Interactive Graphics - Homework 2

A.Y. 2018/2019

Marco Costa 1691388

02 June 2019

1 Introduction

This project consists in the implementation of a simplified horse, based on hierarchical model. All the components of the horse are made with cubes. The torso has a procedural texture with a checkerboard pattern and a linear decrease of intensity from the front to the back. Moreover, there is a simple obstacle, also based on hierarchical model and made with cubes. Finally, there is a button that allows to animate the horse, making it move along the x-axis and jump over the obstacle.

2 Solution

2.1 Point 1 - Create the horse

Starting from the example, the horse was built using a hierarchical model. The first element built is the torso, using the *cube()* function provided, and scaling the size along the x-axis to obtain a parallelepiped. All the other parts of the horse were built with the same principle, and were translated so as to be attached to the torso. This resulted in a simplified horse. To allow movement without the parts separating, the components of the horse have been inserted into a tree, and the torso is the root of it. The advantage of this solution is that any transformation applied to the torso is also applied to all other parts of the body, so you get a single object: the horse.

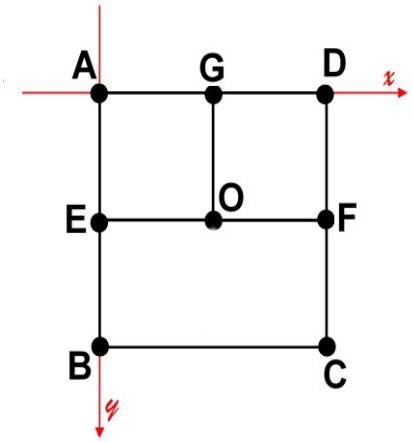
2.2 Point 2 - Apply the texture to the body

Using the *textureCube4* of *Chapter 7* of the examples of the textbook as reference, a texture with a checkerboar pattern was applied to the horse's torso. To apply the texture only to the torso, a Boolean variable has been used in the *traverse()* function (the function that creates the hierarchical model of the horse), so that it's possible to see in the *shader* which part of the horse the texture is being applied to. To apply correctly the linear decrease of intensity from the front to the back of the body,it was necessary to apply different textures to the

various faces of the torso. Since only one texture is passed to the shader anyway, this texture has been divided into 3 so it was possible to put different textures. This texture has been implemented as follows:

```
var linearDecrease = new Uint8Array(4*texSize*texSize);
for (var i = 0; i < texSize/2; i++) {
    //Other 4 faces
    for(var j = 0; j < texSize; j++) {
        linearDecrease[4*i*texSize+4*j] = j;
        linearDecrease[4*i*texSize+4*j+1] = j;
        linearDecrease[4*i*texSize+4*j+2] = j;
        linearDecrease[4*i*texSize+4*j+3] = 255;
    }
}
for (var i = texSize/2; i < texSize; i++) {
    //Dark
    for (var j = 0; j < texSize/2; j++) {
        linearDecrease[4*i*texSize+4*j] = 16;
        linearDecrease[4*i*texSize+4*j+1] = 16;
        linearDecrease[4*i*texSize+4*j+2] = 16;
        linearDecrease[4*i*texSize+4*j+3] = 255;
    }
    //Intense
    for (var j = texSize/2; j < texSize; j++) {
        linearDecrease[4*i*texSize+4*j] = 255;
        linearDecrease[4*i*texSize+4*j+1] = 255;
        linearDecrease[4*i*texSize+4*j+2] = 255;
        linearDecrease[4*i*texSize+4*j+3] = 255;
    }
}
```

It's possible to schematize the texture as follows:



Points coordinates:

- **A (0, 0);**
- **B (0, 1);**
- **C (1, 1);**
- **D (1, 0);**
- **E (0, 0.5);**
- **F (0.5, 1);**
- **G (1, 0.5);**
- **O (0.5, 0.5);**

- The figure *AEOG* identifies a very dark texture and was applied on the back face of the torso.

- The figure *GOFD* identifies a very intense texture and was applied on the front face of the torso.
- The figure *EBCF* identifies a linear decreasing texture and was applied on the other 4 faces of the torso.

