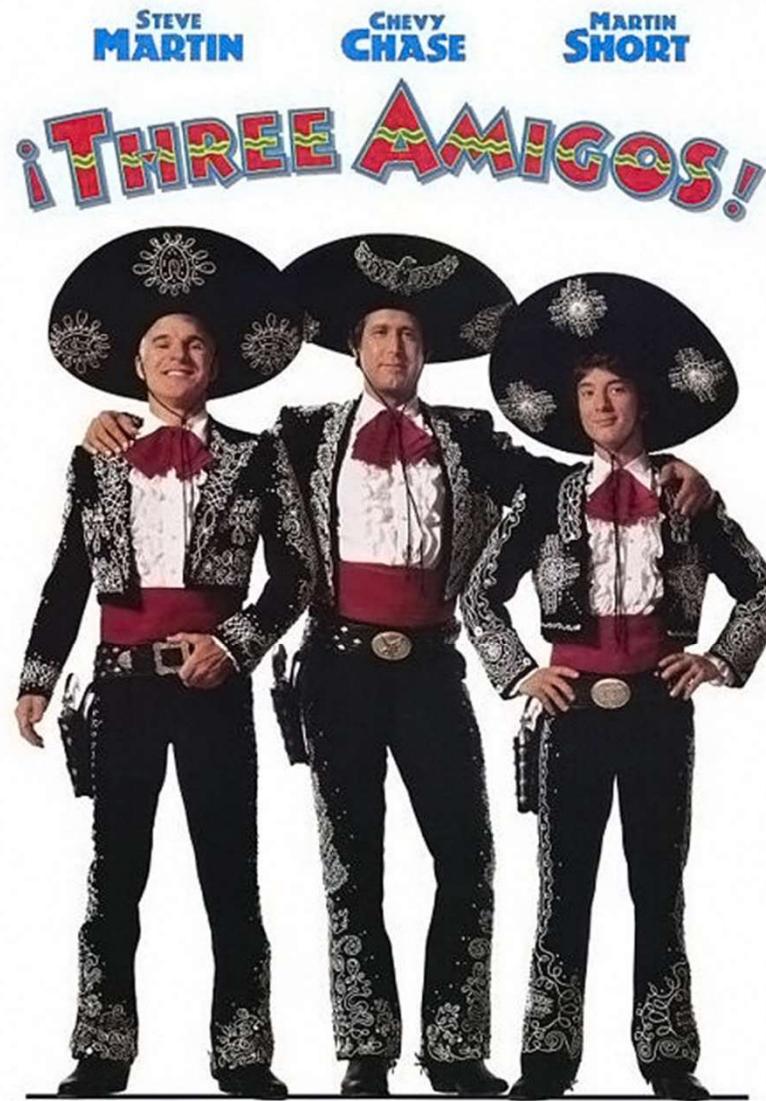
informações gerais

“not a learning strategy”



formação de grupos

formação de grupos



An L.A. FILMS/LORNE MICHAELS Production A LANDIS/FOLSEY Film ¡THREE AMIGOS!
ALFONSO ARAU TONY PLANA PATRICE MARTINEZ Songs by RANDY NEWMAN Score by ELMER BERNSTEIN
Executive Producer STEVE MARTIN Written by STEVE MARTIN • LORNE MICHAELS • RANDY NEWMAN
Produced by LORNE MICHAELS and GEORGE FOLSEY, JR. Directed by JOHN LANDIS

Regras

grupos de **3 membros**

todos os 3 membros têm de pertencer ao **mesmo turno**
(e campus)

grupos definidos na
1^a semana, portanto ...

... vamos definir grupos
AGORA mesmo!

prototipagem de baixa-fidelidade

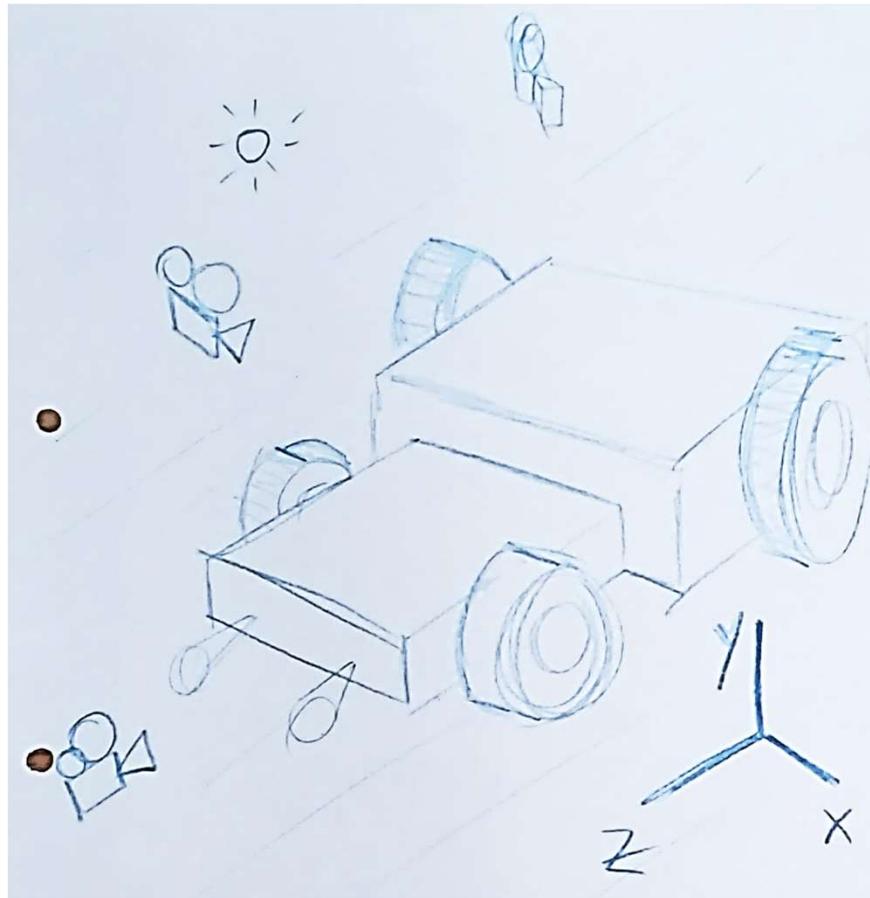
“*sketches*” & *desenho técnico simples*

grafo de cena

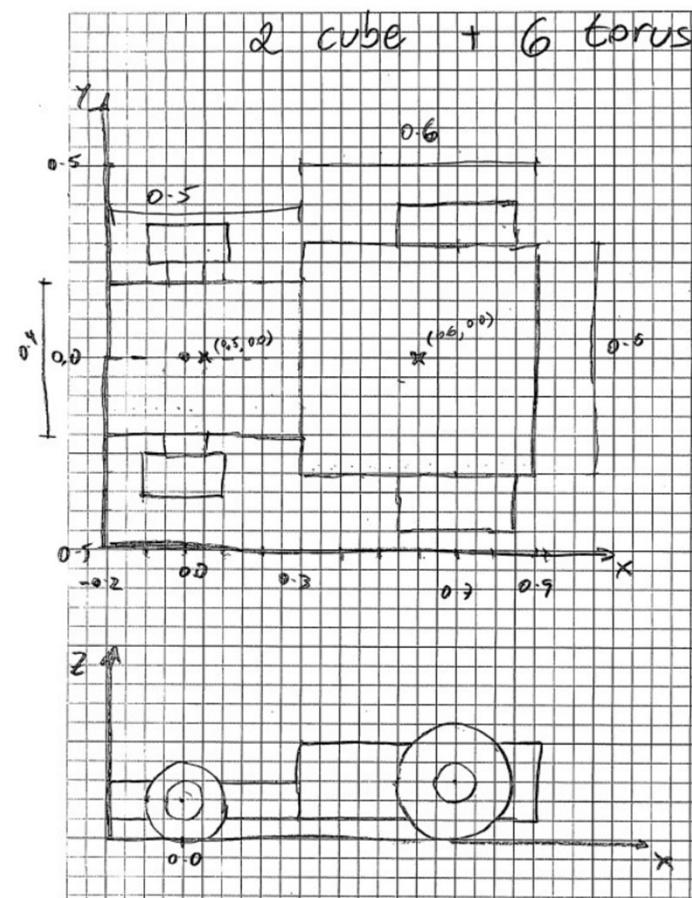
three.js scene graph

“sketches” & des. téc. simples

prototipagem de baixa-fidelidade



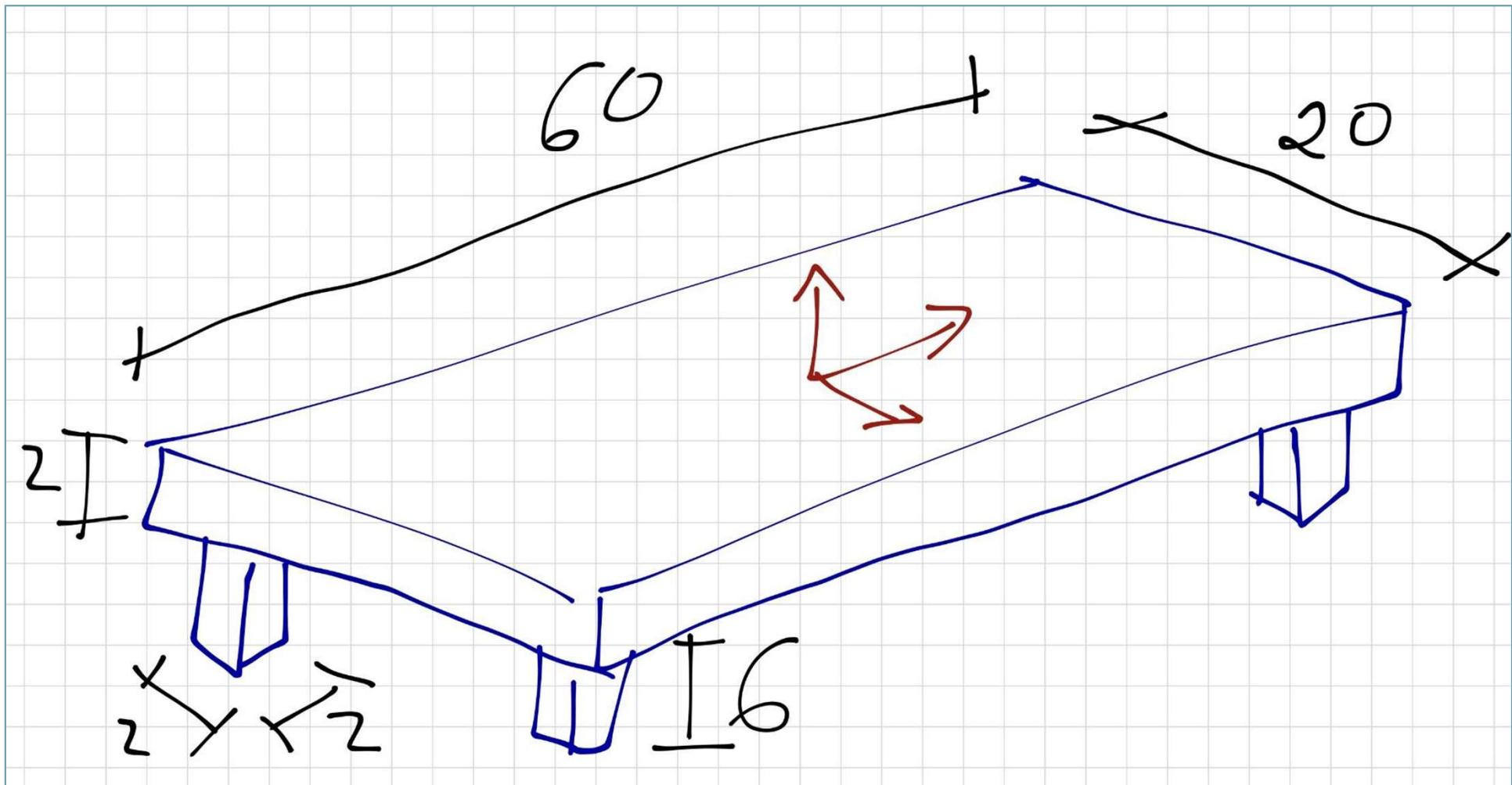
“sketch” de uma
cena simples



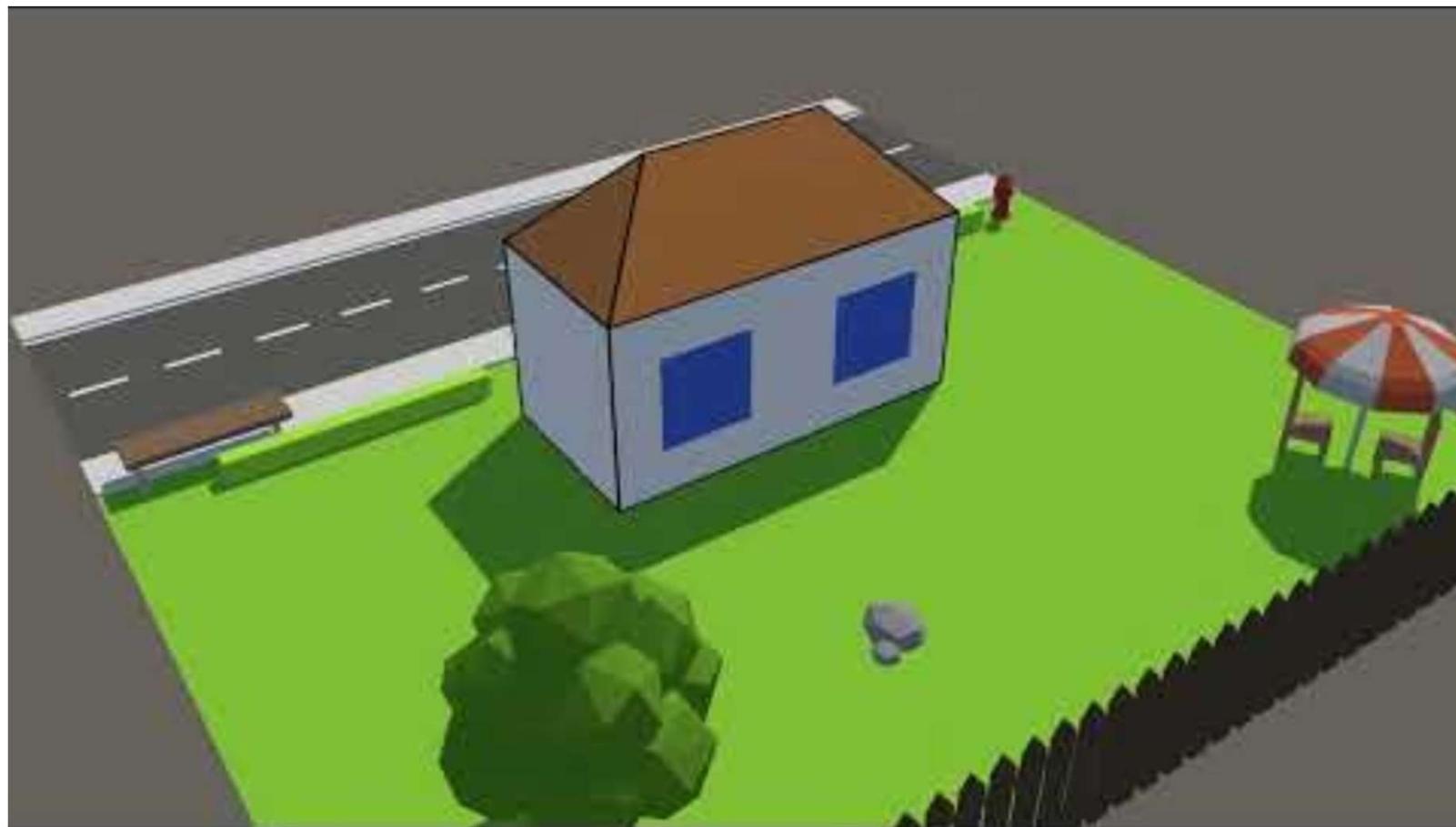
vistas com dimensões
(desenho técnico
simples)

“sketches” & des. téc. simples

geometria da mesa



“sketches” & des. téc. simples
exercício simples (~3 min.)



