U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

Interactive Graphics Systems

Dealing with texture processing

Alexandre Valle | DEI | FEUP

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

Reading texture data…

```
// assuming this.reader is an object of CGFXMLreader();
// example of instancing a texture placeholder
var tex = { id: null, file:null };

tex.id = this.reader.getString(xmlElem,’id’);
tex.file = this.reader.getString(xmlElem,’file’);
```

From texture data to CGFtexture

```
// assuming this is a class extending CGFscene and tex is an object describing a texture
var texture = new CGFtexture(this, tex.file);


// preserve appearance in some array of CGFtextures
// assumed tex.id was previously set
this.textures[tex.id] = texture;
```

The main section of MyScene…

```
// Assuming this is an object extending CGFScene and this.root is the root component
// enable textures
this.enableTextures(true);


…


this.drawComponent(this.root, ..., null);
```

drawComponent method…

```
// Assuming this is an object extending CGFScene and this.root is the root component
// assuming objects of class component have an attribute m keeping the transformation matrix
MyScene.prototype.drawComponent = function(currNode, prevAppearenceId, prevTextureId) {
        // an illustration that selects an id based on some logic
        // you will have to modify this logic depending on this year work requirement!
        var texId = (currNode.textureId !== null ? currNode.textureId : prevTextureId)
        // retrieve corresponding the CGFtexture
        // TODO: check if texture exists and beware textId may be null depending on the required logic!
        for(var i = 0; i < currNode.children.length ;i++) {
                // recursively visit the next child
                this.drawComponent(currNode.children[i], …, texId);
        }
        var currTexture = (texId !== null ? this.textures[texId] : null)
        // requires the current active material
        currAppearence.setTexture(currTexture); // can be null to unset texture
        for(var i = 0; i < currNode.primitives.length ;i++) {
                currNode.primitives[i].updateTexCords(currNode.length_s, currNode.length_t); // TODO: check if function needs to be called
                currNode.primitives[i].display();
        }
}
```

Primitives (extend CGFobject)

```
/*
        Default for rectangle:
        this.texCoords = [
             0, 1, // x1, y1
             1, 1, // x2, y1
             0, 0, // x1, y2
             1, 0, // x2, y2
         ]

*/
updateTexCoords(length_s, length_t) {
     this.texCoords = /* do something with s and t. It depends on the work requirements. */
     this.updateTexCoordsGLBuffers();
 }
```

Texture mapping not required for quadric primitives, but methods need to exist (with no instructions)