To apply each texture to the correct face, other coordinates have been passed to the *shader*, different from the previous ones.

```
var texLinearCoord = [
    //Lat Dx
    vec2(0.0,  0.5),
    vec2(0.0,  1.0),
    vec2(1.0,  1.0),
    vec2(1.0,  0.5),

    //Front
    vec2(0.5,  0.0),
    vec2(0.5,  0.5),
    vec2(1.0,  0.5),
    vec2(1.0,  0.0),

    //Down
    vec2(0.0,  0.5),
    vec2(0.0,  1.0),
    vec2(1.0,  1.0),
    vec2(1.0,  0.5),

    //Up
    vec2(0.0,  0.5),
    vec2(0.0,  1.0),
    vec2(1.0,  1.0),
    vec2(1.0,  0.5),

    //Lat sx
    vec2(0.0,  0.5),
    vec2(0.0,  1.0),
    vec2(1.0,  1.0),
    vec2(1.0,  0.5),

    //Back
    vec2(0.0,  0.0),
    vec2(0.0,  0.5),
    vec2(0.5,  0.5),
    vec2(0.5,  0.0)
];
```

2.3 Point 3 - Create the obstacle

The obstacle was created with the same principle as the horse. So there is a second object, independent of the first, as it is created in a different tree. The horse has been moved back along the x-axis in order to have more space for animation.

2.4 Point 4 - Add animation

When the *START* button is pressed, the horse starts galloping. As he approaches the obstacle, he gets ready and then jumps. When it lands, it starts galloping again. After that, the horse returns at the start point. In order to

perform the animation, there is the *animate* flag set to false. This flag is set on true when the *START* button is pressed. In the *render()* function, if the flag is true, the *performAnimation()* function is performed. This last function lets the horse to move along the x-axis, move along the y-axis when it's time to jump and move all legs. In order to move the legs I used the *interpolate()* function, which allows to move them more naturally. When the horse exceeds a certain value of displacement on the x-axis, it is returned to the initial position, the flag *animate* is set to false and all values are reset to dafault values.
It's possible to pause the animation with the *PAUSE* button.
Since the function *animate()* has many lines of code, I preferred not to attach it.

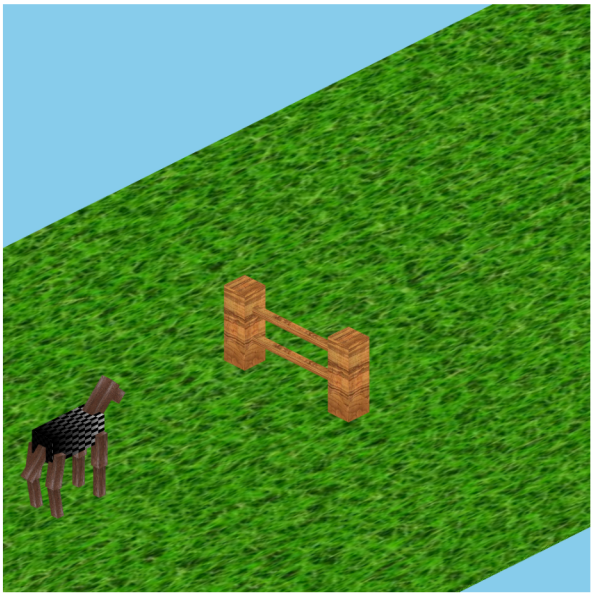
2.5 Extra

In order to see the horse in 3D and to make the animation well visible, I added a viewer position and I used the ortographical projection. It's possible to change the view with the *CHANGE VIEW* button. To make everything more realistic, different textures have been applied:

- A texture for the horse, so as to simulate its skin;
- A texture for the obstacle, so it looks like a wood one;
- A texture for the ground, so as to simulate the grass;

Moreover, CSS has been added to modify the buttons' graphics. Finally I decided to apply a white texture only to the torso to make more visible the linear decrease.

3 Final result


START
CHANGE VIEW