

HOMEWORK 2

Emanuele Bonifazi - Mat. 1815292

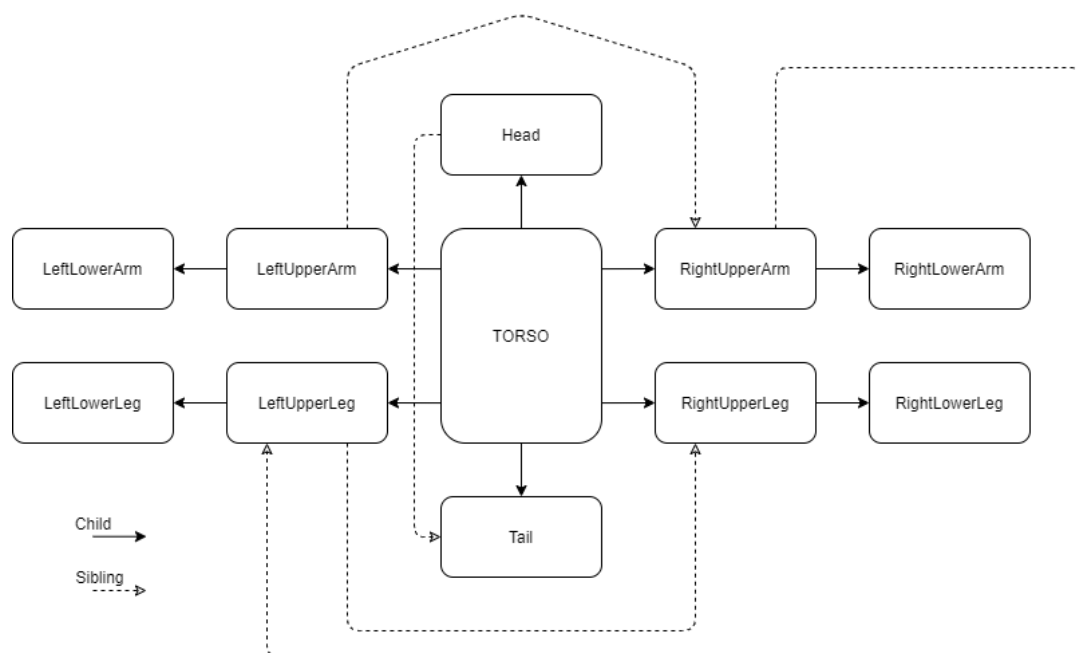
Scene

The proposed scene represents a sheep that jump over a wood fence. These two objects are positioned over a grass.

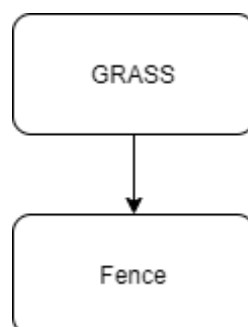
Models

In order to obtain the above scene, we use two different hierarchical model, one to build the grass+fence model and the second to build the sheep. To clearly represent the two models, in the following there are the graph of the models that show links between parts of the model. There are two different kinds of links: sibling link and child link.

Sheep



Grass + Fence



All the components showed in the scene are cubes that are modified by functions to obtain the desired aspect. An example of the function that modify the aspect of the cube to look like tail is:

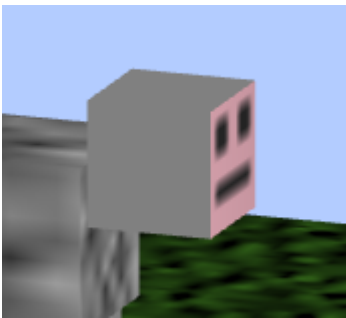
```
505 function tail() {  
506     instanceMatrix = mult(modelViewMatrix, translate(0.0, 0.5 * tailHeight, 0.0));  
507     instanceMatrix = mult(instanceMatrix, scale(tailWidth, tailHeight, tailWidth));  
508     gl.uniformMatrix4fv(modelViewMatrixLoc, false, flatten(instanceMatrix));  
509     for (var i = 0; i < 6; i++) gl.drawArrays(gl.TRIANGLE_FAN, 4 * i, 4);  
510 }
```

It is possible to see that this function scales the size of the cube and draws it in the scene.

