INTERACTIVE GRAPHICS

FINAL PROJECT REPORT

A.Y. 2021/2022

FINAL PROJECT REPORT

Valentina Piccione

piccione.2030930@studenti.uniroma1.it

July 17, 2022

INTRODUCTION

This document is the report for the Final Project of the Interactive Graphic course, held by Professor Marco Schaerf, from the Artificial Intelligence & Robotics MD course at the University of Rome Sapienza.

1 Topic and Project Requirement

The topic of the project is a 3D representation of a room in which my animals are shown. In the scene there are Luna, my cat, Cannella, my mouse and Adi, my fish. All the animals displayed are animated. The user can interact with the cat using the mouse moving around a toy which Luna will follow. The others instead performs fixed animations moving in their space.

All project requirements has been satisfied:

- **Hierarchical Model:** All the animals and decorations in the scene use complex hierarchical models directly implemented in JavaScript.
- **Lights and Textures:** In the scene there are two different lights: an HemisphereLight, and two SpotLights one for the scene and one for the aquarium light. Moreover there are different types of material used for the objects in the scene and image textures applied to the floor, the scratching post, the sand of the aquarium and the bowls.
- **User interaction:** The user can interact with the scene changing the view, zooming and rotating using the mouse, and can interact with the cat also using the mouse, since it represents a toy that the cat follows and try to catch. The user can also change the position of the cat and the toy used for the interaction with buttons in the scene.
- **Animations:** As required most of the objects have animations implemented in the JavaScript.

The project is available as requested in a GitHub repository, at *this web address*.

2 Libraries Used

The project has been implemented using Three.js library. In addiction the following modules/libraries have been used:

- **OrbitControls.js** Module used to move the camera using the mouse.
- **Gsap.js** Library used to implement smooth animations in JavaScript.

- **GLTFLoader.js** Module used to load the models of the *wheel* and the *petpillow*, in .gltf format, created using the software Blender.

3 Environment

3.1 Scene

The model of the room has been implemented in JavaScript using a PlaneGeometry geometry. All the planes can be seen just from the inside the room, to allow the user to watch the scene from every angle.

Everything inside the room has been implemented in JavaScript composed by its own hierarchical model, except for the wheel which has been designed using Blender 2.93 and then imported in the project.

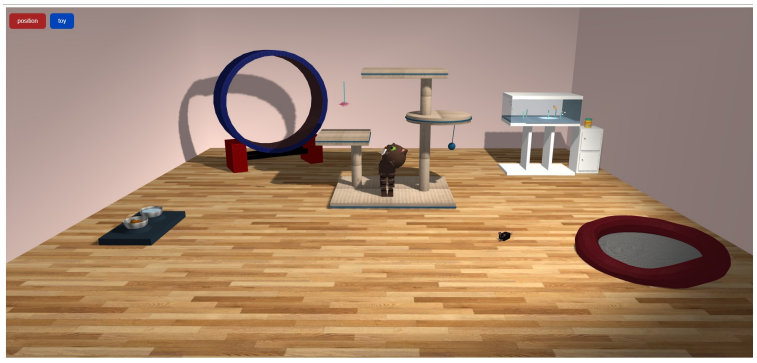

Figure 1: the scene

3.2 Lighting

In the room there are two Spotlights: the first illuminates the entire room, and is located at the point (250, 480, 250) of the scene, directed to (0,0,0). The second is located at the point (360, 155, -240) as to simulate the aquarium light.

In the scene there is also an HemisphereLight positioned directly above the scene, with color fading from the sky color to the ground color. In fig.1 the lights are clearly visible.

3.3 Materials

Almost all objects in the scene use MeshLambertMaterial without specular highlights. The glass of the aquarium uses MeshPhongMaterial in order to simulate shiny surfaces with specular highlights.

3.4 Textures

The textures in the scene are applied to different objects in order to make the scene look more realistic: the floor implements a parquet texture, the scratching post implement a twine texture, the bowls implement a metal texture, the sand of the aquarium implements a gravel texture and the fish food has a label on it. They are all image textures.

4 Elements in the environment

4.1 Cat

The hero of the project is the cat standing in the center of the scene. It has a very articulated hierarchical structure, and the user can switch between two positions of the cat by clicking a

button present in the simulation: standing cat and sitting cat. It has several animations like the eyes that blink randomly and the mouth which opens and meows randomly (I recorded my cat and the sound has been added to the code). Also the tail moves on a continuous animation with a sinusoidal movement.

The most important animations of the cat are the head that follows the user's mouse movement and the legs that try to reach the toy when in range.