“sketches” & des. téc. simples

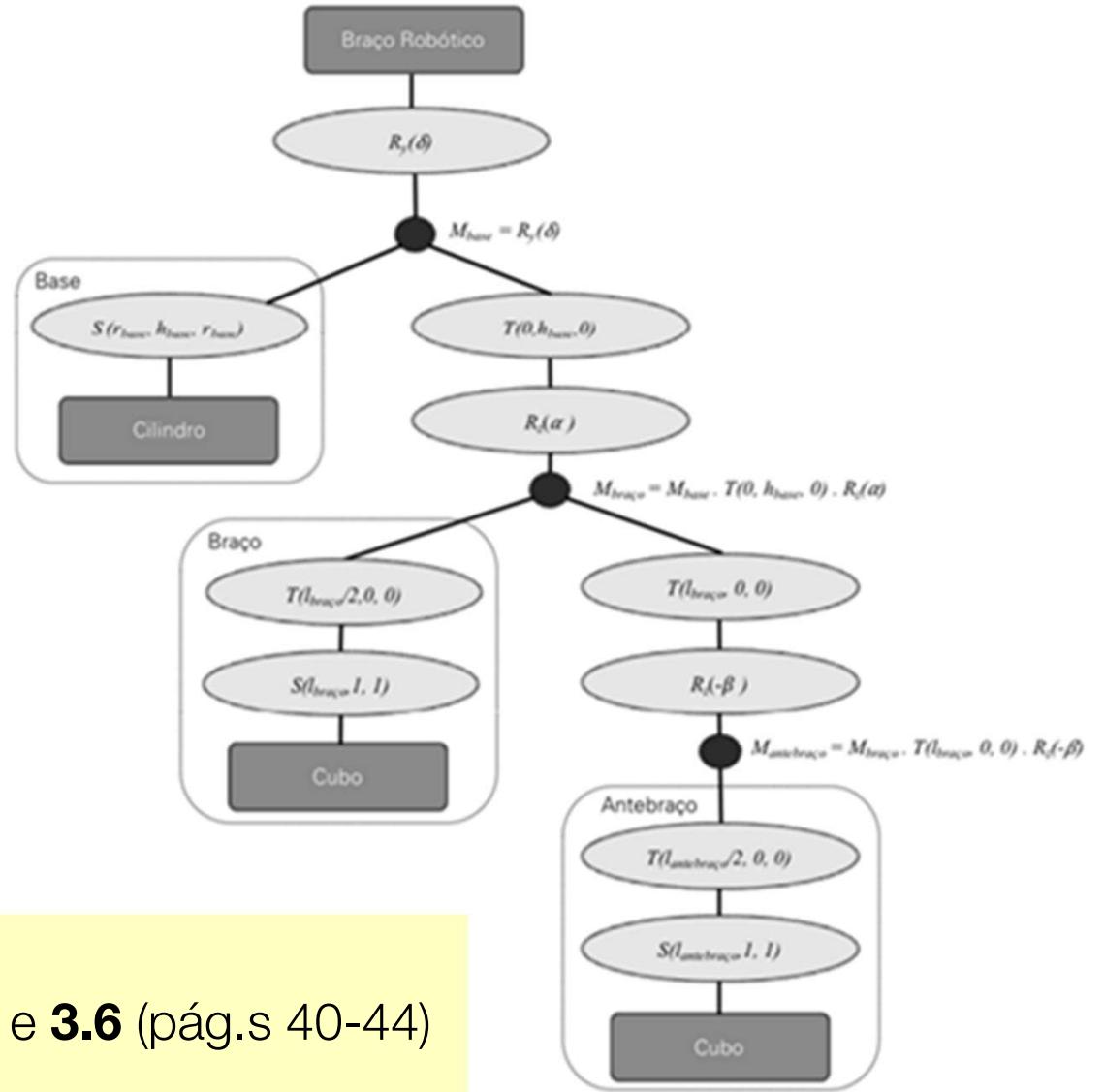
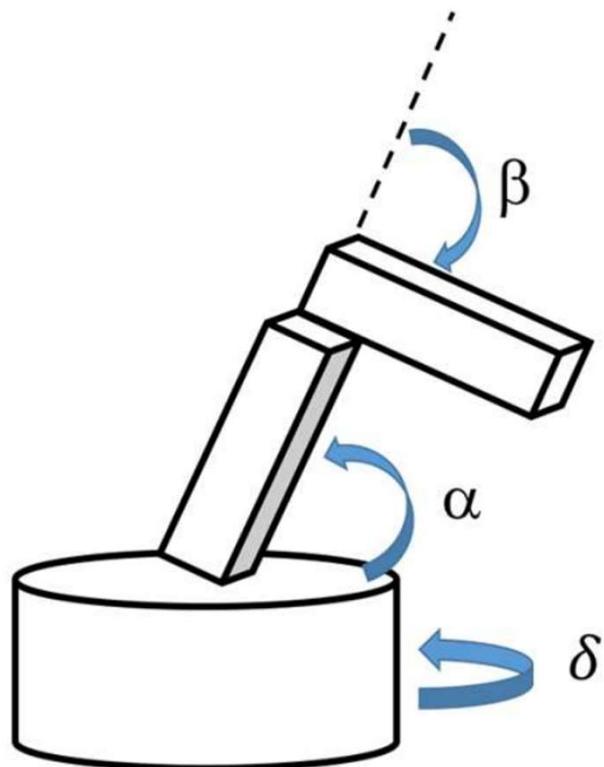
exercício simples (~3 min.)

Quantas e quais as vistas necessárias para reconstruir esta casa (incluindo telhado, porta e janelas)?

Quais as dimensões a indicar?

grafo de cena

diagrama sobre relações de hierarquia



Literatura Recomendada:
sub-secções 2.4 (pág.s 14-15) e **3.6** (pág.s 40-44)
do livro oficial da cadeira

grafo de cena

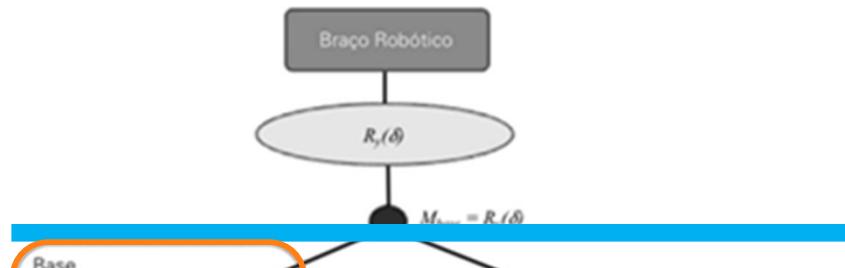
elementos constituintes

Primeiro, listar e definir os elementos constituintes do grafo de cena:

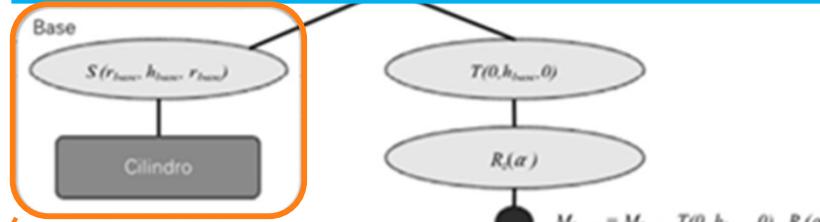
- onde fica o referência global da cena?
 - **WCS**
- qual a configuração final do objecto?
 - **raíz do grafo de cena**
- quantos referenciais locais precisamos?
 - **ref. pai, ref(s) filho(s), ref(s) neto(s), ...**
- quantos Graus de Liberdade (*DOF – Degrees of Freedom*) é que precisamos para descrever o movimento do objecto?
 - **translações em X, Y, Z e/ou rotações segundo X, Y, Z, eixo arbitrário**
- quantas primitivas geométricas precisamos para modelar o objecto?
 - **folhas do grafo de cena**

grafo de cena andares do grafo

Referencial Global (WCS)



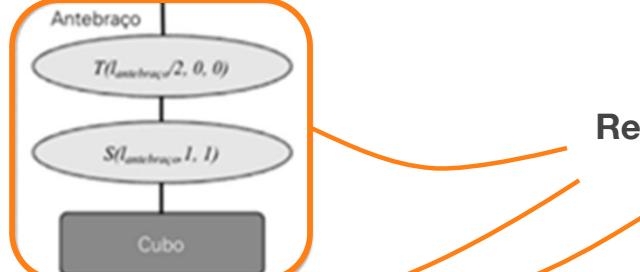
Referencial Local do Pai
(pode coincidir ou não com o WCS)



Referencial(ais) Local(ais) do(s) Filho(s)



Referencial(ais) Local(ais) do(s) Neto(s)



Referencial(ais) Local(ais) da(s) Primitiva(s)
(transformações de instanciação)

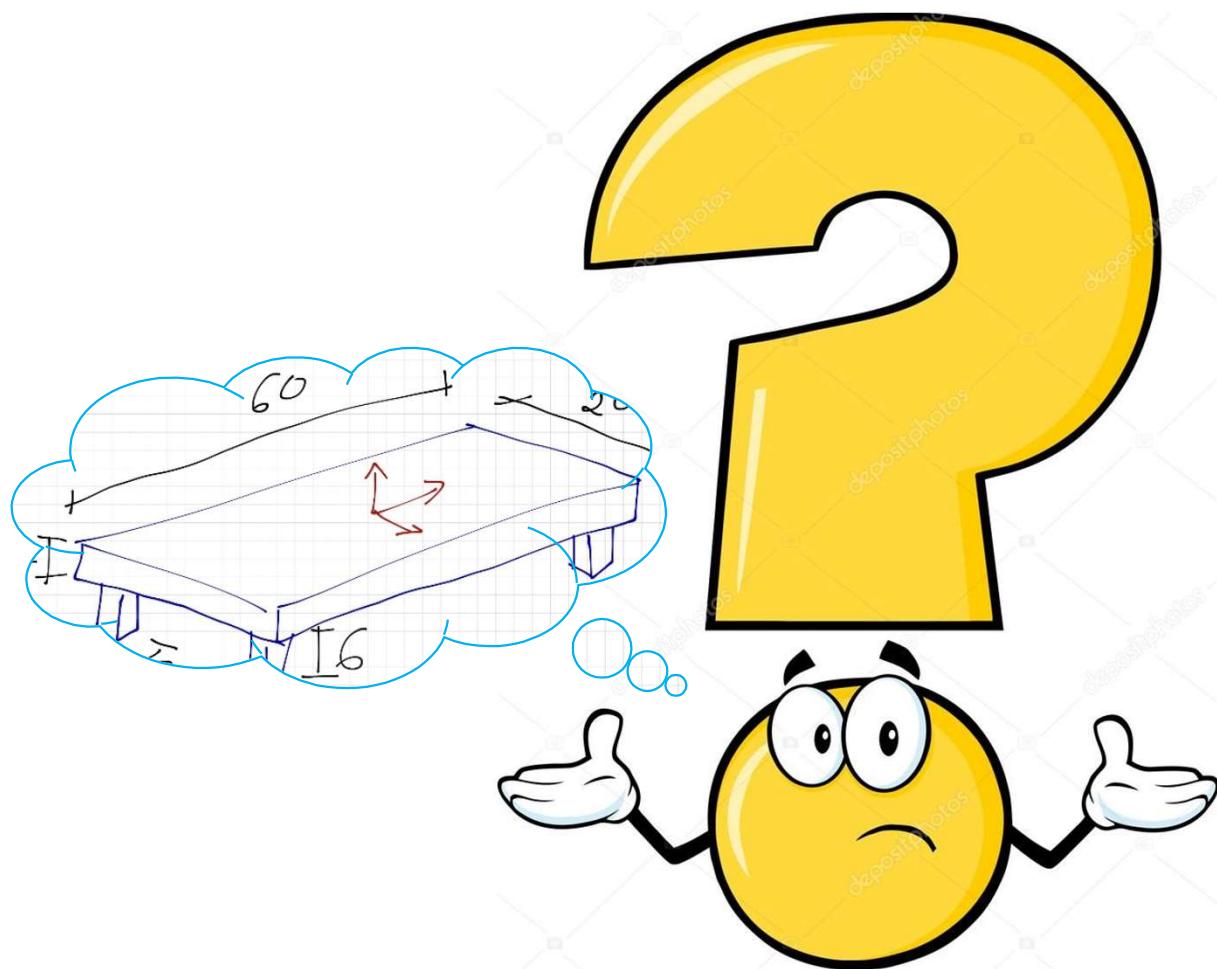
grafo de cena

múltiplas soluções

- **Nota:** Existe mais do que um grafo possível para representar o mesmo objecto na mesma configuração espacial (e.g., figura articulada).
 - **Assim, deve-se sempre projectar um grafo de cena que seja o menos complexo possível, tendo um número mínimo (i.e. óptimo) de folhas, de nós e de transformações geométricas (i.e., um grafo sem redundâncias).**

