Run, monkey, Run!

Interactive Graphics' Project

Giorgia Piernoli

June, 2019

Introduction

The project’s purpose is to present a 3D game. The game rules are simple: the player has to manage a runner monkey avoiding palm trees and trying to get as far as possible in the virtual world. To make the experience more pleasing different animations have been implemented and two new elements have been added: bananas, which give to the player a bonus on the value of the distance (+50), and apples, which give to the player extra lives, so, in case a palm tree is touched, the game does not finish immediately. Being the game set in the amazon rainforest it is possible to spot a parrot flying from time to time.

The world

The world is implemented as a rotating sphere, so it is an infinite looping world, and has a simple color texture representing the grass. There are a global ambient light source and a directional light source, on the top left of the screen, that creates the shadows of the world's objects.
The position of the camera is chosen so that just enough scene is visible to make the game entertaining. There is also a second camera positioned on the top/behind the head of the monkey and it's possible to switch the camera, during the game, pressing the 'C' key on the keyboard.

The hierarchical models

The implementation of all the hierarchical models is done using the three.js library.

- **The monkey**

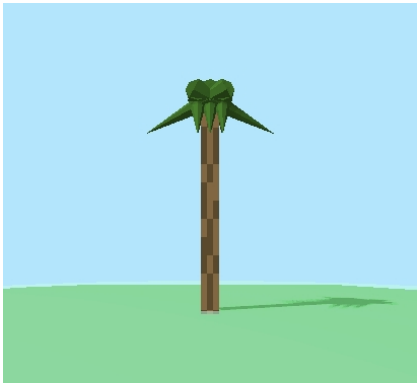
The hierarchical model of the monkey is composed by cubes and, for the thighs, cylinders.
The root of the model is the torso, to which every other part of the model is attached.
This is the only model that has animation functions, in particular it can run and jump.

The movements are created using the rotation function on the motion parts of the bodies, the velocity of the movements is controlled by the velocity of the whole game in relation to the level of complexity reached by the player. While running, the monkey oscillates the torso and moves arms and legs. The legs are divided in upper and lower part and each of them moves individually. When the monkey jumps the arms are raised.

- **Bananas**

The hierarchical model of the banana is composed by 3 parts: 2 cylinders (one for the final part and the other one for the middle part) and a cube for the stalk (the brown part). The banana element gives 50 points of bonus distance and slows down a little bit the constantly speeding world.

- **The palm trees**



The hierarchical model of the palm tree is composed by a long rectangle with a chessboard texture for the trunk and by cylinders (with the bases converging on top) for the leaves.

- **Apples**

The apple is a sphere with 2 cubes, one for the leaf and the other for the stalk, which make the apple look more realistic. The apple element is a life, so if the money catches one of them and, subsequently, touches a palm tree, it will survive but will loose one life point.

- **Birds**

The bird has a body that is a rectangle like the head. The beak, the tail, the paws and the wings are cones. The bird is animated so when it flies it moves the head and the wings.

The user interaction

An homepage for the game explains the rules to the player, when he's ready he can start the game pressing the button "Start", which is controlled by the HTML DOM getElementById() method.
During the game, the player can jump pressing the spacebar on the keyboard (this interaction is controlled by a onkeyup event). If the game ends, he can restart the game simply clicking where he wants on the screen (instead, an EventListener controls this other one).
All these structures (labels, buttons, instructions, etc..) are implemented in the style.css file.

Functions for the game's management

- **The updateElementsPosition() function**

The spawn of the apples, bananas and palm trees is managed in that way: 30 random elements (decided by a random number, one for each of the 3 types of objects) are generated at the start of the game. The distance between the elements is composed by a fixed variable plus a random variable, anyway is calculated to leave enough space to make the jumps possible. When an element reaches a certain position degree in the world's

sphere, this element is substituted with another random element (that could be also the same type) and the random variable of the distance between element changes. A similar approach is used for birds, there are 8 of them at approximatly the same distance one another.

- **The loop() function and the update functions**

The loop function is necessary to manage the movements of the hierarchical models and to update the position of the objects and the speed of the whole game. The level is updated after a certain time space sets by a TimeInterval (+1 level every 5000 milliseconds), the increase of the level involves an increase of the game's speed. When the game restarts, the level, the speed and the distance are re-initialized to the native values.
The updateRotation() function updates the velocity of the world's sphere rotation in relation to the level reached by the player.
The updateDistance() function updates the distance reached by the monkey using the speed and the seconds passed since the TimeInterval was set.
In the end, the checkCollision() function checks the monkey's position respect to the position of the other objects: if a collision with a palm tree is detected, and no extra lives are available, the game is over; if a collision with a banana or with an apple is detected the distance value is updated or a +1 is added to the value of lives.

References

[1] Three.js documentation: https://threejs.org/docs/

[2] Inspiration: https://github.com/MarcoSchaerfCourses/t-rex-runner-game