HOMEWORK 2 – INTERACTIVE GRAPHICS

Denise Landini – 1938388

NOTE: THE PROJECT WORKS ONLY IN GOOGLE CHROOME.

1. **Create a hierarchical model of a (simplified) sheep, composed of the following parts: body, 3 legs, each one composed of 2 independent components (upper and lower leg), head and tail. All components are cubes, use the cube function present in the file. The sheep has a white/light grey color.**

   The sheep is represented with a hierarchical structure. The structure is a "tree" in which the root and then the other nodes can have sibling and child.

   I use a hierarchical structure because is more practical to do the animation with the sheep.
   I have just to work with the root of the hierarchy (torso) to move all the component of the sheep (i.e., walking on or walking back).

   The hierarchical structure that I build is as follows:

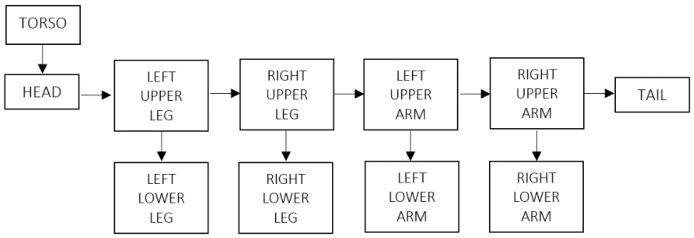

Figure 1: Hierarchical structure of sheep

2. **Add a surface on which you position the sheep that corresponds to a grass field. Attach to it a texture (color, bump or both) to give the appearance of a grass field.**

   I decide to use only one node for the grass field, and I create only one face of the cube to obtain a surface as required.
   In order to give the appearance of a grass field I add a bump texture image.
   The vector texCoord that I use is the following:
   `var texCoord = [vec2(0, 0), vec2(0, 1), vec2(1, 1), vec2(1, 0)].`
   In the function `grass()` I set the variable `textureGrass = true` and all the other texture flags to false. In this way, I assigned the exact texture to the grass field.
   As requires, both sheep and fence are positioned on the grass surface.

   I decide to create another independent hierarchy for the sun composed only by a yellow cube.
   This is needed in order to work with the bump map and this is the source of light of my environment.

   In the html file, I decide to implement a checkbox to disable the bump map. I do this choice (even if it is not required) only to highlight the difference between the scene with the bump map and the scene without it.

Figure 2: Scene without bump map

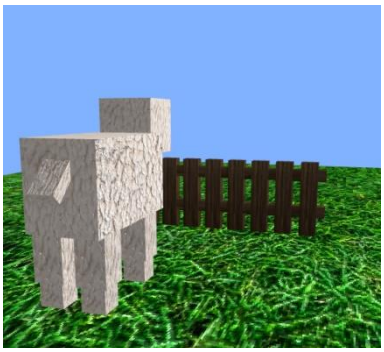

Figure 3: Scene with bump map

3. Load or generate at least two more textures. A color texture to be attached to the front face of the head and a bump texture to be applied to the sides of the body to give the "wool effect".

   For sheep I use 2 different textures and all these two textures are added using normal map.
   For the management of the textures, I used the flags.
   In particular, I used:

   - the texture the gives the wool effect for all the body and a part of the head as we can see in Figure 3. In this case I use the vector texCoordBody for the body of the sheep.
     `var texCoordBody = [vec2(0, 0), vec2(0, 0.5), vec2(0.5, 0.5), vec2(0.5, 0)].`

     I use the vector texCoordLeg for all the components of the leg and the tail.
     `var texCoordLeg = [vec2(0, 0), vec2(0, 0.25), vec2(0.25, 0.25), vec2(0.25, 0)].`

     As required, I applied a bump map for all the body of the sheep to give the wool effect.

   - a different texture to the front of the head as required. In fact, in order to have a realistic sheep with eyes and mouth, I decide to apply a texture with a fantastic face of the sheep. For this I use the same texCoord that I used for the grass.
     In this case I did not apply the bump map, but I simply use a texture image.

Figure 4: *Texture image attached to the face*

4. Create a (very simplified) model of a fence and position it on the surface and near the sheep.

   The fence is represented by another hierarchical structure, so it is completely independent from the sheep. In particular, the fence is composed by 11 vertical parallelepipeds and 2 horizontal parallelepipeds that are all siblings. The fence is clearly visible in Figure 4.

