U.PORTO
FEUP FACULDADE DE ENGENHARIA UNIVERSIDADE DO PORTO

Interactive Graphics Systems

Component transformations

v1.1 20221019

Alexandre Valle | DEI | FEUP

Pictures: two details of HCI from movie T*he Dark Knight (credits BLIND, Ltd.)*

Requirements

- WebGCF already references mat4 and vec3 class definitions

  https://glmatrix.net/docs/module-mat4.html

Matrix creation and configuration

```
// example of instancing a matrix object (object m)
var m = mat4.create();

//  example of setting the matrix to identity
mat4.identity(m);

// example of translation 1.0 over X axis
mat4.translate(m,m,vec3.fromValues(1,0,0));

// example of rotation around Y
var rads = 3.14159 / 2;
mat4.rotate(m,m,rads,[0,1,0]);

// example of scale to 110% of original size
mat4.scale(m,m,vec3.fromValues(1.1,1.1,1.1));
```

The main section of MyScene…

```
// Assuming this has the scope of a class extending CGFScene and root is an attribute holding the root component

// Initialize Model-View matrix as identity (no transformation
this.updateProjectionMatrix();
this.loadIdentity();

// Apply transformations corresponding to the camera position relative to the origin
this.applyViewMatrix();

// preserve the scene current matrix
this.pushMatrix()
this.drawComponent(this.root);

// restore the last preserved scene matrix
this.popMatrix()
```

drawComponent method…

```
// Assuming this has the scope of a class extending CGFScene
// assuming objects of class MyComponent have an attribute m keeping the component's transformation matrix
MyScene.prototype.drawComponent = function(currNode, …) {
            // other stuff regarding materials, textures, etc...
            ...
            // multiply the current scene transformation matrix by the current component matrix
            this.multMatrix(currNode.m);
            for(var i = 0; i < currNode.children.length ;i++) {
                        // preserve current scene transformation matrix
                        this.pushMatrix();
                        // recursively visit the next child component
                        this.drawComponent(currNode.children[i],...);
                        // restore scene transformation matrix
                        this.popMatrix();
            }
}
```

Transformations

Concept

An object requires the following transformations:

1. translation
2. Rotation

XML representation

```
<transformation…>
     <rotate …>     <!-- 2nd -->
     <translate …> <!-- 1st -->
<transformation…>
```

(transformations are to declared in reverse order)

Javascript program

Build the *mat4* object based on the order transformations are presented in XML:

```
var m = mat4.create(); // sets to identity
mat4.rotate(m,m,...);
mat4.translate(m,m,vec3.fromValues(…));
```