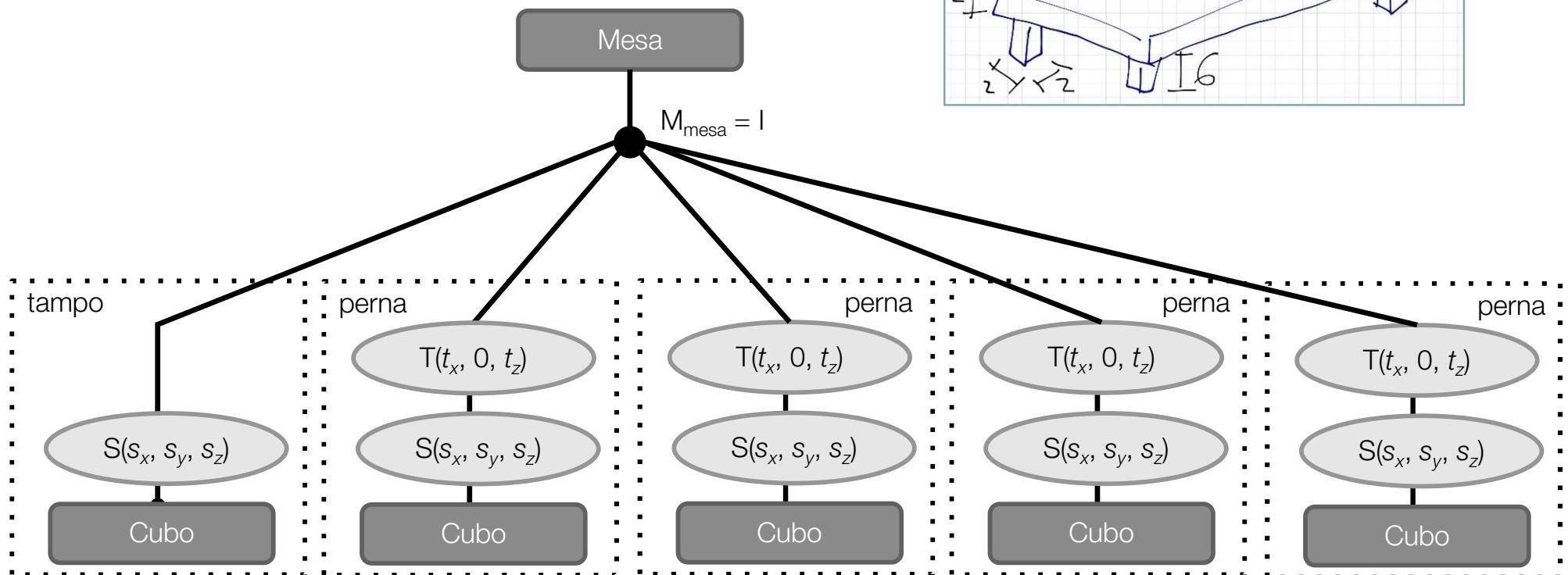
grafo de cena

diagrama sobre relações de hierarquia



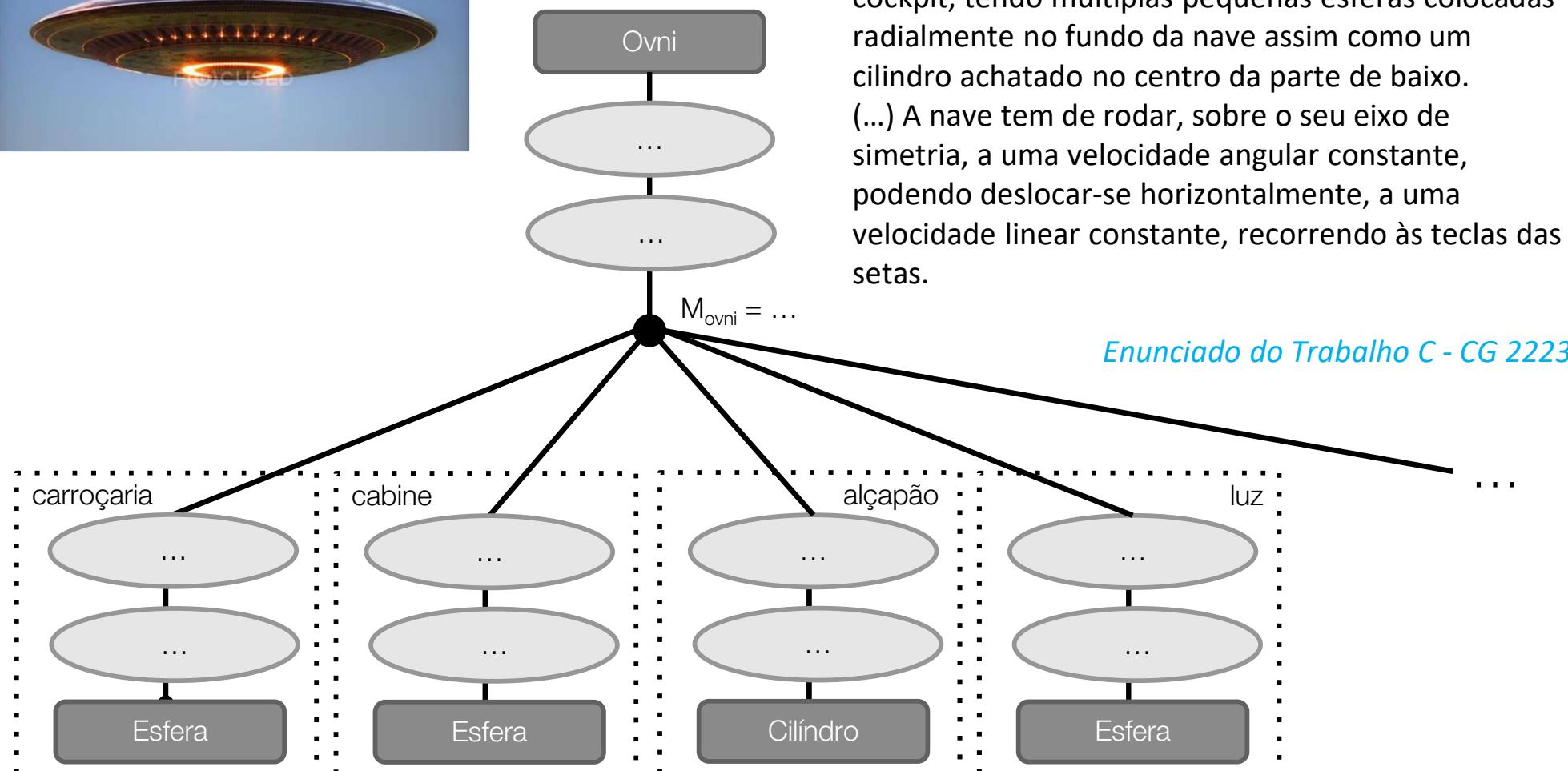
grafo de cena

diagrama sobre relações de hierarquia



grafo de cena

diagrama sobre relações de hierarquia



grafo de cena

diagrama sobre relações de hierarquia

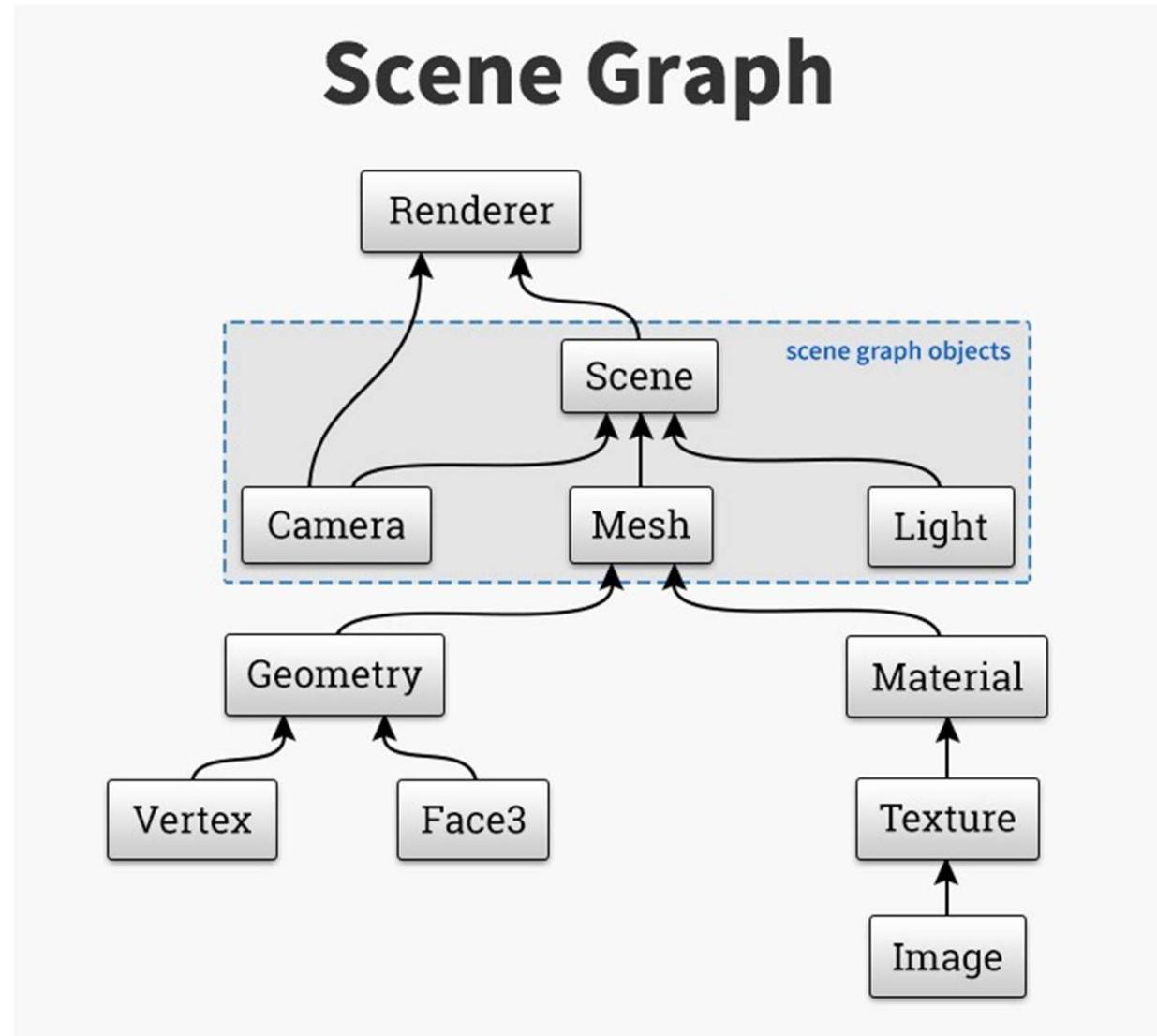


(...) Sendo um objecto articulável, é necessário definir uma hierarquia de transformações entre as diferentes peças que compõem o objecto, isto, por forma a converter o robô para camião e vice-versa. As peças são as seguintes: (i) cabeça que inclui detalhes para olhos e antenas, (ii) braços com dois tubos de escape e antebraços, (iii) tronco, (iv) abdómen, (v) cintura com duas rodas, (vi) coxas, (vii) pernas com quatro rodas, e (viii) pés. É necessário modelar as peças recorrendo a pelo menos uma das seguintes primitivas geométricas disponíveis na biblioteca Three.js,: cilindros, cubos e cones.

(...) Permitir ao utilizador realizar a conversão entre robô-camião, isto é, permitir movimentar os graus de liberdade representados na Figura 1 (C) (vide setas amarelas) recorrendo às seguintes teclas:

- ‘Q(q)’ e ‘A(a)’ para controlar o ângulo θ_1 que roda o eixo de revolução dos pés;
- ‘W(w)’ e ‘S(s)’ para controlar o ângulo θ_2 que roda o eixo de revolução da cintura;
- ‘E(e)’ e ‘D(d)’ para controlar o deslocamento δ_1 que translada os membros superiores medial e lateralmente;
- ‘R(r)’ e ‘F(f)’ para controlar o ângulo θ_3 que roda o eixo de revolução da cabeça.

three.js scene graph



introdução ao Three.js

informações gerais

web app “Hello 3D World”

informações gerais

Three.js

Biblioteca JavaScript

API Gráfica

informações gerais

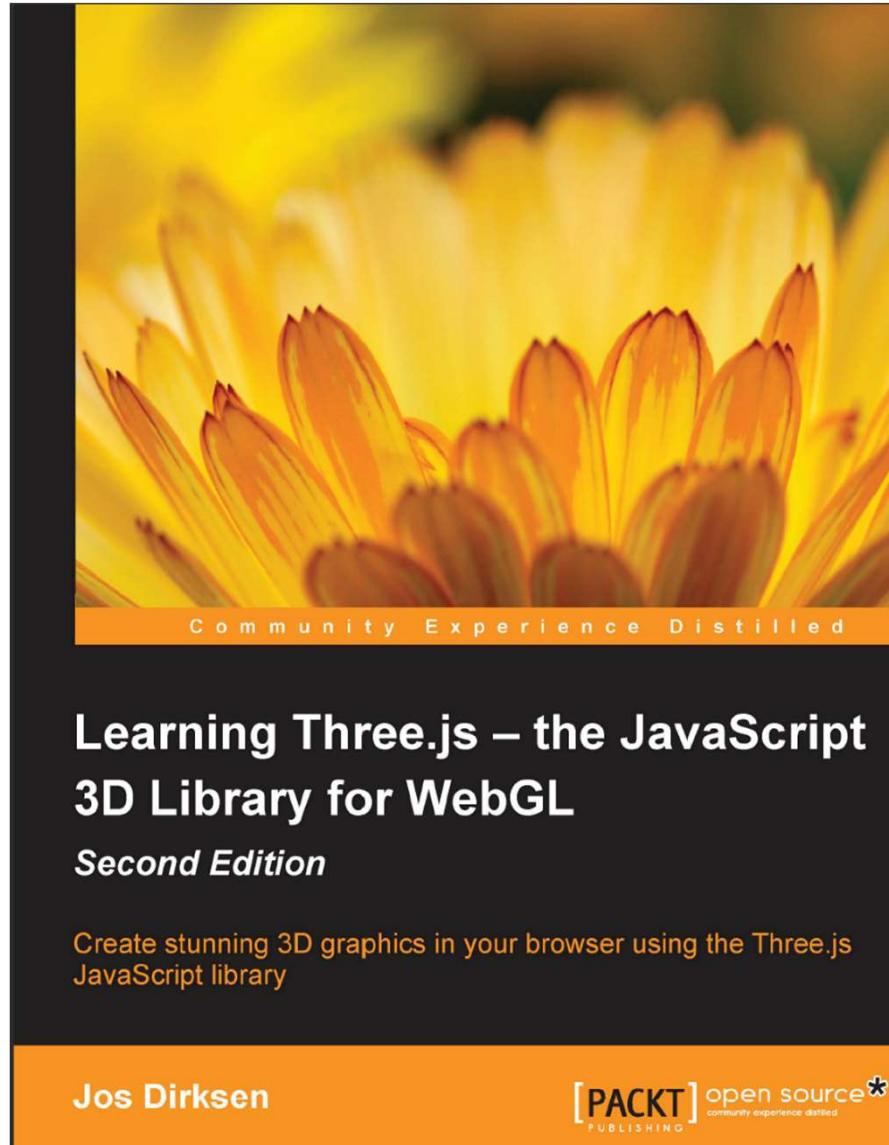
página oficial

A collage of 20 images showcasing various 3D projects and applications built with three.js, including NASA Mars 2020, Spline, Medal of Honor, Weedense Enteret, Voice of Racism, and many more.

threejs.org
threejs.org/n

introdução ao Three.js

livro recomendado



introdução ao Three.js

browsers e formatos



introdução ao Three.js

browsers e formatos



introdução ao Three.js

editor(es)

Visual Studio Code
(OS X, Windows, Linux)

a opção recomendada!
(local server “Live Server” integrado)

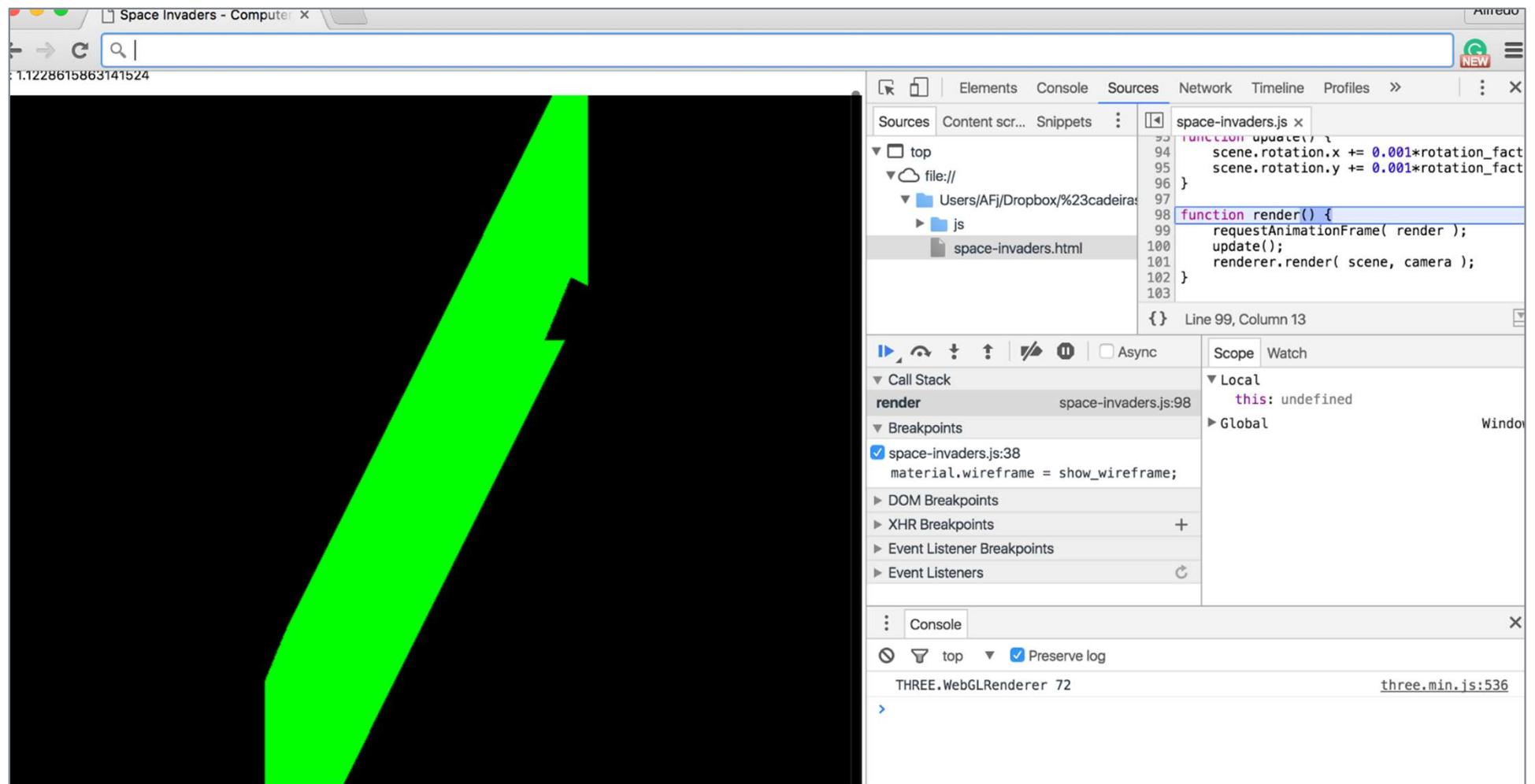
Sublime Text Editor
(OS X, Windows, Linux)

Atom
(OS X, Windows, Linux)

... ou outro da vossa preferência

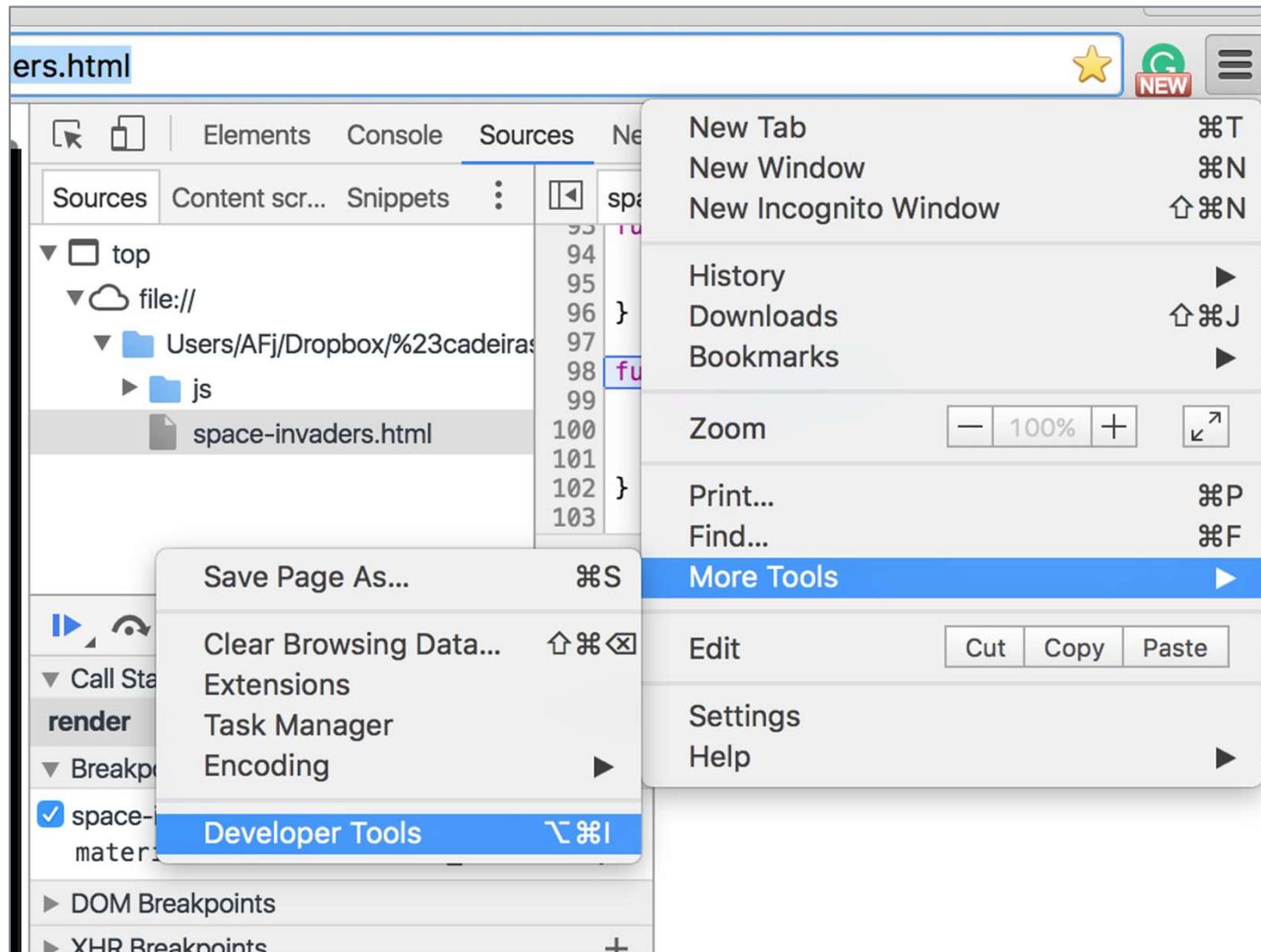
introdução ao Three.js

debugging



introdução ao Three.js

debugging (em Chrome)



introdução ao Three.js

obter o Three.js

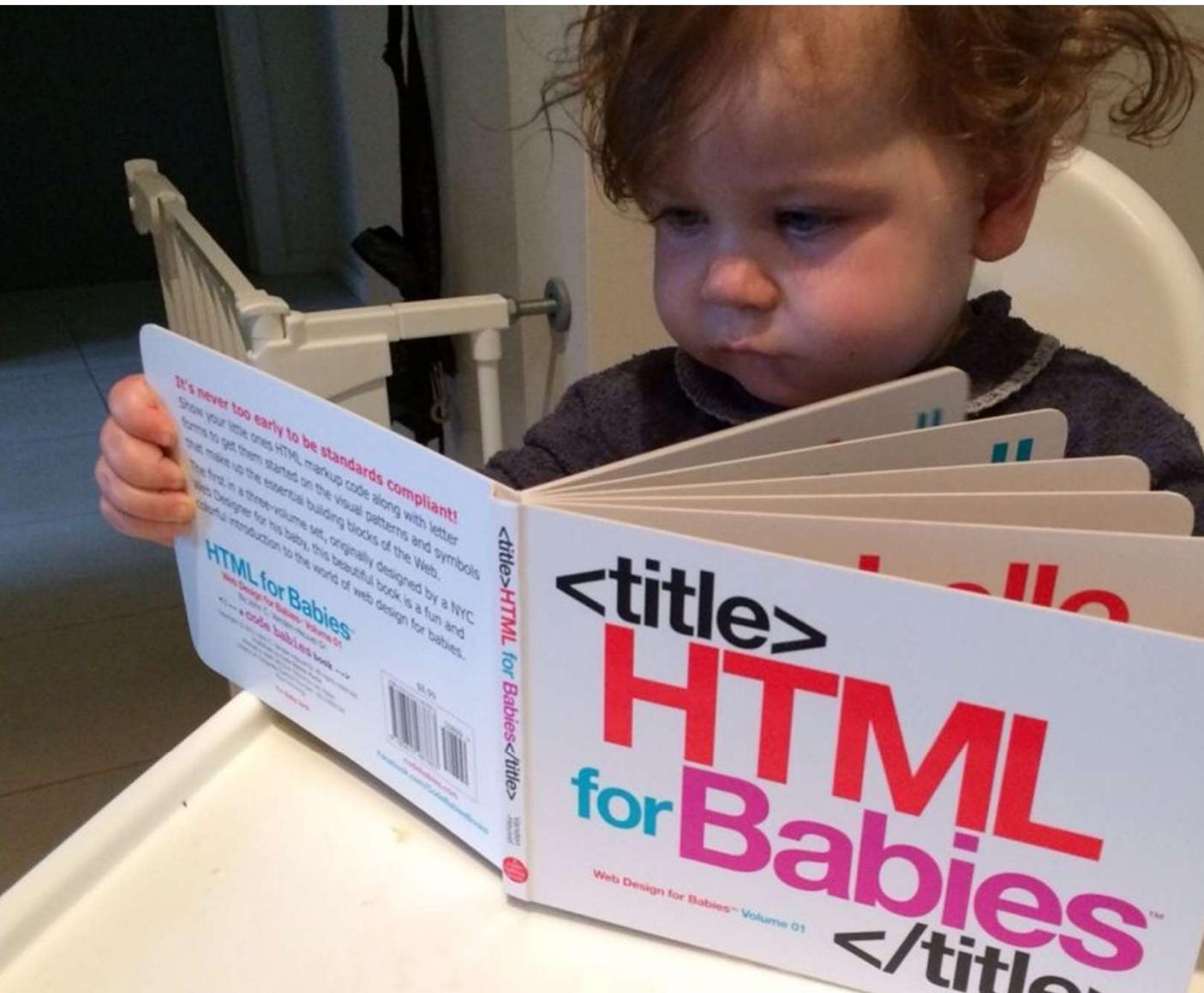
descarregar **versão completa** da página oficial

