Report Homework 2

Gioele Migno - 1795826

6 June 2021

1 Geometric Scene

1.1 Sheep

The sheep is modeled using an hierarchical model composed by: Torso, Head, Tail and the legs. The legs are divided into front and back legs, each of them are formed by upper and lower leg. The root of the hierarchical structure is Torso, in Fig.1 is shown the hierarchical model.

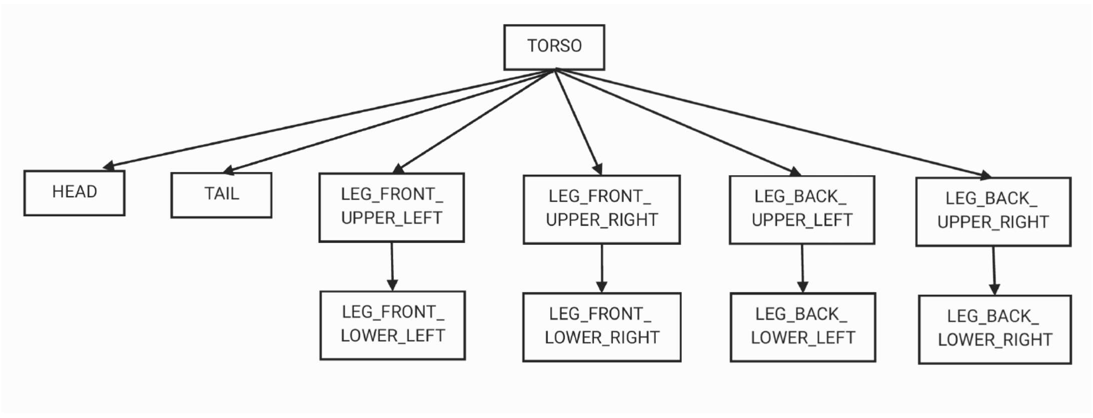

Figure 1: Hierarchical Model of Sheep

The pose and the position of the sheep are represented by its state that is coded inside the following variable:

```
var state = {
    // TORSO --------------
    torso_RX: 0, // rotation around x-axis
    torso_RY: 0, // rotation around y-axis
    torso_RZ: 0, // rotation around z-axis

    torso_TX: 0, // translation x
    torso_TY: 0, // translation y
    torso_TZ: 0, // translation z
    // --------------------

    // HEAD ---------------
    head_RX: 0, // rotation around x-axis
    head_RY: 0, // rotation around y-axis
    //---------------------

    // FRONT LEGS -------------------
    leg_front_left_upper: 0, // rotation around x-axis
    leg_front_left_lower: 0, // rotation around x-axis

    leg_front_right_upper: 0, // rotation around x-axis
    leg_front_right_lower: 0, // rotation around x-axis
    // ------------------------------

    // BACK LEGS ---------------------
    leg_back_left_upper: 0, // rotation around x-axis
    leg_back_left_lower: 0, // rotation around x-axis

    leg_back_right_upper: 0, // rotation around x-axis
    leg_back_right_lower: 0, // rotation around x-axis
    // -------------------------------

    // TAIL --------------------------
    tail: 0 // rotation around x-axis
    // -------------------------------
}
```

1.2 Fence

The fence is rendered using several vertical pickets with size 1x7x0.5 and placed with a distance of 3 from each other and also using two horizontal stakes. The fence is then placed on a line parallel to x-axis passing through z=-6.

2 Animation

The scene starts with the sheep at rest in z=25, then clicking on the button START the animation starts and the sheep enters in walk mode, the key points of the walk, considering a single leg, are shown in Fig.2. In horizontal axis are reported the timestamps of each pose expressed in seconds.

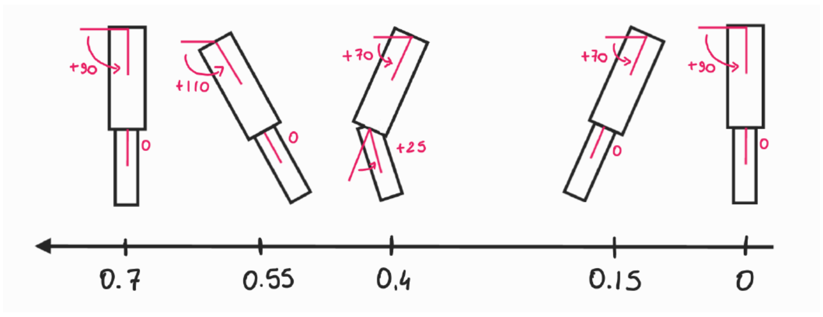

Figure 2: Key points of walk animation of a leg

Considering the front legs (right/left), they use the same splines but phased-shifted, namely:

```
var walk_timestamp = [0, 0.15, 0.4, 0.55, 0.70];

// right leg
var step_upper_leg_dx = [90, 70, 70, 110, 90];
var step_lower_leg_dx = [0, 0, 25, 0, 0];

// left leg
var step_upper_leg_sx = [70, 90, 90, 110, 70];
var step_lower_leg_sx = [0, 0, 0, 0, 25];
```

Then the back legs use the same couple of splines but with left and right swapped.

