Homework 2 - Interactive Graphic

1 Create Hierarchical Model of a simplified Grizzly Bear

The bear is represented by a hierarchical structure. Hierarchical structures make it possible to represent relationships between even very complex objects. In particular, the structure is a "tree" formed by nodes, each node can have a son and/or brother. In this case, most of the nodes in the tree were already present in the file, i just rotated the bear to do it move on all fours and also added the tail and two eyes to make it more realistic. The tail and eyes are children of the torso.
The structure of bear is as follows:

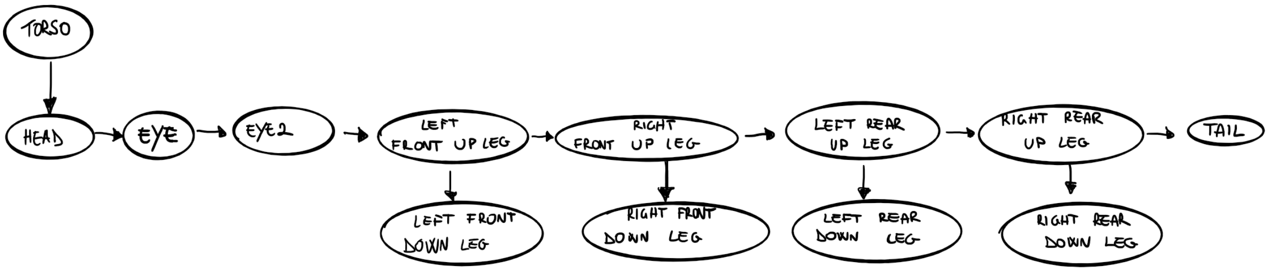

Figure 1: Structure of Bear

The practicality of hierarchical structures consists in animations, because if we want to apply rotation to the whole object (to the whole figure) we are going to do it directly to the root of the object, instead of applying n rotations to the individual components. As a result, the change is made to all the children of that root (or internal node).

2 Add a texture to the all the faces of the bear, except the head. The head has a separate texture.

The textures were added using images, then with the image base texture mapping technique.
The management of the textures to be applied was done through the flags. I inserted four textures: two for the bear (body and head) and two for the tree (crown and trunk).
I have choosen four boolean variables in the fragment shader to manage the different textures, which will be set to True or False depending on where I am. For example, if I am in the "head function of the bear" and I want to apply a particular texture, I will have set to True only the flag of the texture that I care about, instead the others will be set to False.

Activate and Bind the four textures using the WebGL functions `gl.activeTexture` and `gl.bindTexture`. In particular, with *activeTexture* i specified the ID of the associated texture and in *bindTexture* the texture that I am considering.

Code example:

```
    gl.activeTexture(gl.TEXTURE0);
    gl.bindTexture(gl.TEXTURE_2D, textureB);

    gl.activeTexture(gl.TEXTURE1);
    gl.bindTexture(gl.TEXTURE_2D, textureH);

    gl.activeTexture(gl.TEXTURE2);
    gl.bindTexture(gl.TEXTURE_2D, textureT);

    gl.activeTexture(gl.TEXTURE3);
    gl.bindTexture(gl.TEXTURE_2D, textureC);
```

3 Create a (very simplified) model of a tree and position it near the bear.

The tree was also represented using a hierarchical structure. The tree is composed of several components: a trunk and five crowns; it is independent of the bear, The trunk is the root of the structure; i used the same vector that I use to represent the bear, therefore the nodes in the tree structure are all brothers.
I added more crowns, even if they weren't required, to try to represent the tree more truthfully.

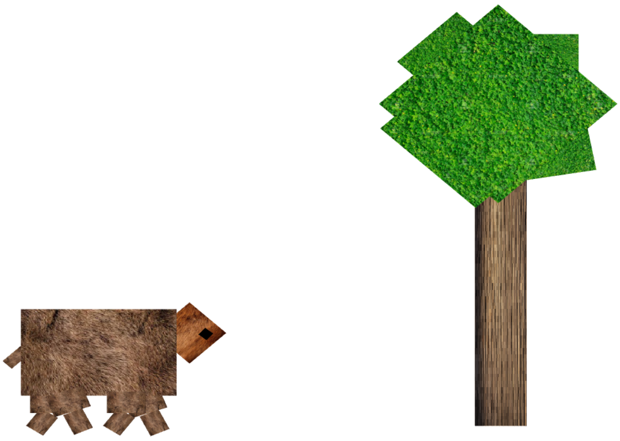

Figure 2: Bear and tree

4 Add a button that starts 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 (alternatively back and forth) the legs,then stands up and starts scratching its back against the tree.

The animation of the bear that approaches the tree and then gets up by scratching its back on it, is done by working on the rotations and translations of the components of hierarchical structure created to represent the bear.
At the beginning to move the bear itself, the value of the X coordinate of the translation of the torso is increased, and at each modification the value of X is updated, passing torsoId to the function `initNodes`. The legs and head also make forward and backward movements, controlled by flag variable `move`, to simulate the real behavior of the bear. After that, when the bear gets close enough to the tree, it is rotated on itself along the Y axis of torso until it is with its back to the tree.
Finally, he gets up, applying a rotation around the Z axis of torso, and starts making an up and down movement with legs and Y axis of torso to simulate the fact that he is scratching his back.
In addition to the animation button, I inserted a button to reset the animation itself, which once pressed refreshes the page.

To allow continuity to the animation, I used Javascript's `setIntervall` function, where I specified an interval of 150 ms.