<http://threejs.org/>

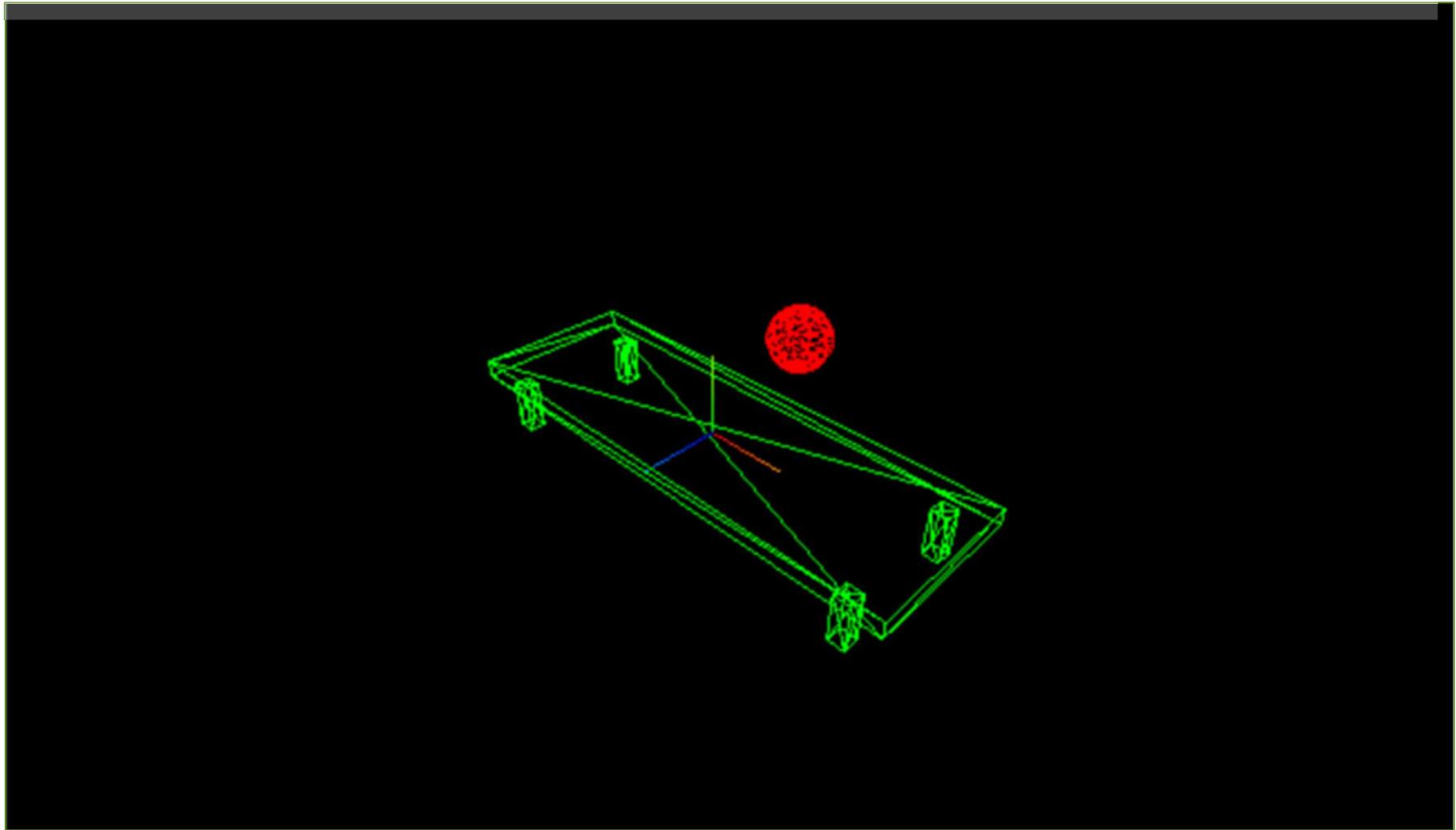
(não recomendado, a menos que estejam a pensar trabalhar offline)

introdução ao Three.js

“HTML + CSS ... for babies”

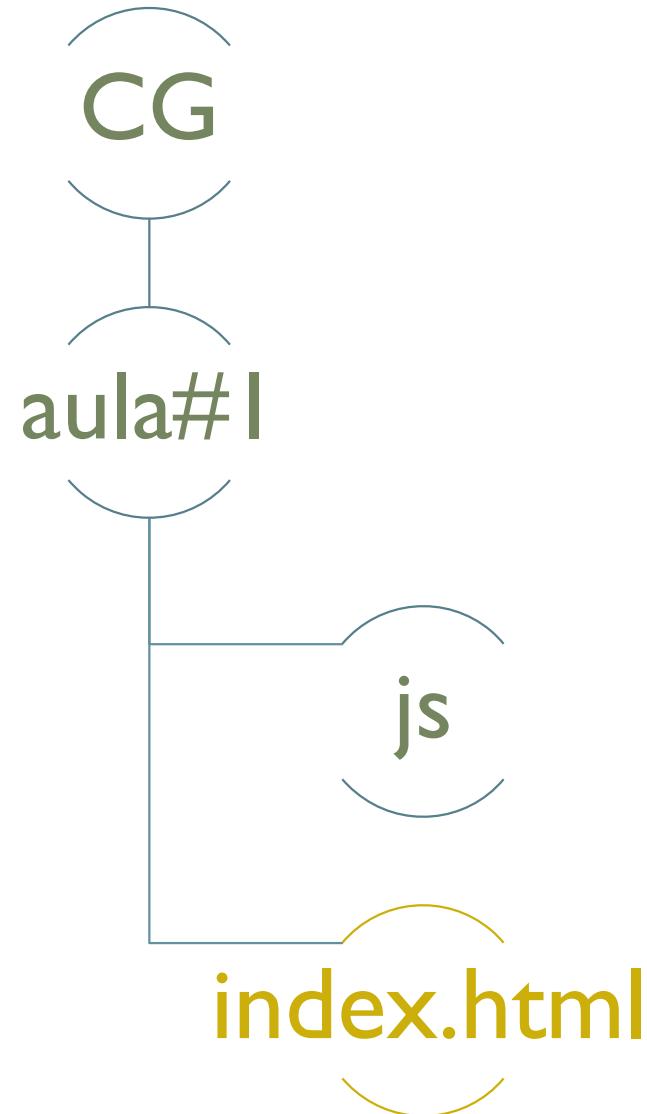


web app “Hello 3D World”



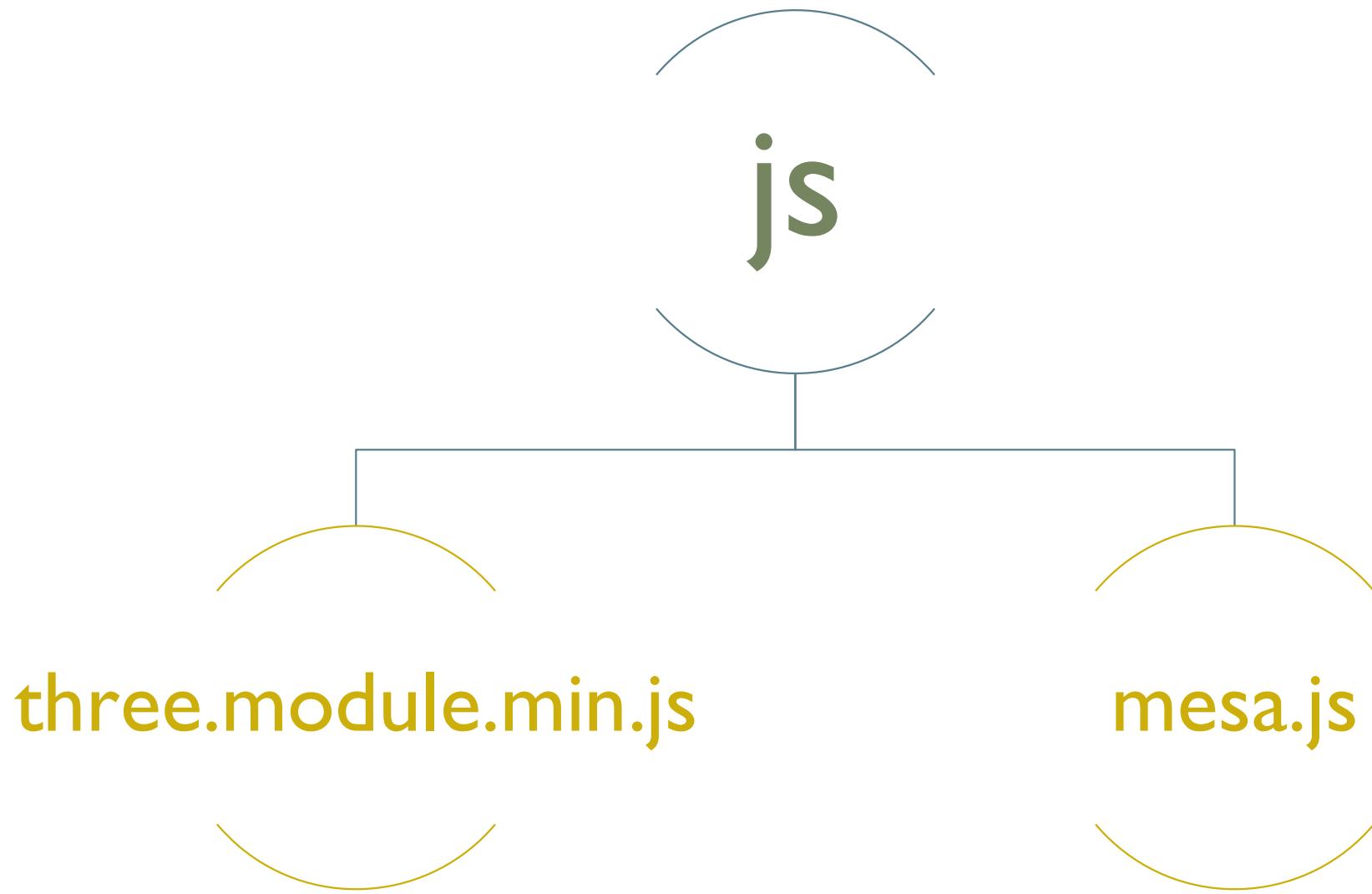
web app “Hello 3D World”

criar pastas e ficheiros



web app “Hello 3D World”

criar pastas e ficheiros



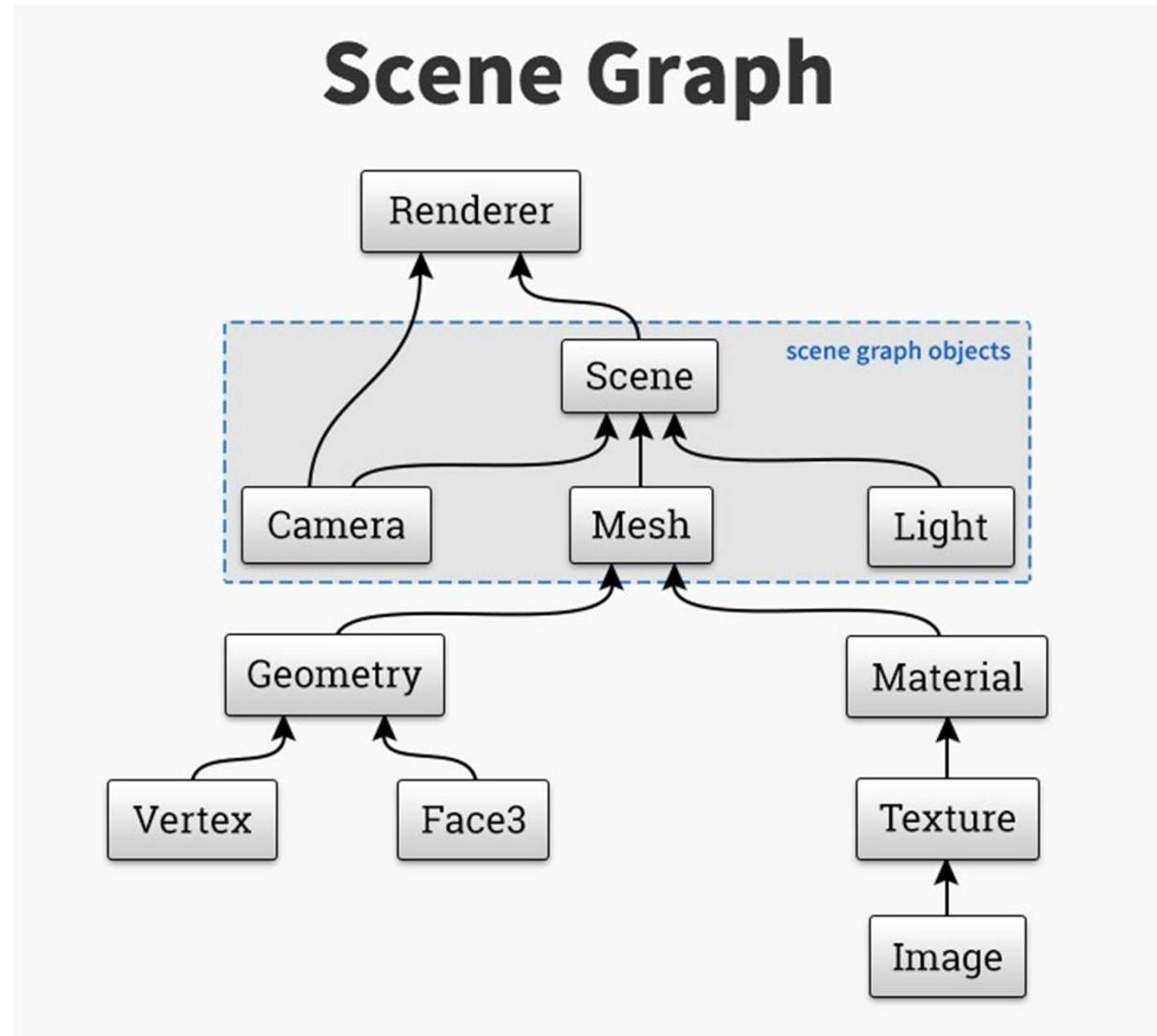
web app “Hello 3D World”

index.html

```
1  <!doctype html>
2  <html>
3  <head>
4      <meta charset="utf-8">
5      <title>1&ordf; aula - CG@IST/UL</title>
6      <style>
7          body {
8              margin: 0px;
9              background-color: #fff;
10             overflow: hidden;~
11         </style>
12         <script type="importmap">
13             {
14                 "imports": {
15                     "three": "https://cdn.jsdelivr.net/npm/three@v0.163.0/build/three.module.js"
16                 }
17             }
18         </script>
19     </head>
20     <body>
21         <script type="module" src="js/mesa.js"></script>
22     </body>
23 </html>
```

web app “Hello 3D World”

Three.js Scene Graph



web app “Hello 3D World”

mesa.js

```
1 import * as THREE from 'three';
2
3 var camera, scene, renderer;
4
5 function render() { }
6
7 function init() { }
```

web app “Hello 3D World”

mesa.js

```
1 import * as THREE from 'three';
2
3 var camera, scene, renderer;
4
5 function render() { }
6
7 function init() {
8     'use strict';
9     renderer = new THREE.WebGLRenderer({ antialias: true });
10    renderer.setSize(window.innerWidth, window.innerHeight);
11    document.body.appendChild(renderer.domElement);
12
13    render();
14 }
```

web app “Hello 3D World”

mesa.js

```
1 import * as THREE from 'three';
2
3 var camera, scene, renderer;
4
5 <function render() {
6   'use strict';
7   renderer.render(scene, camera);
8 }
9
10 > function init()...
```

web app “Hello 3D World”

mesa.js

```
1 import * as THREE from 'three';
2
3 var camera, scene, renderer;
4
5 >function render()...
9
10 <function createScene() {
11   'use strict';
12
13   scene = new THREE.Scene();
14
15   scene.add(new THREE.AxesHelper(10));
16
17 }
18
19 >function init()...
```

web app “Hello 3D World”