Textures

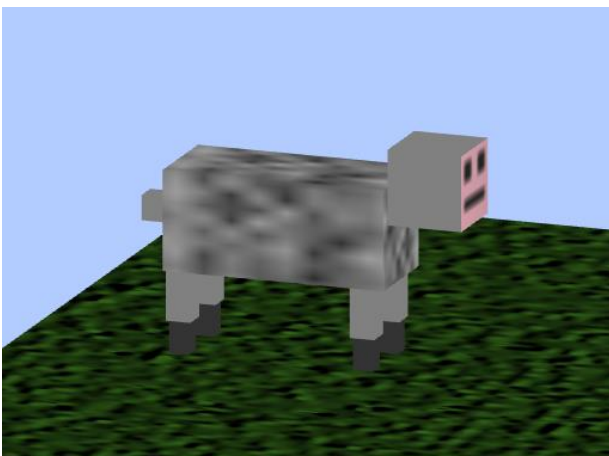
There are present two different kind of textures applied over some parts of the scene the first is a colour texture used to colour the face of the sheep and the second is a bump texture used to obtain grass and wood texture.

Colour texture



This texture is used to obtain the face of the sheep and basically it consists in a matrix of pink and black colour where the black colour is used to draw eyes and mouth.

Bump texture



In this picture it is possible to see the two bump texture used to obtain grass and wool effects. The idea is to generate a rougher surface for the grass and a smooth surface for the wool.

The two function used to obtain these textures are the following:

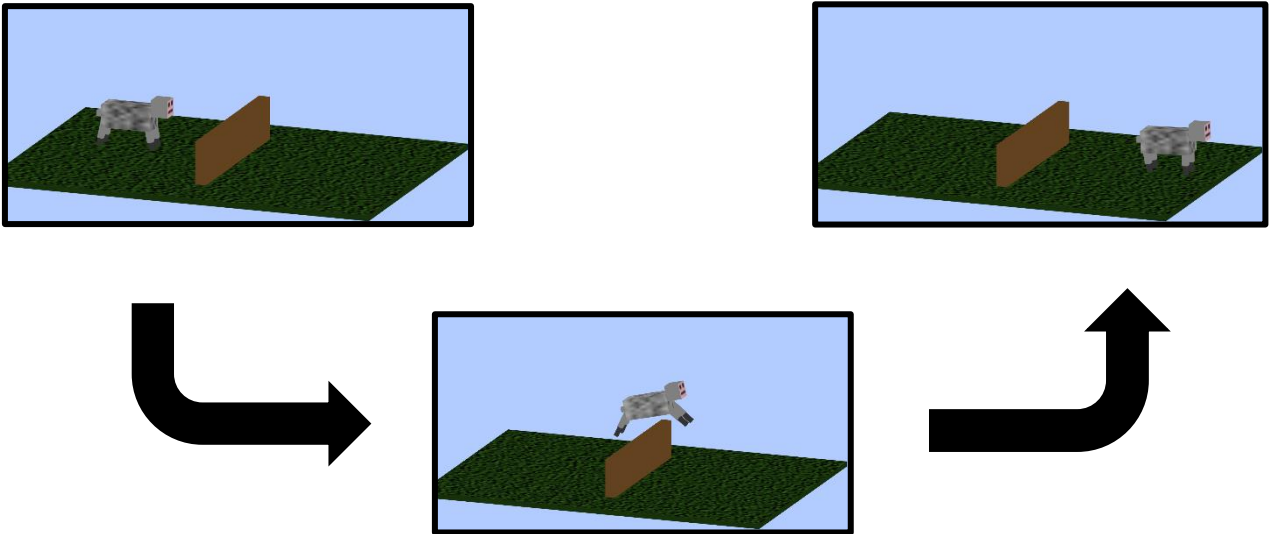
```
132 data[i][j] = Math.sin(Math.random()*10)*Math.log10(Math.random());
```

 → Used for grass

```
185 data2[i][j] = Math.random()*0.5;
```

 → Used for wool

Animations



Animations are provided by multiply the matrix that represent the instance by a transformation matrix that can be for example a rotation matrix or a translation matrix, it depends on what we want to obtain.

To clarify the used method this is an example of transformation:

```
725 figure[torsoId].transform = mult(figure[torsoId].transform, translate(0.0,0.05,0));
```

In this case the matrix stored in `figure[torsoId].transform` is modified by multiplying itself for a translation matrix. We obtain that the torso is translated forward of 0.05 units.

The important aspect is that due the hierarchical structure of the model if I translate the torso all the child translate with it.

This is very important because simplifies the animation building. Without this structure we should translate every single part on its own.

The animation is put in loop to obtain sequential jumps over the fence.

It is also possible to stop the animation at every moment by pushing a button.