For all the animation, the torso moves by changing linearly respect to time the z-position $z(t) = START\_Z - t/100$ where $START\_Z = 25$ and t represents the milliseconds elapsed since the beginning of the animation.
When the sheep's torso reaches z=5, it starts to jump, so the y-position must change, the key y-position and the related timestamps are: step_jump = [0, 7, 12, 7, 0], jump_timestamp = [0, 0.6, 1.2, 1.8, 2.4]. Once reached z=-19, the sheep ends the jump and restarts to walk and it stops at the end of the land.

2.1 Splines

In order to animate the scene using splines function, two libraries have been developed, the first is called Matrix.js and contains the essential utilities to work with matrices, the second instead, Splines.js implements the algorithm necessary to compute the coefficients of the chain of cubic polynomials. Spline.js library is based on the technique shown in Robotics 1 course and guarantees the continuity up to acceleration between internal knots. To use Spline.js is necessary to build a new Splines object with the constructor $Splines(\_array\_position, \_array\_time\_step, \_v\_start, \_v\_end)$ and then run $Splines.init()$ that using the internal function $spline\_resolver()$ computes the coefficients of the cubic polynomials. Finally the values of the splines function can be obtained using $Splines.step(time)$ where $time$ must belong to the time intervals defined by _array_time_step.

```
// Example
spline_upper_leg_dx = new Splines(step_upper_leg_dx, walk_timestamp, 0, 0);
spline_upper_leg_dx.init();
```

3 Textures

In the scene two image textures have been used, $TEXTURE0$ is applied on the land and it represents a grass field. $TEXTURE1$ is used instead for the sheep's face, the image chosen is the sheep's face of Minecraft.

3.1 Bump texture

In addition to the two previous image textures, a bump texture is applied on the sheep's body in order to give to it a wool effect. The code used to generate the bump texture is based on the book's example of Honolulu in which the altitude has been replaced with random values in [0, 500], moreover, the tangents and the normals to the object's surface have been generalized by computing the last as the cross product between the edges of the squared, and the tangent instead, as the vector lying on a edge of the square.

4 Screenshots

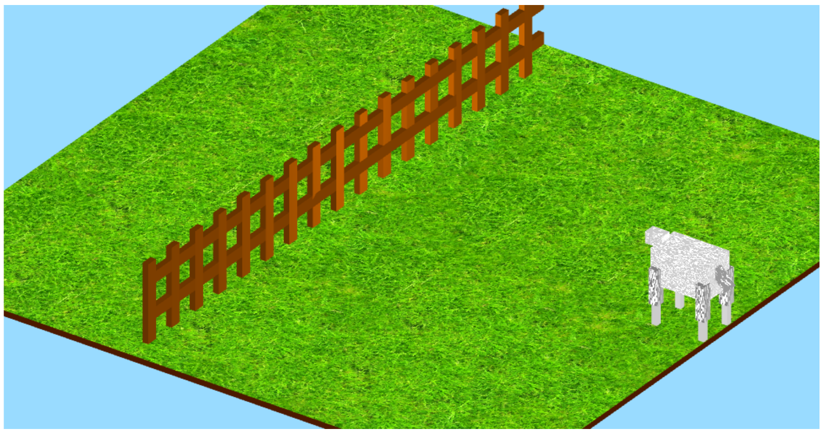

Figure 3: Screenshot of the scene

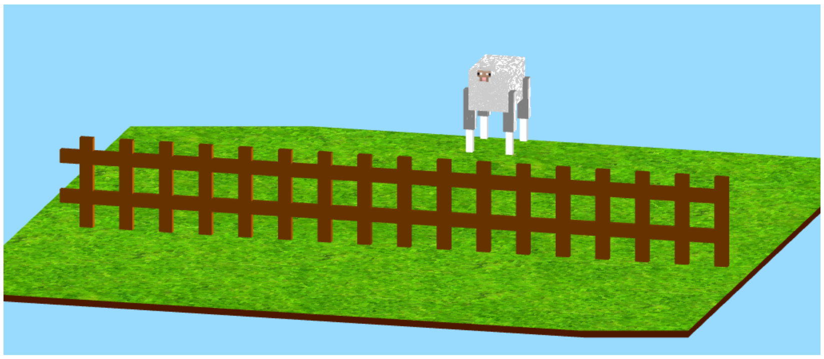

Figure 4: Screenshot of the scene

As it can be seen from the previous screenshots, the lighting is applied to all objects in the scene, the light position is at (0, 2, 0). Considering the material of the objects inside the scene, the specular component of the light has not been used.

5 Browser compatibility

The code has been tested in Ubuntu 18.04 using a python3 server (*$ python3 -m http.server*), Google Chrome v.90.0.4430.212 and Firefox v.88.1.