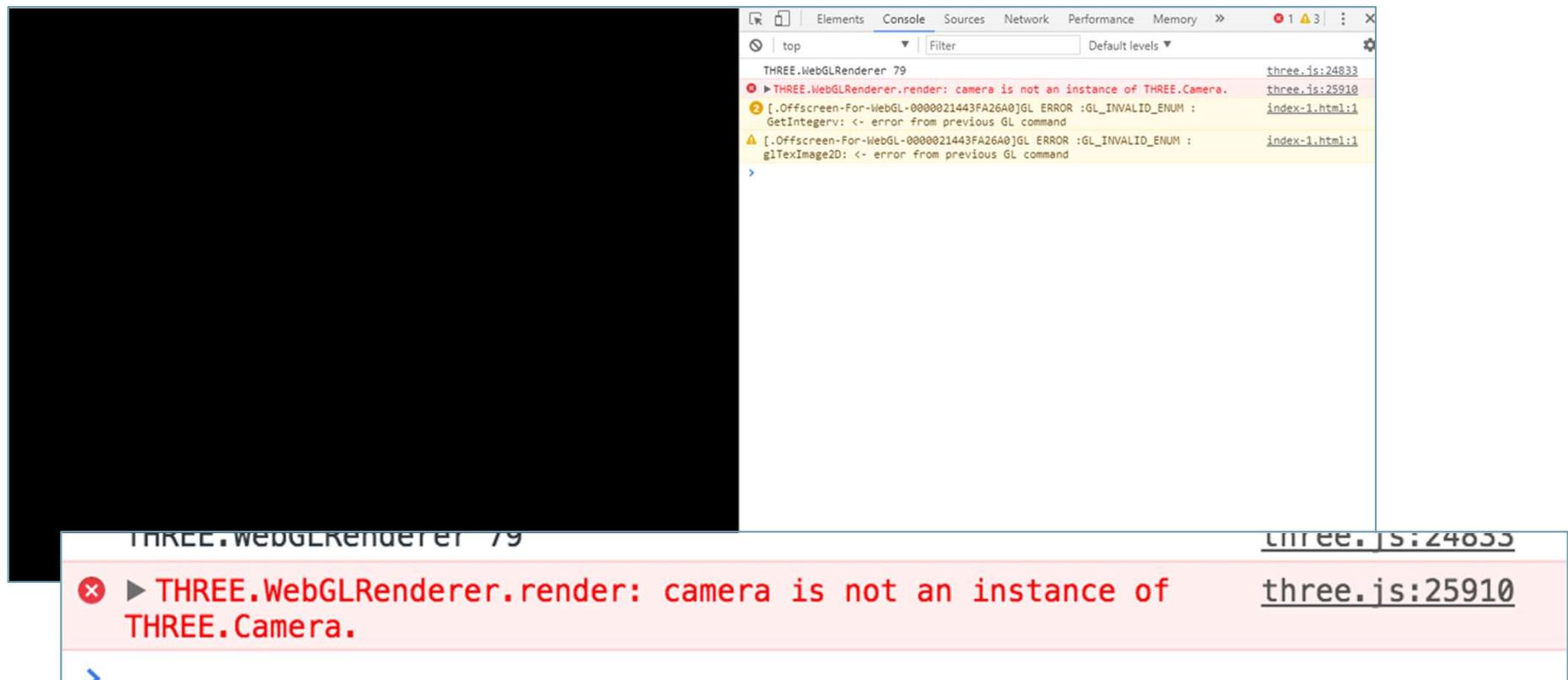
mesa.js

```
1 import * as THREE from 'three';
2
3 var camera, scene, renderer;
4
5 >function render()...
9
10 >function createScene()...
18
19 <function init() {
20   'use strict';
21   <renderer = new THREE.WebGLRenderer({ antialias: true });
22   renderer.setSize(window.innerWidth, window.innerHeight);
23   document.body.appendChild(renderer.domElement);
24
25   createScene();
26
27   render();
28 }
29
30 init();
```

web app “Hello 3D World”

mesa.js

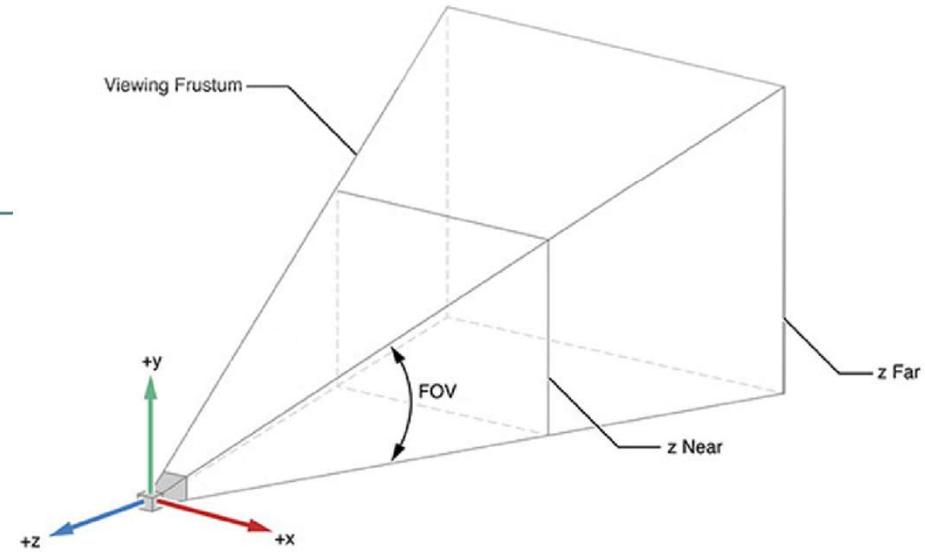
mas, ao executar a web app ...



web app “Hello 3D World”

mesa.js

```
1 import * as THREE from 'three';
2
3 var camera, scene, renderer;
4
5 >function render()...
9
10 <function createCamera() {
11   'use strict';
12   //PerspectiveCamera( fov, aspect, near, far )
13   camera = new THREE.PerspectiveCamera(70,
14                                         window.innerWidth / window.innerHeight,
15                                         1,
16                                         1000);
17
18   camera.position.x = 50;
19   camera.position.y = 50;
20   camera.position.z = 50;
21   camera.lookAt(scene.position);
22 }
22
23 >function createScene()...
32
33 >function init()...
44
45   init();
```



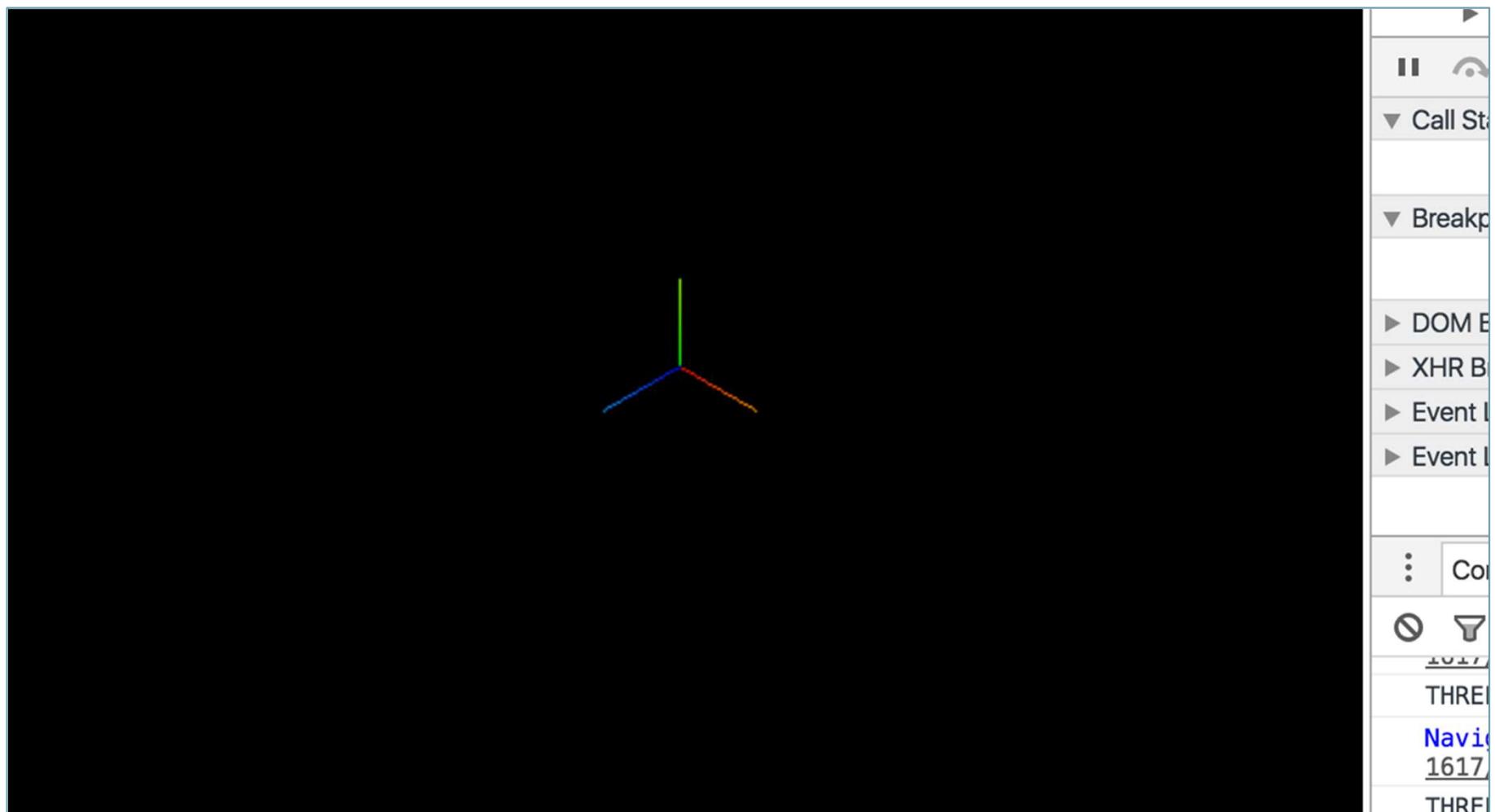
web app “Hello 3D World”

mesa.js

```
1 import * as THREE from 'three';
2
3 var camera, scene, renderer;
4
5 >function render()...
9
10 >function createCamera()...
22
23 >function createScene()...
32
33 >function init() {
34   'use strict';
35   < renderer = new THREE.WebGLRenderer({ antialias: true});
36   renderer.setSize(window.innerWidth, window.innerHeight);
37   document.body.appendChild(renderer.domElement);
38
39   createScene();
40   createCamera();
41
42   render();
43 }
44
45 init();
```

web app “Hello 3D World”

mesa.js



web app “Hello 3D World”

mesa.js

```
1 import * as THREE from 'three';
2
3 var camera, scene, renderer;
4
5 var material;
6
7 function createTable(x, y, z) {
8     'use strict';
9
10    var table = new THREE.Object3D();
11
12    material = new THREE.MeshBasicMaterial({ color: 0x00ff00, wireframe: true });
13
14    scene.add(table);
15
16    table.position.x = x;
17    table.position.y = y;
18    table.position.z = z;
19}
20
21 >function createCamera()...
22
23 >function createScene()...
24
25 >function render()...
26
27 >function init()...
28
29    init();
```

web app “Hello 3D World”

mesa.js

```
3  var camera, scene, renderer;
4
5  var material;
6
7  <function createTable(x, y, z) {
8      'use strict';
9
10     var table = new THREE.Object3D();
11
12     material = new THREE.MeshBasicMaterial({ color: 0x00ff00, wireframe: true });
13
14     addTableTop(table, 0, 0, 0);
15     addTableLeg(table, -25, -1, -8);
16     addTableLeg(table, -25, -1, 8);
17     addTableLeg(table, 25, -1, 8);
18     addTableLeg(table, 25, -1, -8);
19
20     scene.add(table);
21
22     table.position.x = x;
23     table.position.y = y;
24     table.position.z = z;
25 }
26
27 >function createCamera()...
28
29 >function createScene()...
30
31 >function render()...
```

web app “Hello 3D World”

mesa.js

```
1 import * as THREE from 'three';
2
3 var camera, scene, renderer;
4
5 var geometry, material, mesh;
6
7 <function addTableTop(obj, x, y, z) {
8     'use strict';
9     // BoxGeometry(width, height, length)
10    geometry = new THREE.BoxGeometry(60, 2, 20);
11    mesh = new THREE.Mesh(geometry, material);
12    mesh.position.set(x, y, z);
13    obj.add(mesh);
14 }
15
16 > function createTable(x, y, z)...
35
36 > function createCamera()...
47
48 > function createScene()...
59
60 > function render()...
64
65 > function init()...
```

web app “Hello 3D World”

mesa.js

```
1 import * as THREE from 'three';
2
3 var camera, scene, renderer;
4
5 var geometry, material, mesh;
6
7 <function addTableLeg(obj, x, y, z) {
8     'use strict';
9
10    geometry = new THREE.BoxGeometry(2, 6, 2);
11    mesh = new THREE.Mesh(geometry, material);
12    mesh.position.set(x, y - 3, z);
13    obj.add(mesh);
14 }
15
16 >function addTableTop(obj, x, y, z)...
24
25 >function createTable(x, y, z)...
44
45 >function createCamera()...
56
57 >function createScene()...
68
69 >function render()...
```

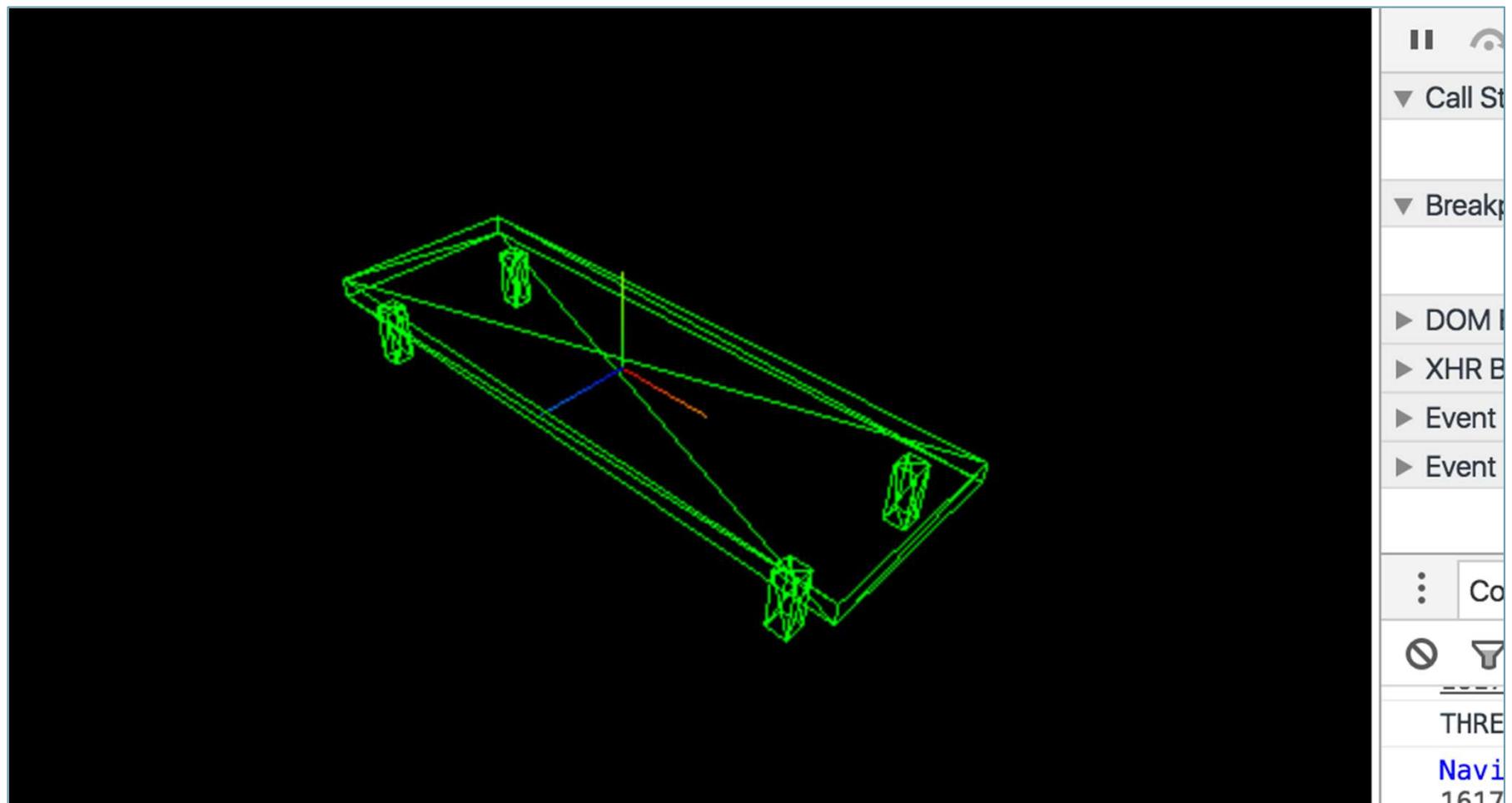
web app “Hello 3D World”

mesa.js

```
3  var camera, scene, renderer;
4
5  var geometry, material, mesh;
6
7  >function addTableLeg(obj, x, y, z)...
15
16 >function addTableTop(obj, x, y, z)...
24
25 >function createTable(x, y, z)...
44
45 >function createCamera()...
56
57 <function createScene() {
58   'use strict';
59
60   scene = new THREE.Scene();
61   | scene.add(new THREE.AxesHelper(10));
62
63   createTable( 0, 0, 0);
64
65 }
66
67 >function render()...
```

web app “Hello 3D World”

mesa.js



web app “Hello 3D World”

mesa.js

```
5   var geometry, material, mesh,
6
7   var ball;
8
9   <function createBall(x, y, z) {
10    'use strict';
11
12    ball = new THREE.Object3D();
13    ball.userData = { jumping: true, step: 0 };
14
15    material = new THREE.MeshBasicMaterial({ color: 0xff0000, wireframe: true });
16
17    //SphereGeometry(radius, segmentsW, segmentsH, phiStart, phiLen, thetaStart, thetaLen)
18    geometry = new THREE.SphereGeometry(4, 10, 10);
19    mesh = new THREE.Mesh(geometry, material);
20
21    ball.add(mesh);
22    ball.position.set(x, y, z);
23
24    scene.add(ball);
25  }
26
27  >function addTableLeg(obj, x, y, z)...
35
36  >function addTableTop(obj, x, y, z)...
43
44  >function createTable(x, y, z)...
```