In fact the cat stares continuously at the user's mouse and when it comes closer than a particular value defined by the variable $isInDistance$, tries to reach it. The attack is divided in two phases: the $prepareAttack$ phase where the cat just lift the leg, and the $attack$ phase where it extends leg and paw and touches the toy.

```
Sitting_Luna.prototype.prepareAttack = function(side, v){

    var angles = getAngles(v, this.rightShoulder.position, this.armHeight);
    this.updateArm(side, angles, 1, "back.easeOut", null);
}

Sitting_Luna.prototype.attack = function(side, v, direction){
    _this = this;
    var shoulder = (side == "right")? this.rightShoulder : this.leftShoulder;
    var angles = getAngles(v, shoulder.position, this.armHeight);
    this.updateArm(side, angles, .15, "back.easeOut", function () {
      var isInDistance = direction.length()<(_this.armHeight*2 + _this.handHeight+20);
      if (isInDistance)
        _this.transferPower = { x:-direction.x*(Math.random()*.5)-.1+Math.random()*.2,
                                y:-direction.y*Math.random()*.5 };
      _this.isAttacking = false;
    });
}
```

The cat attacks always with just one foot and replaces the foot when the previous attack has ended (for example when the mouse is no more in range and the foot used is going back to the ground). If the mouse stays in range the cat continues to attack with random intensity.

Figure 2: the cat in the scene

The movement of the legs in during the attack and the replacement of the foot is animated using the $back.easeOut$ movement, which is an interpolation function that allows a realistic smooth movement of the legs, following the graph in fig3:

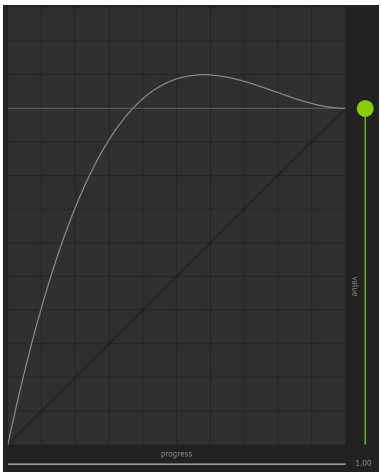

Figure 3: the curve of the easeOut movement

4.2 Fish

In the right side of the room there is the aquarium with the fish. The whole structure is a complex hierarchical model with animations inside of it.
The fish is animated in a way that it moves on a circular path of radius equal to 0.6 on the (x, z) plane and with a sinusoidal movement on the y axis. The tail rotates continuously on its y axis with a sinusoidal movement.

Figure 4: the aquarium

```
var tail_ang = 0;
var k = .5;
Fish.prototype.updateTail = function() {
  tail_ang += .1;
  var rot = Math.cos(tail_ang)*k;
  this.tail.rotation.y = rot;
}

var r = 0.6;
var ang = 120* Math.PI/180.0;
var y_mov;

Fish.prototype.moveAround = function() {
  y_mov = r * Math.cos(ang);
  ang -= 0.025;
```

```
  this.goldFish.position.y += y_mov*0.25;
  this.goldFish.rotateY(Math.PI/220);
}
```

In the aquarium also the bubbles are animated. The two blocks of bubbles are moving in two different ways: the right side block moves along the x and the y axes, while the left side block moves along the z and y axes. They have been created like a concatenation of bubbles and their motion is similar to an elastic up/down and right/left movement.

4.3 Mouse

The last animal in the project is the mouse, composed by a less complex hierarchical model than the cat. The animation of the mouse is similar to the fish one, it moves on a circumference of radius 6 around the scratching post, at the center of the room. The difference is that the movement of the mouse on the path depends on a random variable n: it starts and stop moving randomly.

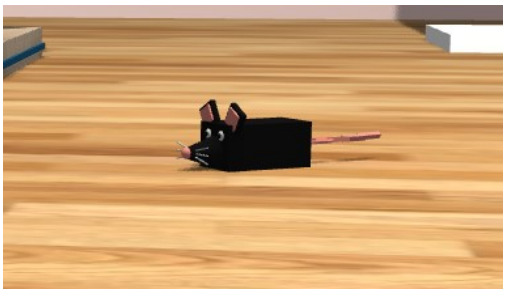

Figure 5: the mouse

The tail of the mouse is also animated in a similar way of the cat's, but with much smaller movements, visible in particular when the mouse stops.

```
Mouse.prototype.updateTail = function(t) {
    for (var i=0; i<this.tailNSegs; i++){
      var angleStep = -i*.3;
      var angleAmp = Math.PI/(600/(i+1));
      var rotZ = Math.cos(t+angleStep)*1.7*angleAmp;
      var st = this.tailSegments[i];
      st.rotation.z = rotZ;
    }
}

var r = 6;
var ang =  Math.PI/180.0;
var x_mov;
var z_mov;
var isMoving = true;
Mouse.prototype.moveAround = function() {
    var n = Math.random();
    if (n > .98) { isMoving = !isMoving; }
    if(isMoving) {
        x_mov = r * Math.cos(ang);
        z_mov = r * Math.sin(ang);
        ang += 0.025;
        this.body.position.x += x_mov;
        this.body.position.z += z_mov;
        this.body.rotateY(-Math.PI/125.5);
    }
    else { return; }
}
```

4.4 Toys

The toys are models that will follow the user's mouse pointer and will allow the cat to interact with them. Two toys are implemented: a ball and a feather. While the ball is implemented as a sphere dangling from a string, the feather is a hierarchical model, computed directly in the JavaScript source.

