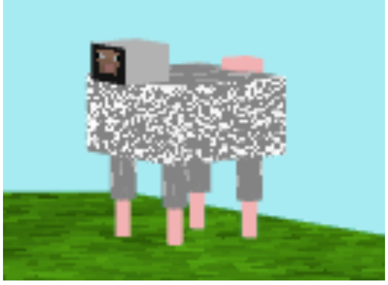
1. Create a hierarchical model of a (simplified) sheep https://en.wikipedia.org/wiki/Sheep, composed of the following parts;

a. body
b. 4 legs, each one composed of 2 independent components (upper and lower leg)
c. head
d. tail

All components are cubes, use the cube function present in the file. The sheep has a white/light grey color.

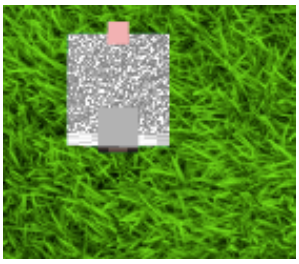


Starting from the initial model given in the files, the body, arms and legs have been rotated and adjusted to resemble the shape of a simplified sheep. The adjustments have been made in the cases of the function createNode with the auxiliary functions translate and rotate.

The torso of the sheep and its upper legs and arms have been colored in a grey shade, while the lower legs and arms have been colored in pink to distinguish better the different parts of the legs while keeping a realistic color.

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.

The surface to create the grass field have been created using a cube with dimension 20 x 20 x 0.1. The corresponding node is created with no siblings nor parents in the hierarchical model so in the rendering must be called its creation independently.

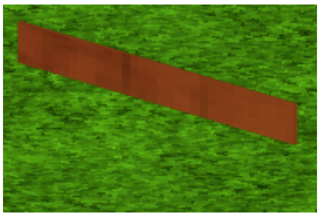


In the portion of code of createNode corresponding to the creation of the grass field, it’s activated a color texture to give it a rough appearance, along with the image of a real grass to make it even more realistic.

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”.

As shown in the first image of the sheep, it has a bump texture attached to its torso and upper legs and arms. The head is texturized with a color texture along with an image (the muzzle of Minecraft sheep since its shape is very similar to the cube-based simplified sheep modeled in this homework) to make it look more realistic.

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



It has been created with a cube of dimensions 0.1 x 10 x 1 and placed at the middle of the grass field, in front of the sheep. To make also this element a little bit more realistic, a color texture and an image of wood have been attached to it.

In order to load correctly all the different textures of the elements, the fragment shader have been modeled in order to have an adaptive behavior based on the use of flags. Those flags have the purpose to activate the correct behavior of the shader based on which component of the scene has to be rendered.

```
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 walk of the sheep have been modeled subdividing it in two main steps: the first step is when the sheep moves the legs from the "resting" position and the second step is when it puts back the legs in the initial resting position.

-**step 1**: the legs are moved of a certain angle from the resting position until the final position is reached. The increase of angles at every iteration is 5 degrees.

-**step 2**: all the legs are just moved decreasing the angles which have been increased in step 1, but now to start again the process in a specular way to make it look realistic, the left upper legs/arms are shifted with the right upper legs/arms respectively.

In order to code this, the ids of the legs have been saved in a separate array modified every time the sheep comes back in a "resting" position.

Before starting the jump, the angles of the legs and arms are stored and reset to the initial values to make the sheep jump in a smooth way.

The jump has been segmented in four main steps: the first step where the sheep reaches the fence and stops to lift the arms preparing the jump, the second step where the sheep pushes itself to start jumping and reaches the area over the fence, a third phase where the sheep leans towards the ground again to touch the grass, and finally a fourth phase where the sheep puts down its legs and starts walking again.

All the steps move the sheep in little increases and decreases of angles of the different parts of the body in order to make the animation as smooth as possible.

After the jump, since the angles of the legs before the jump have been saved, the sheep resumes its walking starting from the saved angles. If the animation is not reset, when the sheep reaches the left border of the screen it comes back walking from the right side, ready to jump again cyclically until the user resets.

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

This feature has been added implementing the modelViewMatrix as lookAt(eye, at, up), allowing the camera to be moved in every moment.

The projection matrix has been changed to a perspective projection allowing also to zoom in and out in the scene.