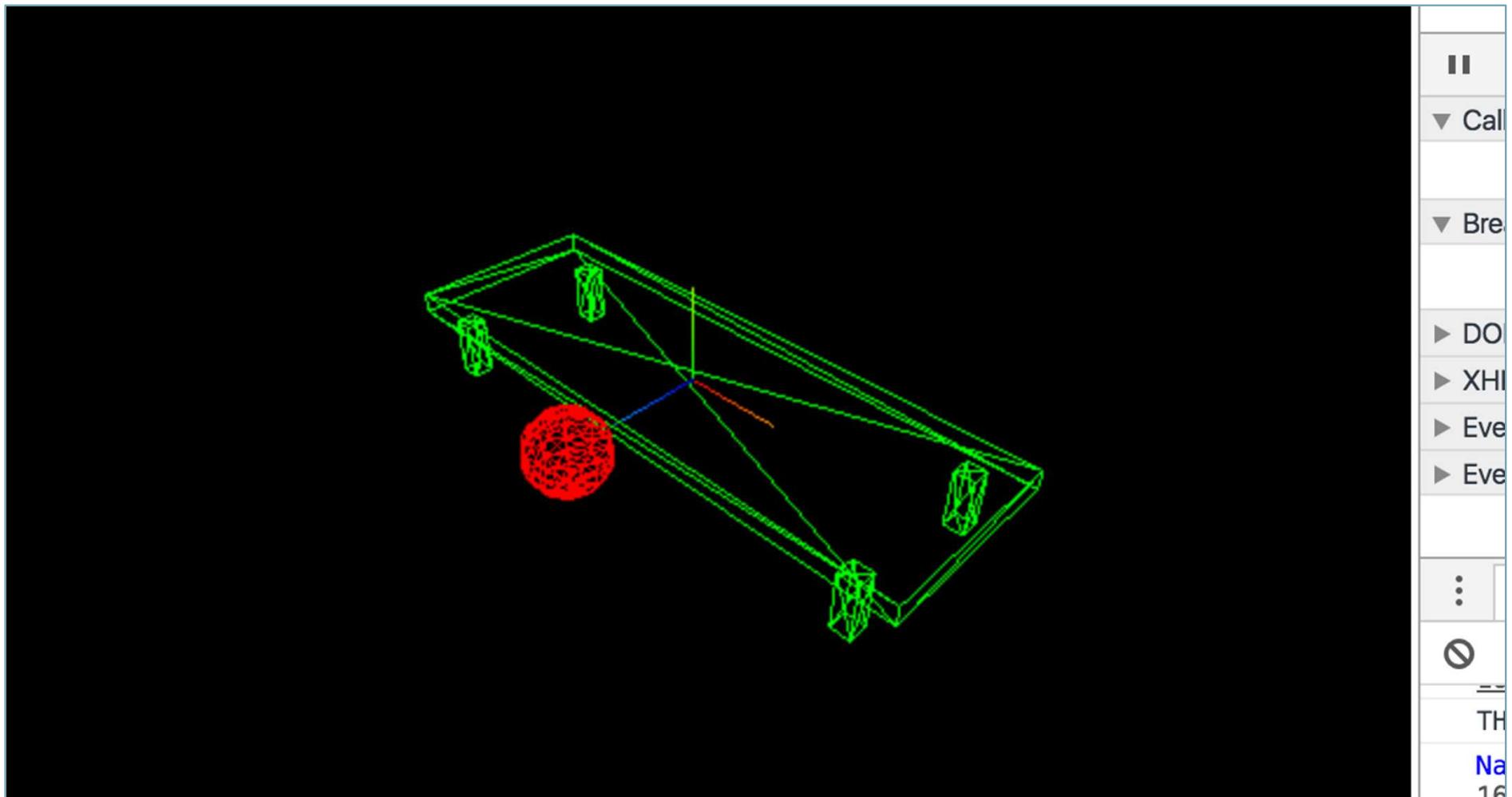
web app “Hello 3D World”

mesa.js

```
7   var ball;
8
9   >function createBall(x, y, z)...
26
27   >function addTableLeg(obj, x, y, z)...
35
36   >function addTableTop(obj, x, y, z)...
43
44   >function createTable(x, y, z)...
63
64   >function createCamera()...
75
76   <function createScene() {
77       'use strict';
78
79       scene = new THREE.Scene();
80
81
82       scene.add(new THREE.AxesHelper(10));
83
84       createTable( 0, 0, 0);
85       createBall(0, 0, 15);
86   }
```

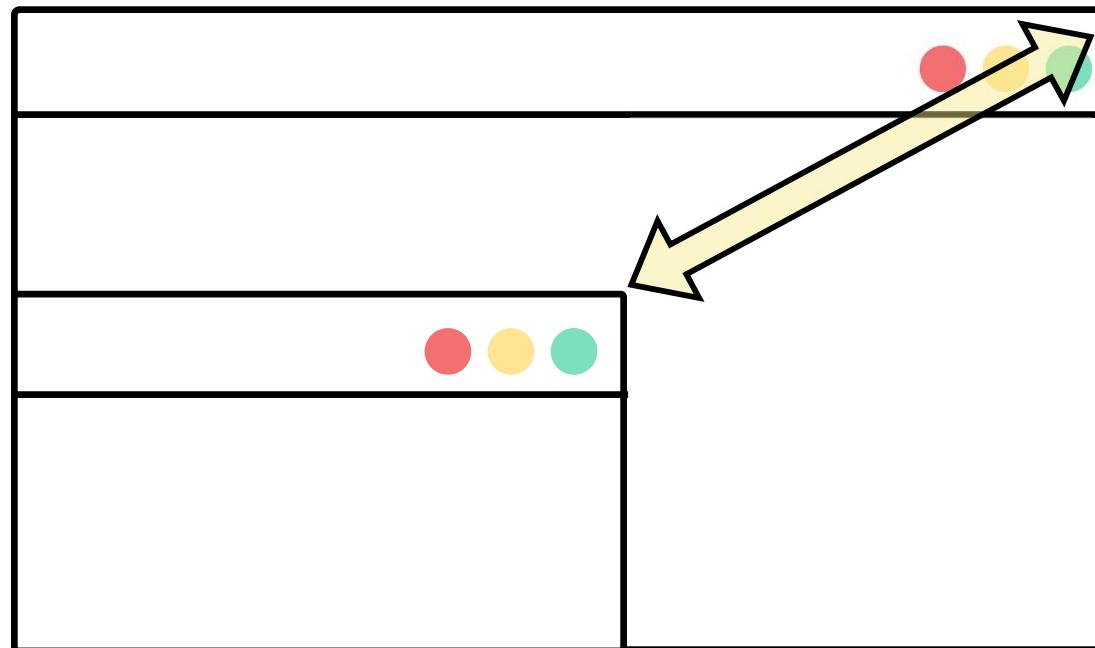
web app “Hello 3D World”

mesa.js



web app “Hello 3D World”

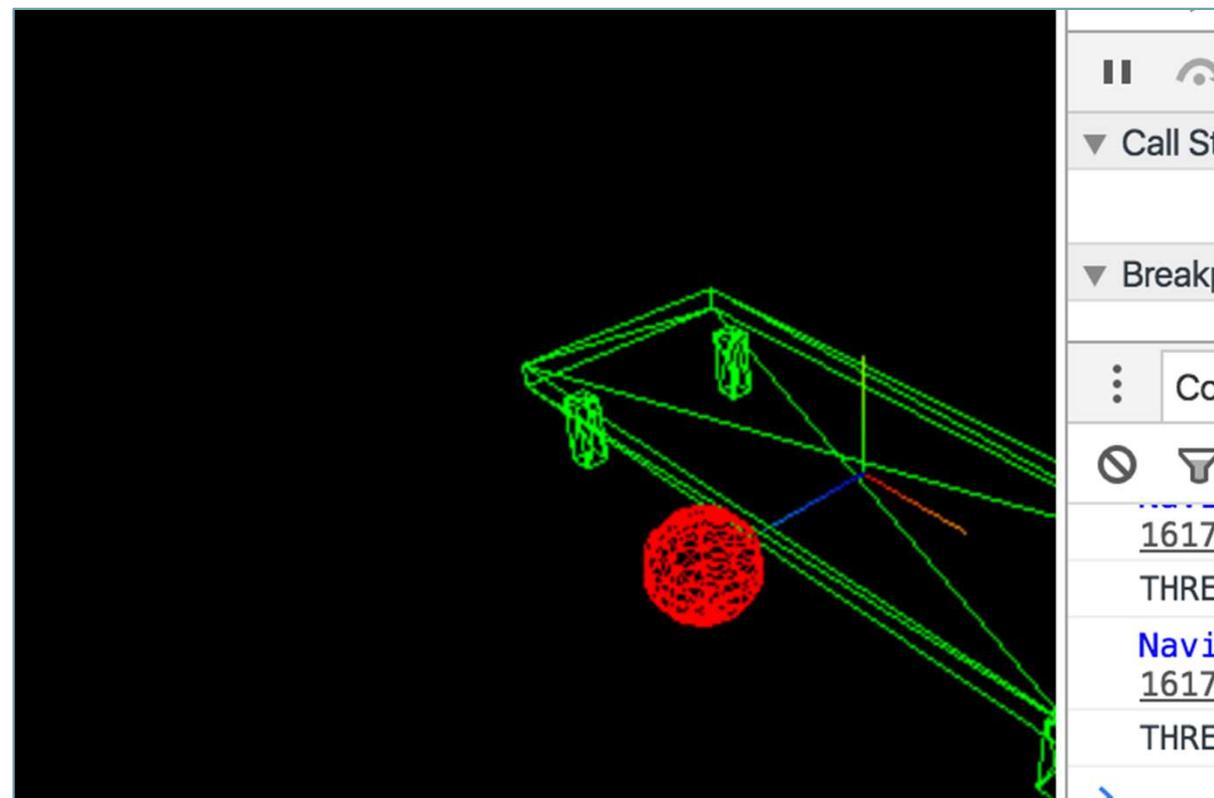
mesa.js – redimensionar a janela



web app “Hello 3D World”

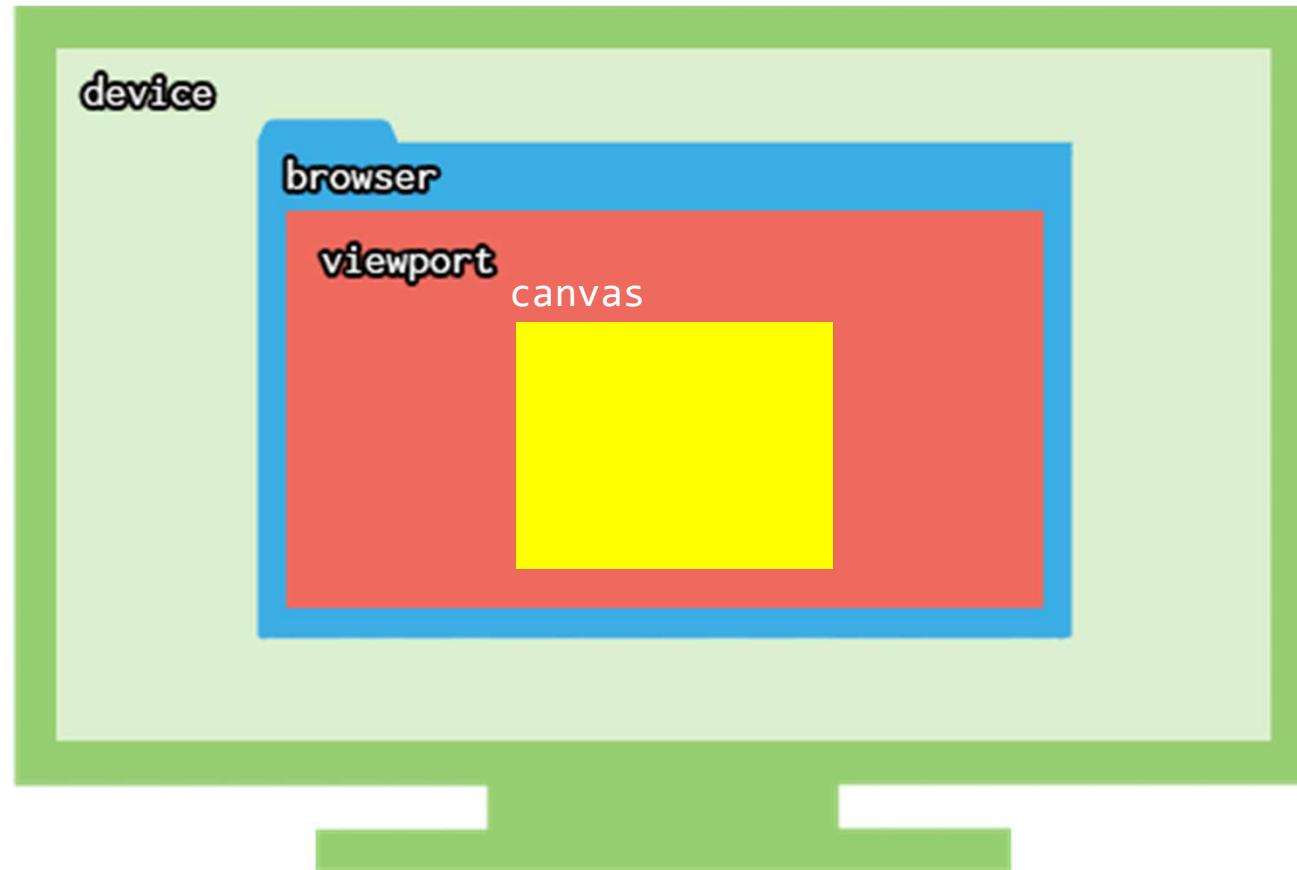
mesa.js – redimensionar a janela

mas, ao redimensionar a janela ...



web app “Hello 3D World”

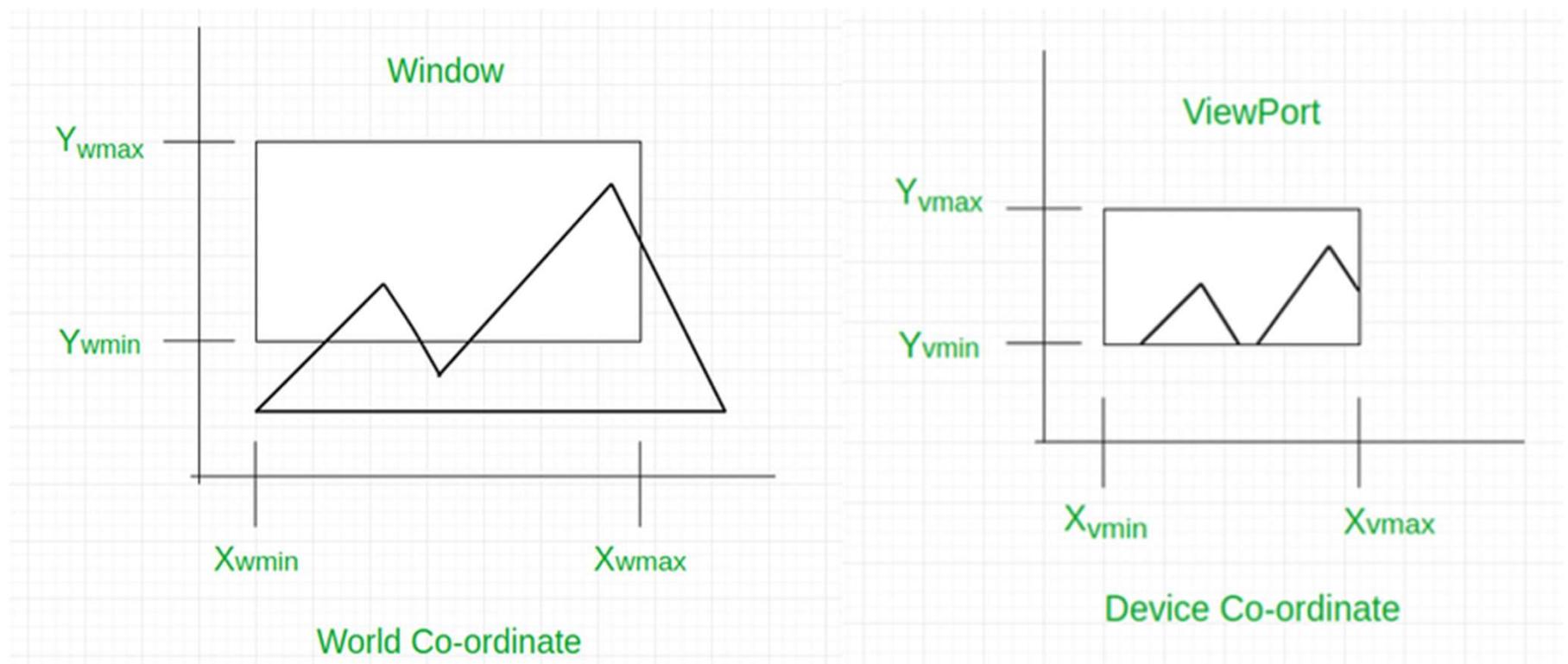
mesa.js – redimensionar a janela



Jargão ‘Web Design’

web app “Hello 3D World”

mesa.js – redimensionar a janela



Jargão ‘Computer Graphics’

web app “Hello 3D World”

mesa.js – redimensionar a janela

```
42 > function createCanvas(x, y, z)...
61
62   > function createCamera()...
73
74   > function createScene()...
85
86   <function onResize() {
87     |   'use strict';
88
89     |   renderer.setSize(window.innerWidth, window.innerHeight);
90   }
91
92   > function render()...
96
97   > function init()...
```

web app “Hello 3D World”