Both toys will move in a defined plane, and will not exceed at any time the boundaries of the room. Both toys are dangling from a string, and a wind-like effect is applied to them, making them rotate on their axis when the user moves the mouse.

When the cat tries to interact with one of the toys, a slight pull effect is applied on the cat's paw, making it slightly attract the toy towards the cat, making the interaction more visible and real.

Figure 6: the two toys

4.5 Decorations

In the room also other elements are present, not animated, but used as details for the scene.

4.5.1 Bowl and Scratching Post

The *bowls* on the left and the *scratchingpost* where the cat is standing are both hierarchical models, computed directly in the JavaScript.
Both implements a texture: the metal part of the bowls is a metal image texture, and the scratching post has a twine texture. The ball of the scratching post is the only animated thing in the decorations, it moves like the string attached to the level of the scratching post is an elastic.

4.5.2 Wheel and Pet Pillow

The last two elements in the scene are the *wheel* and the *petpillow*. Unlike the other elements they have been designed using Blender 2.93 and imported in the project.

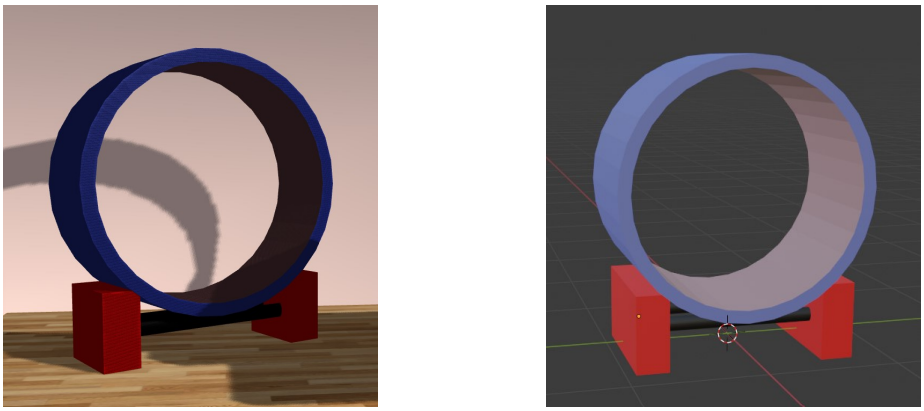

Figure 7: the wheel

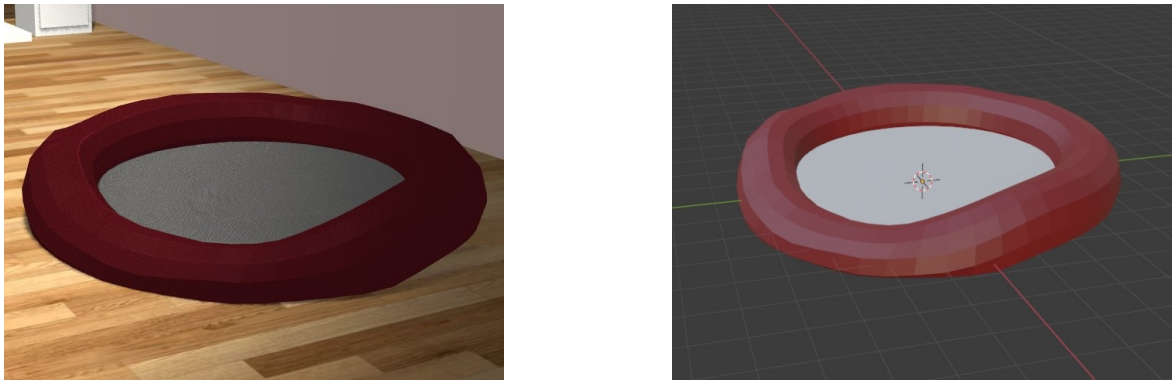

Figure 8: the pet pillow

5 User Interaction

5.1 Mouse Interaction

The user can move the mouse and play with the cat, moving around the toy in such a way that the cat will follow it with the head and try to catch it when in range.

5.2 Changing Position and Toys

As previously mentioned, the user can change the position of the cat using the red button *position* in the upper left corner of the screen in such a way as to have the cat standing or the cat sitting. The user can also change the toy attached to the mouse with the blue button *toy* in the upper left corner of the screen: it can be a $feather$, or a $ball$.

5.3 Change View

The user can also move the camera view by pressing the left button of the mouse and moving it. Moreover can zoom-in or zoom-out using the mouse washer.

6 The Models

Figure 9: Luna and Cannella

Figure 10: Adi