   The hierarchical structure that I build is as follows:

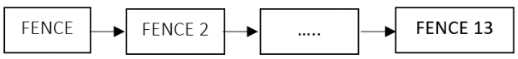

Figure 5*: Hierarchical structure of fence*

   To make the fence more realistic, I assign it a texture image that gives the appearance of the wood.
   The vector texCoord that I use is the following:
   `var texCoord = [vec2(0, 0), vec2(0, 1), vec2(1, 1), vec2(1, 0)].`
   In all the function referred to the fence I set `textureFence = true` and all the other texture flags to false. In this way, I assigned the exact texture to the fence.

5. Add a button that starts an animation of the sheep so that, starting from an initial position where it is in a walking mode, it walks on the surface towards the fence by moving (alternatively back and forth) the legs, then jumps over the fence and lands on the surface on the other side of the fence.

   The sheep is in an initial position that is given by the variable
   `posTorso = [-12.0, 1.4, 0].`

The animation of the sheep that moving near the fence and then jump it is done by the function `go_jump()` that I have implemented by using the switch case structure.
In particular, I have followed these steps:

1. Walk: to allow the sheep to move alternatively the right/left arms and legs, I rotate them, by changing the theta angle of the right/left arms and legs that I built in the hierarchical structure and, at the same time, I increase the X values of the position of the Torso to allow the sheep to advance towards the fence, but the Y values remain the same because it is just going on.
2. Jump: to allow the sheep to jump, I used a semi-circumference to increase either the X and Y values of the position of the Torso because, in this way, it is easy for the sheep to overcome the fence. In this step, the legs and arms of the sheep reach the maximum possible opening and I do this always by rotating the theta angle of the right/left arms and legs that I built in the hierarchical structure.
   When the sheep has to go down from the jump, the Y value of the Torso decrease because it is going down, but the X value continues to increase.
3. Turn itself: to allow the sheep to return on the initial position (even if it is not required), I decided to turn the sheep on itself by increasing the theta angle of the Torso until it is in front of the fence.
4. I follow these 3 steps also to allow the sheep to jump back the fence. In this case the only thing that change is the changing of the X values that, instead of increasing it, I decrease it because the sheep is going back. In Figure 6 there are the back steps.

In the html, I created two buttons to start and reset the animation.

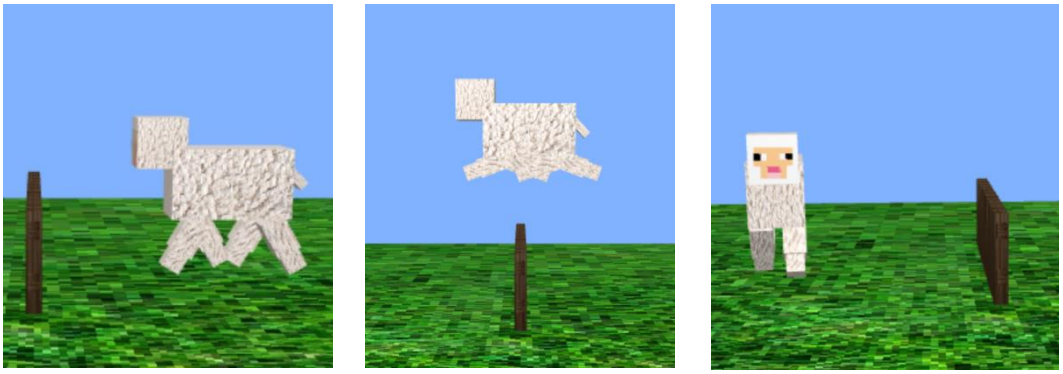

***Figure 6**: From left to right: Sheep walk on, Sheep jumps, Sheep turns itself.*

6. **Allow the user to move the camera before and during the animation.**

In order to allow the user to move the camera before and during the animation I add the sliders to change the modelView values. The range are whose that are set in the table.

	Min	Max	Current value
Parameters used for viewer position:			
Radius: represents the distance from the origin.	1.0	100.0	45.0
Theta: used to change the camera view.	-180	180	10
Phi: used to change the camera view.	-180	180	-40
Parameters used for perspective projection:			
Fovy: how wide the eyes open along y.	10	50	45.0
Aspect	0.5	2.0	1.0
Near: the near plane distance from camera for the jar.	0.1	2.5	0.4
Far: the far plane distance from camera for the jar.	1.0	2000.0	2000.0

The advantage of using the two parameters Theta and Phi in that interval is to have a global view of the sheep (360 degress). The radius and all the other parameters are set to have all the objects clearly visible at the beginning, but we can change these to obtain all the other situation that we want, in fact, the range values are large enough.