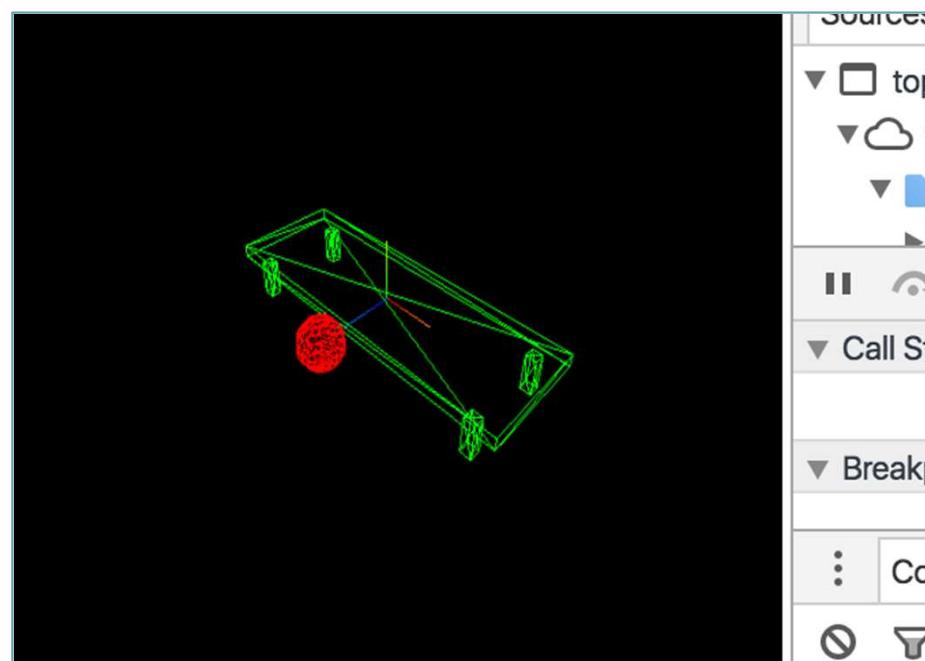
mesa.js – redimensionar a janela

```
85
86     >function onResize()...
87
88     >function render()...
89
90
91     >function init() {
92         'use strict';
93
94         renderer = new THREE.WebGLRenderer({ antialias: true});
95         renderer.setSize(window.innerWidth, window.innerHeight);
96         document.body.appendChild(renderer.domElement);
97
98         createScene();
99         createCamera();
100
101         render();
102
103         window.addEventListener("resize", onResize);
104     }
105
106     init();
```

web app “Hello 3D World”

mesa.js – redimensionar a janela

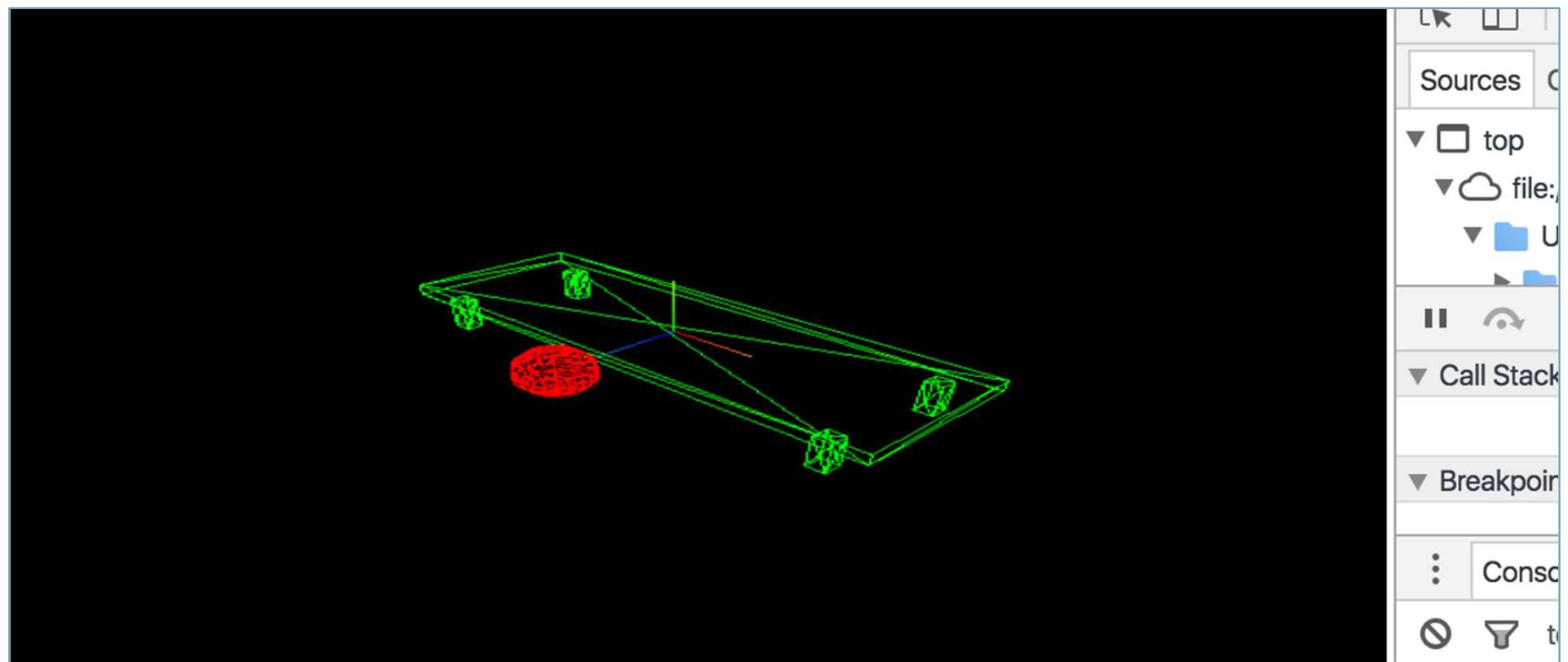
ao redimensionar a janela ...



web app “Hello 3D World”

mesa.js – redimensionar a janela

ao redimensionar a janela ...



web app “Hello 3D World”

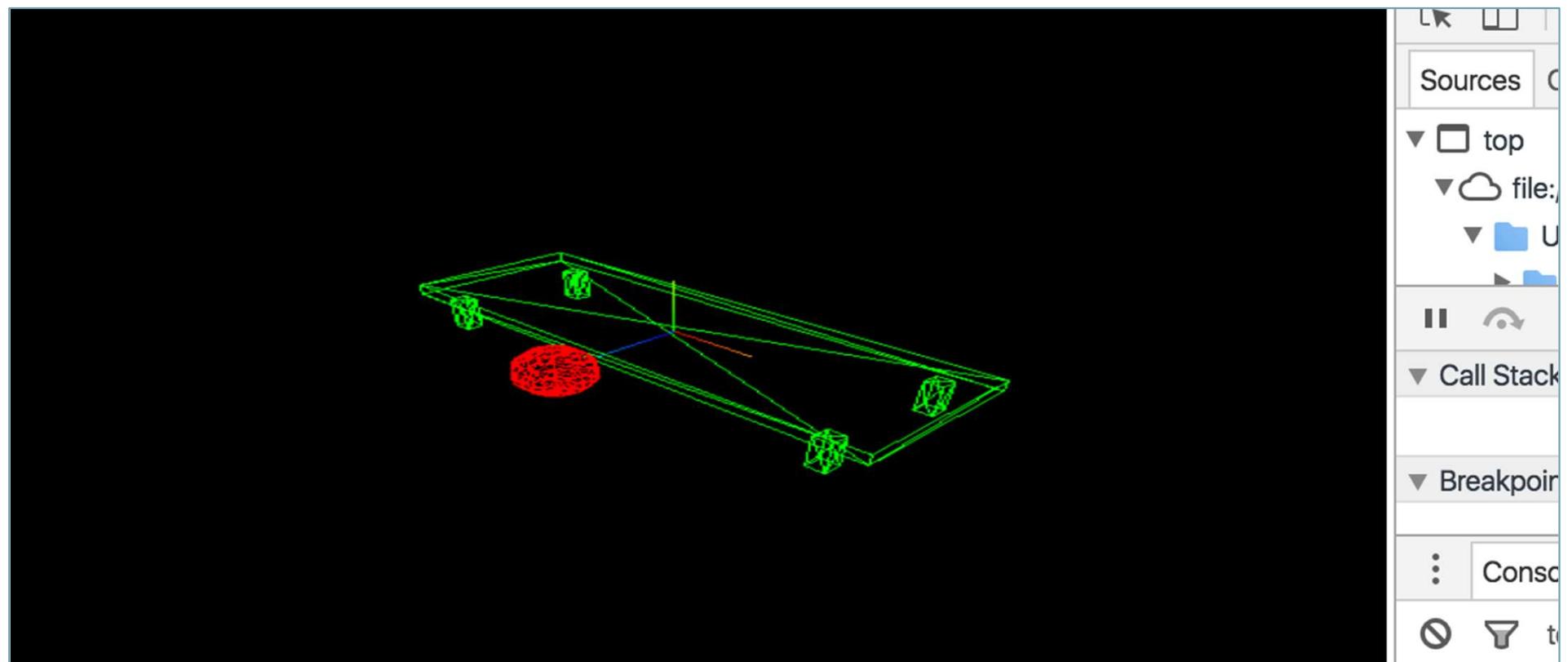
mesa.js – redimensionar a janela

```
 62 > function createCamera() ...
73
74   > function createScene() ...
85
86   <function onResize() {
87     'use strict';
88
89     renderer.setSize(window.innerWidth, window.innerHeight);
90
91     if (window.innerHeight > 0 && window.innerWidth > 0) {
92       //camera.aspect = window.innerWidth / window.innerHeight;
93       camera.aspect = renderer.getSize().width/renderer.getSize().height;
94       camera.updateProjectionMatrix();
95     }
96   }
97
98   > function render() ...
102
103   > function init() ...
```

web app “Hello 3D World”

mesa.js – redimensionar a janela

ao redimensionar a janela ...



web app “Hello 3D World”

mesa.js – redimensionar a janela

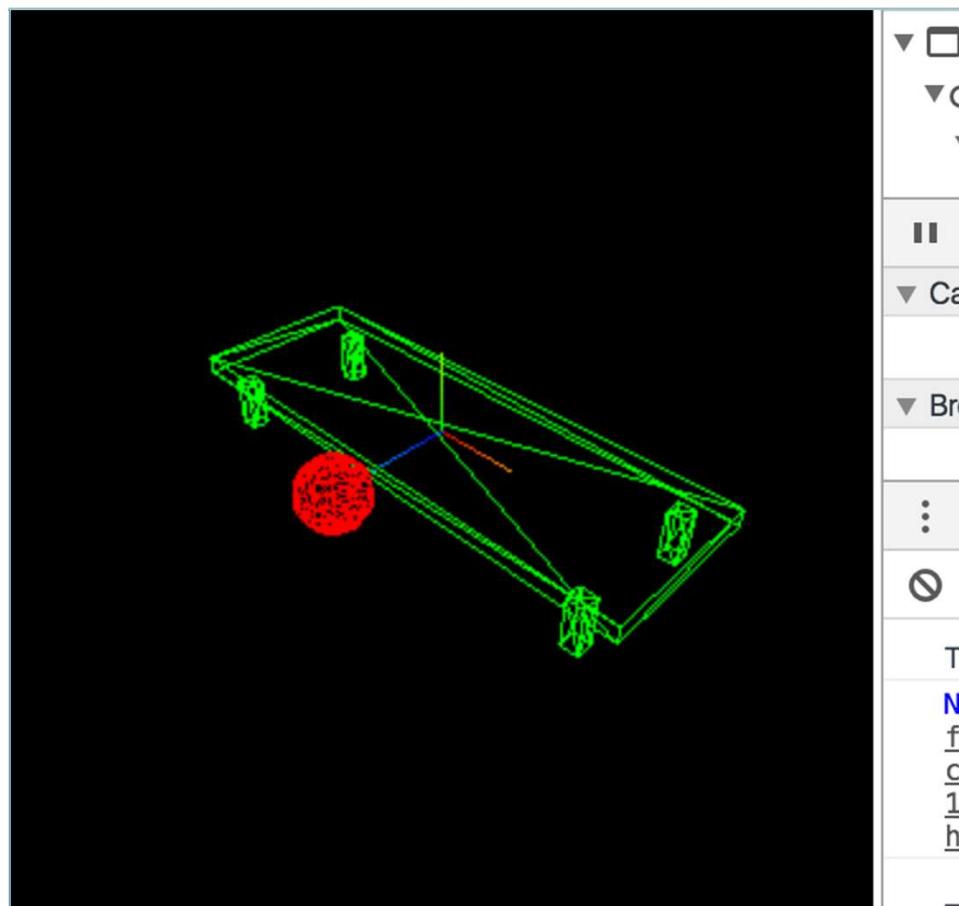
```
86  <function onResize() {
87    'use strict';
88
89    renderer.setSize(window.innerWidth, window.innerHeight);
90
91    if (window.innerHeight > 0 && window.innerWidth > 0) {
92      //camera.aspect = window.innerWidth / window.innerHeight;
93      camera.aspect = renderer.getSize().width/renderer.getSize().height;
94      camera.updateProjectionMatrix();
95    }
96
97    render();
98  }
99
100 >function render(...)
```

```
104 >function init(...)
```

web app “Hello 3D World”

mesa.js – redimensionar a janela

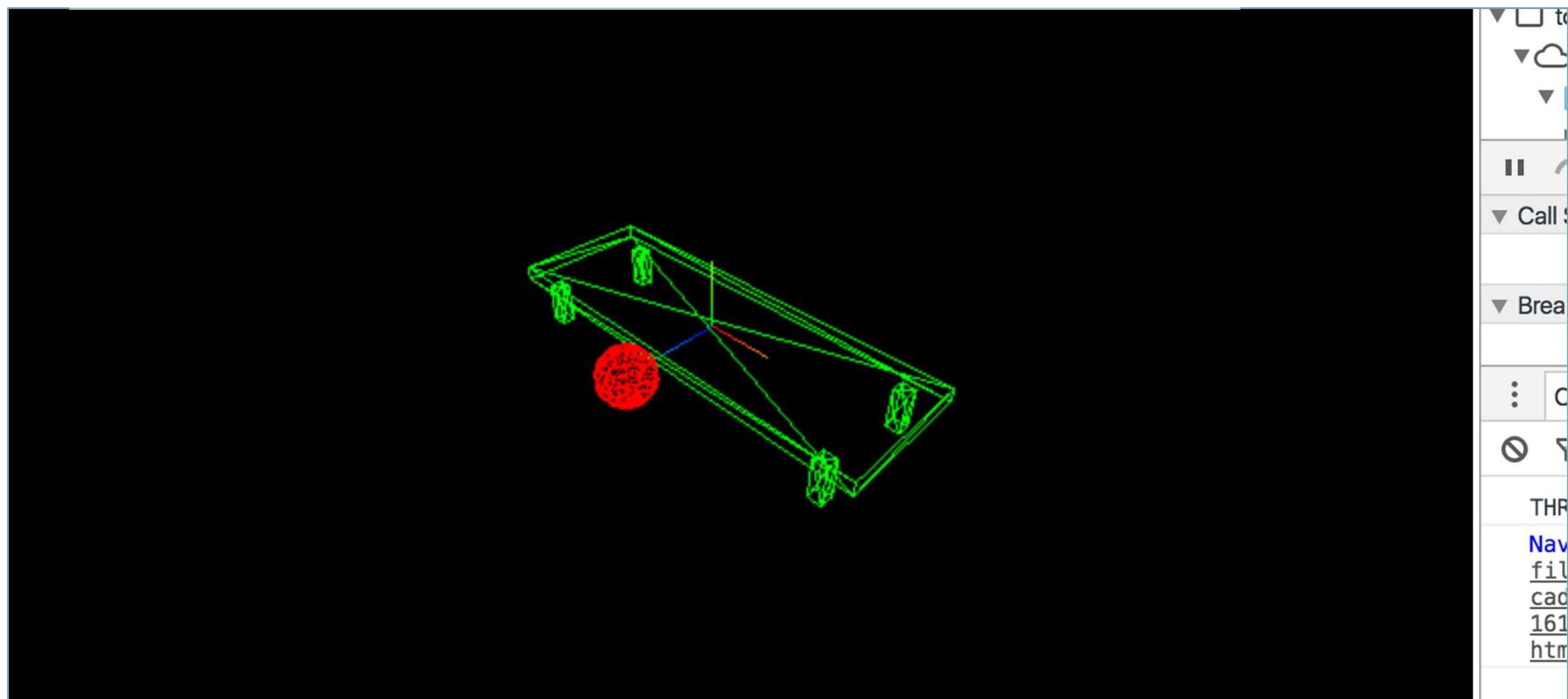
ao redimensionar a janela ...



web app “Hello 3D World”

mesa.js – redimensionar a janela

ao redimensionar a janela ...



web app “Hello 3D World”

mesa.js – interacção

```
74  >function createScene()...
85
86  >function onResize()...
98
99  >function render()...
103
104 <function init() {
105   |  'use strict';
106   |< renderer = new THREE.WebGLRenderer({ antialias: true});
107   |< renderer.setSize(window.innerWidth, window.innerHeight);
108   |< document.body.appendChild(renderer.domElement);
109
110   |< createScene();
111   |< createCamera();
112
113   |< render();
114
115   |< window.addEventListener("resize", onResize);
116   |< window.addEventListener("keydown", onKeyDown);
117 }
118
119 <init();
```

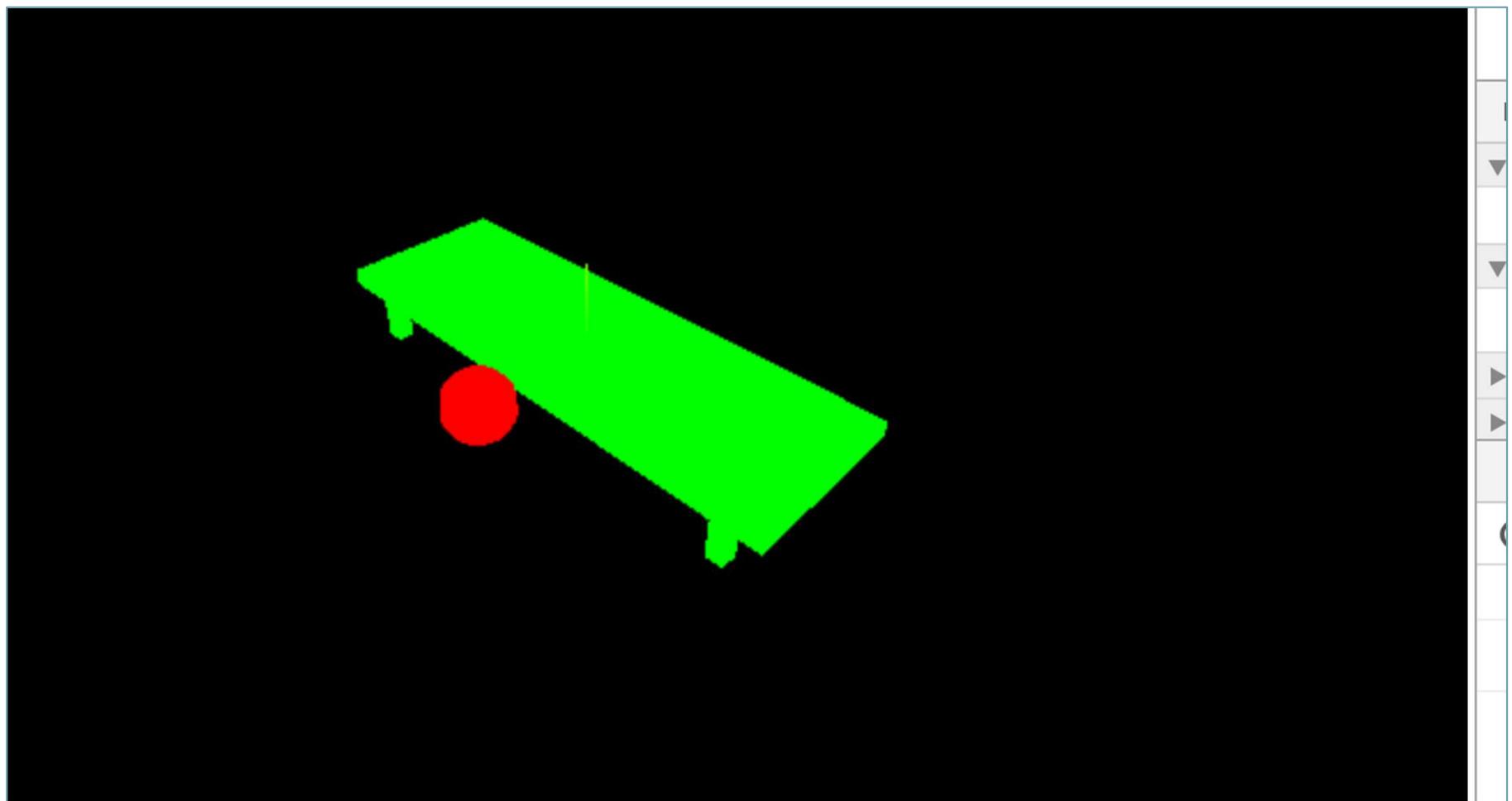
web app “Hello 3D World”

