Homework 2 - Interactive Graphics

Veronica Romano - 1580844

June 13, 2020

1 Introduction

To realize this homework I had to start from the code provided by the professor and later to modify it to realize my personal framework and animation. In particular the tasks that I had to implement are the following:

1. Create a hierarchical model of a (simplified) Grizzly bear formed by a body, 4 legs (upper and lower part), head and tail.
2. Add a texture to the all the faces of the bear, except the head. It has a separate texture.
3. Create a very simplified model of a tree and position it near the bear.
4. Add a button to start an animation of the bear so that, starting from an initial position where it is in a walking mode, it walks towards the tree by moving the legs, then stands up and starts scratching its back against the tree.

2 Implementations of Tasks

First of all I need to create the framework with the bear and the tree and then I implemented the main task which is the animation. I will explain all the tasks separately, to illustrate how I reached the final result.

2.1 Creation of the Bear

I started from creating the bear made up of cubes, with the code provided by the professor. I modified it in a way I needed. I created first the torso and I positioned it in the space, then I defined the head as its child. I positioned the head with respect to the torso, and for this I established no child but a sibling which is *leftUpperArm*. For the upper left part of the arm (front paw), I defined the structure in the way that all the upper parts, left and right, of the legs and of the arms was siblings to each other, and that of the lower part, left and right, of the legs and arms was the child, each of its corresponding superior part. Finally I defined the tail as the sibling of the upper part of the right leg and it has no further sibling an no child. In this way I defined the tree structure of the bear. Then I apply the colors to each part in the vertex-shader.

2.2 Adding Texture to the body and the head of the bear

According to the task I need to create 2 different textures, one for all the faces of the body of the bear except the head, and one for the head. I created the textures directly in the JS file. I used a boolean flag to identify if a part belongs to the bear or to the tree to apply on which I superimposed the textures, and another boolean flag to stabilize which of the two textures to apply to a specific part. I decided to apply a texture to the body of the bear that seems like a shade. In this way to distinguish visually the different parts that compose it, is possible. While for the head I used a simple chessboard texture. I applied the textures according to the boolean flag in the fragment-shader, by superimposing textures on colors.

2.3 Creation of the Tree

I created the simplified model of the tree in the same way that I used for the bear. My tree is composed of two parts: the trunk and the treehair. First I establish the position of the trunk and then I created its child which is the treehair. This second part have no child and no sibling. Then I apply the color to these two parts in the vertex-shader as I done also for the bear.

The result of these implementations is represented in Figure 1.

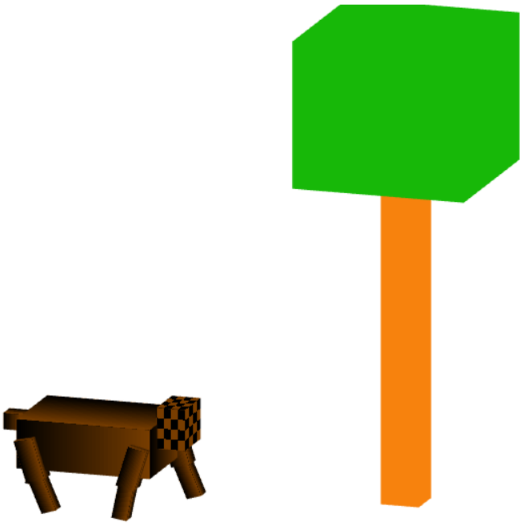

Figure 1: Representation of the bear and the tree

2.4 Adding Animation

The task was that the bear, starting from a established position, should came toward the tree by moving the paws, stand up and scratching its back against the tree. The movement is activated through the button *Start Animation*. To implement all the movements I created different functions. I started by creating a function *startMovement()* that implements the walk of the bear. Another function that is recalled in this, implements the movement of the leg that happens simultaneously with the translation of the body of the bear. Starting from their initial position and orientation, the paws move back and forth. This function works until the bear arrives in a specific point (a specific x coordinate). Then with the function *standup()* the bear rotate its torso until it is parallel to the trunk of the tree. The bear changes also the position of the head and of the posterior paws and it is ready for scratching. This last operation is implemented through two different function, one to go down and the other to go up. When the bear is standing, with the function *scratchDown()* the body of the bear is translated along the y axis by an y coordinate variation and it folds the posterior paws. To make this flexion, I changed the angle that determine the inclination of the paws. To establish how this angles should change, I compute through a trigonometric function the values, one for the upper part of the leg and one for the lower part of the leg. In this way, at the same time, the body is shifted downwards and the legs bend over. When the movement downward is finished, the function *scratchUp()* to go up is called. This function implement the movement in the same way of the *scratchDown()* function, but reversed. So it computes an upwards translation with the same variation used in the *scratchUp()* function. The same rotation of the posterior paws is applied. This movement is repeated continuously.