mesa.js – interacção

```
74  >function createScene()...
85
86  >function onResize()...
98
99  >function render()...
103
104  <function onKeyDown(e) {
105      'use strict';
106
107      switch (e.keyCode) {
108          case 65: //A
109          case 97: //a
110              scene.traverse(function (node) {
111                  if (node instanceof THREE.Mesh) {
112                      node.material.wireframe = !node.material.wireframe;
113                  }
114              });
115              break;
116      }
117
118      render();
119  }
120
121  >function init()...
```

web app “Hello 3D World”

mesa.js – interacção



web app “Hello 3D World”

mesa.js – interacção

```
101
102     <function onKeyDown(e) {
103         'use strict';
104
105         switch (e.keyCode) {
106             case 65: //A
107             case 97: //a
108                 scene.traverse(function (node) {
109                     if (node instanceof THREE.Mesh) {
110                         node.material.wireframe = !node.material.wireframe;
111                     }
112                 });
113                 break;
114             case 83: //S
115             case 115: //s
116                 ball.userData.jumping = !ball.userData.jumping;
117                 break;
118             }
119         }
120     }
121
122     >function init()...
```

web app “Hello 3D World”

mesa.js – interacção

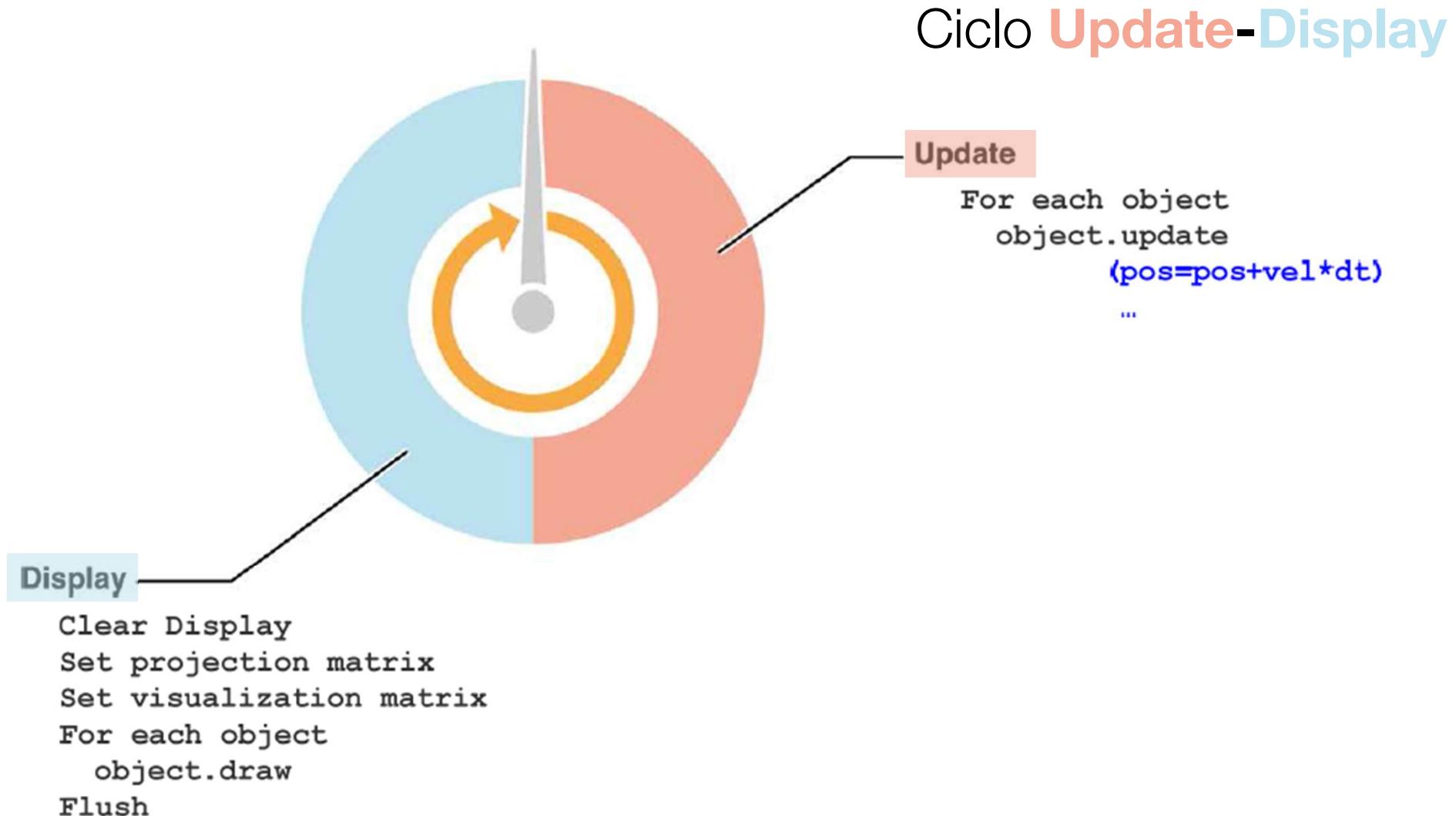
web app “Hello 3D World”

mesa.js – interacção

```
101
102  <function onKeyDown(e) {
103    'use strict';
104
105    switch (e.keyCode) {
106      case 65: //A
107      case 97: //a
108        scene.traverse(function (node) {
109          if (node instanceof THREE.Mesh) {
110            node.material.wireframe = !node.material.wireframe;
111          }
112        });
113        break;
114      case 83: //S
115      case 115: //s
116        ball.userData.jumping = !ball.userData.jumping;
117        break;
118    }
119  }
120
121  >function init()...
```

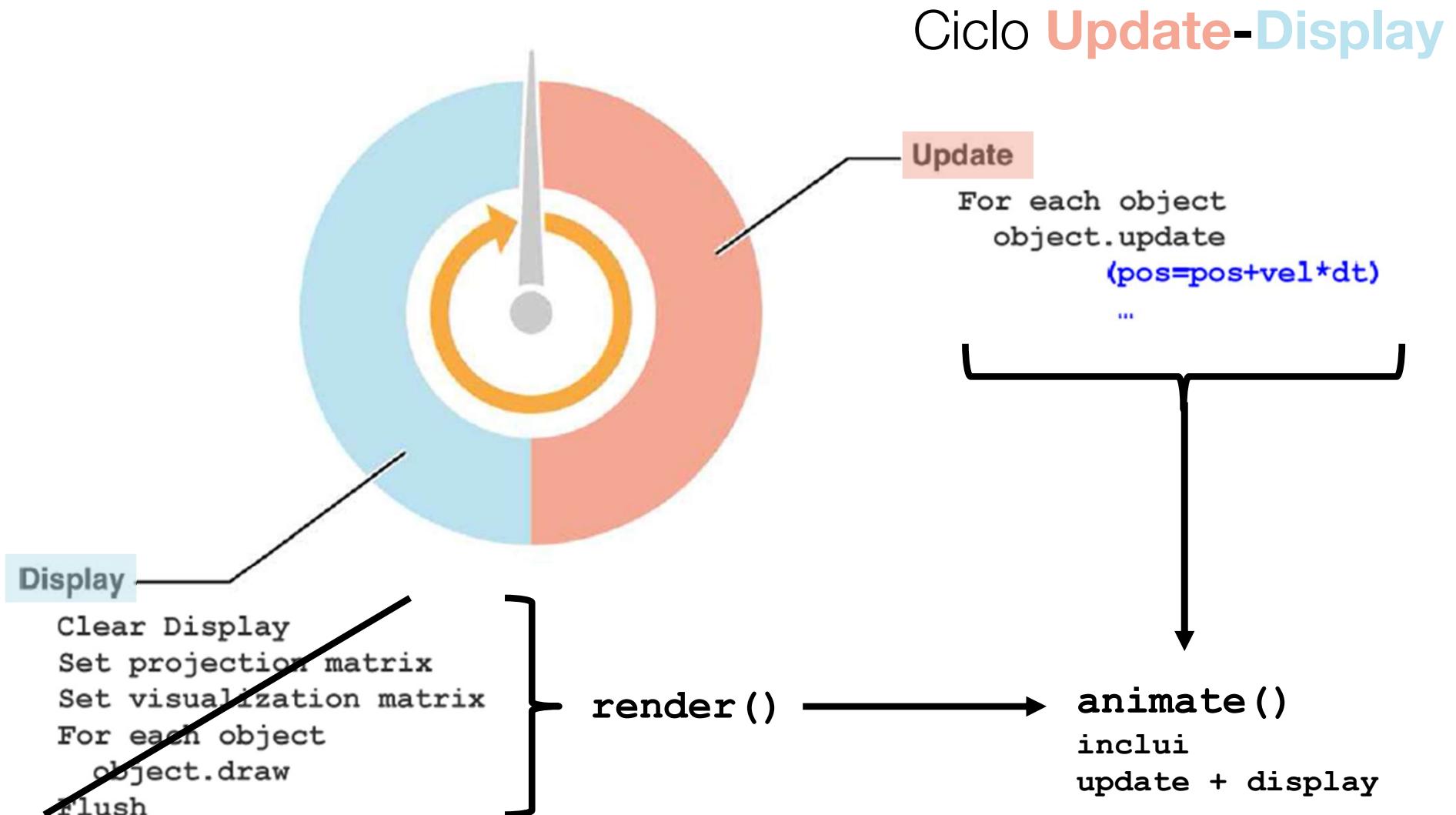
web app “Hello 3D World”

mesa.js – animação



web app “Hello 3D World”

mesa.js – animação



web app “Hello 3D World”

mesa.js – animação

```
102  > function onKeyDown(e)...
121
122  ✓function animate() {
123      |  'use strict';
124
125      |<
126      |  if (ball.userData.jumping) {
127          |      ball.userData.step += 0.04;
128          |      ball.position.y = Math.abs(30 * (Math.sin(ball.userData.step)));
129          |
130          |      render();
131
132          |      requestAnimationFrame(animate);
133      }
134
135  > function init()...
149
150      init();
151
152      animate();
```

web app “Hello 3D World”

mesa.js – animação

```
91  ✓function onResize() {
92    'use strict';
93
94    renderer.setSize(window.innerWidth, window.innerHeight);
95
96    ✓if (window.innerHeight > 0 && window.innerWidth > 0) {
97      camera.aspect = window.innerWidth / window.innerHeight;
98      camera.updateProjectionMatrix();
99    }
100
101   ✓render(),
102 }
103
104 ✓function onKeyDown(e) {
105   'use strict';
106
107   ✓switch (e.keyCode) {
108     case 65: //A
109     case 97: //a
110       ✓scene.traverse(function (node) {
111         if (node instanceof THREE.Mesh) {
112           node.material.wireframe = !node.material.wireframe;
113         }
114       });
115       break;
116     case 83: //S
117     case 115: //s
118       ball.userData.jumping = !ball.userData.jumping;
119       break;
120     }
121
122   ✓render(),
123 }
124
125 >function animate()...
```

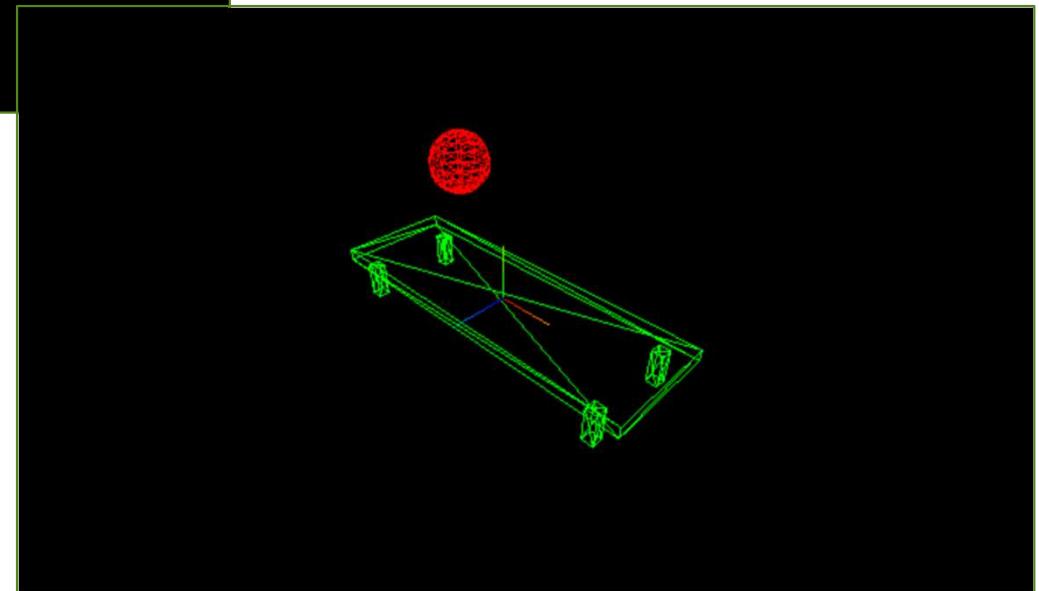
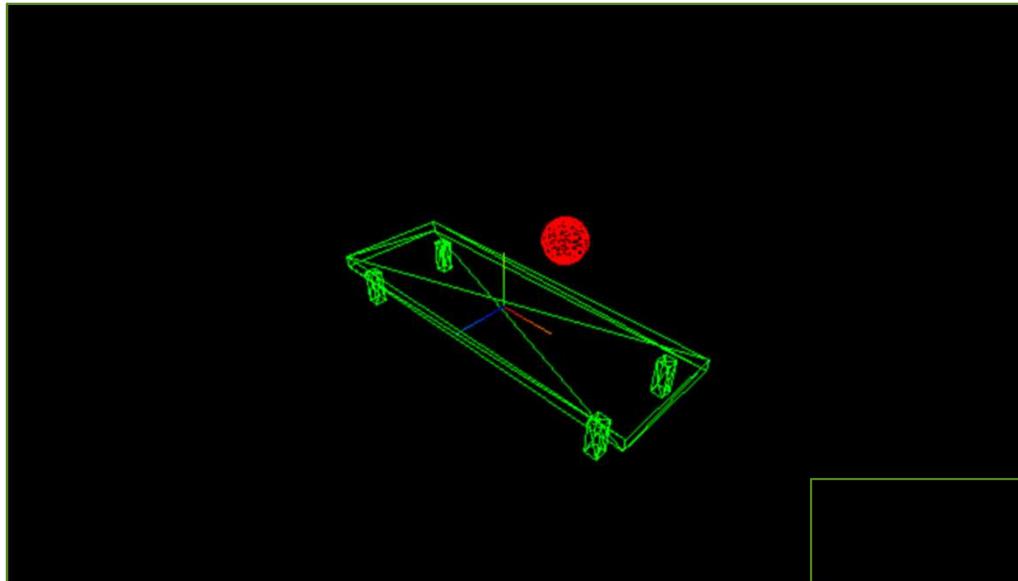
web app “Hello 3D World”

mesa.js – animação

```
91  <function onResize() {
92    'use strict';
93
94    renderer.setSize(window.innerWidth, window.innerHeight);
95
96    if (window.innerHeight > 0 && window.innerWidth > 0) {
97      camera.aspect = window.innerWidth / window.innerHeight;
98      camera.updateProjectionMatrix();
99    }
100  }
101
102 <function onKeyDown(e) {
103   'use strict';
104
105   switch (e.keyCode) {
106     case 65: //A
107     case 97: //a
108       scene.traverse(function (node) {
109         if (node instanceof THREE.Mesh) {
110           node.material.wireframe = !node.material.wireframe;
111         }
112       });
113       break;
114     case 83: //S
115     case 115: //s
116       ball.userData.jumping = !ball.userData.jumping;
117       break;
118     }
119   }
120
121 >function animate()...
```

web app “Hello 3D World”

mesa.js – animação



NOTA IMPORTANTE

Este exemplo ([mesa.js](#)) tem, propositadamente, vários erros e segue opções de codificação incorrectas.

A sua utilização nos trabalhos dará origem a penalização na avaliação.

É suposto os alunos serem capazes de identificar e ultrapassar estas situações.

apoio à execução dos trabalhos

apoio

Trabalho A – Tarefa 2 e Tarefa 3

NOTA:

O objecto chave do Trabalho B é a ...

Grua

O objecto chave do Trabalho C é o ...

Carrossel Matemático

apoio

Trabalhos B & C – HTML Template

...\Template – Trabalhos B e C\index.html

```
1 <!doctype html>
2 <html>
3 <head>
4     <meta charset="utf-8" />
5     <title>Hello 3D World!</title>
6     <link rel="stylesheet" href="css/style.css">
7     <script type="importmap">
8         {
9             "imports": {
10                 "three": "https://cdn.jsdelivr.net/npm/three@v0.163.0/build/three.module.js",
11                 "three/addons/": "https://cdn.jsdelivr.net/npm/three@0.163.0/examples/jsm/"
12             }
13         }
14     </script>
15 </head>
16 <body>
17     <script type="module" src="js/main-script.js"></script>
18 </body>
19 </html>
```

apoio

Trabalhos B & C – CSS Template

...\Template - Trabalhos B e C\css\style.css

```
1 body {  
2     margin: 0px;  
3     background-color: #fff;  
4     overflow: hidden;  
5 }
```

apoio

Trabalhos B & C – JS Template

...\Template – Trabalhos B e C\js\main-script.js

```
/* GLOBAL VARIABLES */  
/* CREATE SCENE(S) */  
/* CREATE CAMERA(S) */  
/* CREATE LIGHT(S) */  
/* CREATE OBJECT3D(S) */  
/* CHECK COLLISIONS */  
/* HANDLE COLLISIONS */  
/* UPDATE */  
/* DISPLAY */  
/* INITIALIZE ANIMATION CYCLE */  
/* ANIMATION CYCLE */  
/* RESIZE WINDOW CALLBACK */  
/* KEY DOWN CALLBACK */  
/* KEY UP CALLBACK */
```

apoio

