

Report Interactive Graphics

'Snowball'

Teacher: Marco Schaerf

Falesiedi Flavio Massimo 1595746

Pinto Franco 1592439

a.y. 2018-2019

Contents

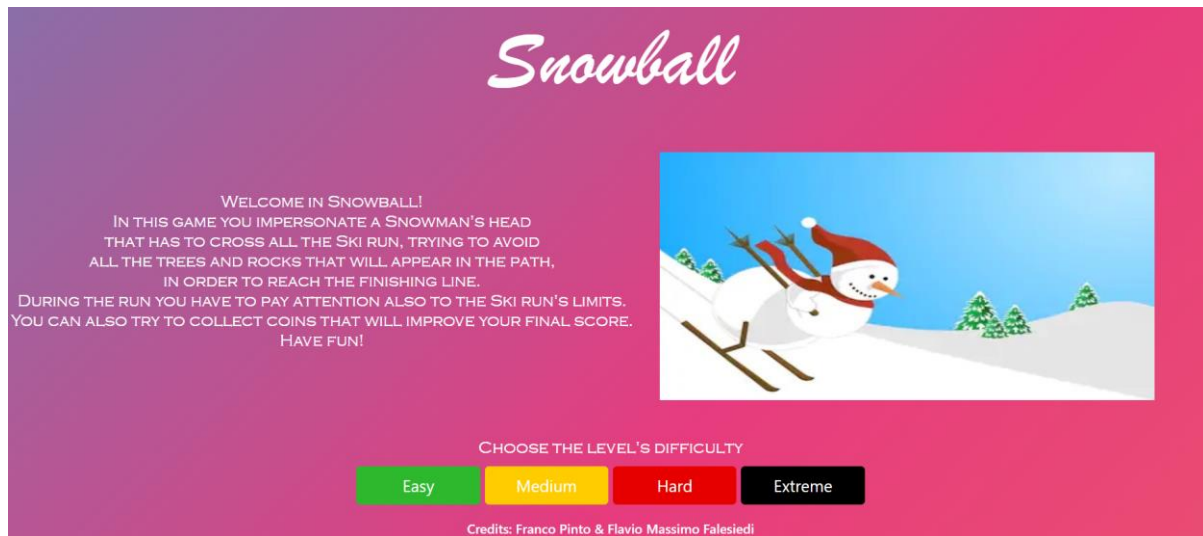
1 Snowball Game	2
1.1 Overview	2
1.2 Introduction	2
2 Implementation	
2.1 Environment	3
2.2 Libraries	3
3 Technical aspects	3
3.1 Lights and Textures	4
3.2 Models	5
3.2.1 Hierarchical structures	5
3.3 Animation	7
4 User interactions	7
5 References	8

1 Snowball Game

1.1 Overview

Our game is based on the “Chilly Snow” mobile game, in which we have a snowman that has to traverse a ski run until it reaches the finishing line, trying to avoid all the present obstacles.

There are two interfaces, the first (mainpage.html) is the one in which we have the level selection and the description of the game itself.



While the second one(index.html) presents the real game.



1.2 Introduction

Snowball is a game in which the user impersonates a snowman that has to be controlled with the keyboard, the spacebar in particular, in fact will change its direction all along the path.

The aim is to avoid all the static objects, like the trees that are randomly generated in the ski run, and dynamic objects like rocks, that will traverse the environment when the snowman is near to them, in order to reach the end of the level.

If the snowman crashes against the obstacles presented above or on the fences present on the edges of the ski run, an hitting sound is produced and a window will pop up giving to the user the possibility to restart or come back to the level selection.

Another important aspect is that a user can collect coins in order to improve its final score that otherwise will be influenced only by the reaching of the finishing line that has a value of 3000 points. In particular there are 15 coins each one with value of 100 points, so the maximum available score will be 4500.

There are four different levels of difficulty in our game: easy, medium, hard, extreme. According to the level selected the number of trees and rocks will change. Each level is then accompanied by a soundtrack that will loop during the various level in order to add some emotion to this experience.

2 Implementation

2.1 Environment

The environment used for our project is Babylon.js. Babylon.js is a real time 3D engine using a JavaScript library for displaying 3D graphics in a web browser via HTML5. The source code is available on GitHub and distributed under the Apache License 2.0. The Babylon.js 3D engine and user code is natively interpreted by all the web browser supporting the HTML5 standard and WebGL to undertake the 3D rendering.

2.2 Libraries

The library used are:

- *bootstrap.min.css*: through this library is possible to debug the babylon objects.
- *babylon.dynamicTerrain.min.js*: it provides a way to display a large terrain dynamically morphed from a map of 3D data.
- *babylon.gui.min.js*: it is used for inserting graphical elements into the user interface.
- *babylonjs.loaders.min.js*: it contains Babylon.js loaders to enable loading filetypes such as .gltf.
- *pep.js*: to ensure pointer events work consistently in all browsers
- *jquery.js*: jQuery is a fast, small, and feature-rich JavaScript library. It makes things like HTML document traversal and manipulation, event handling, animation, and Ajax much simpler with an easy-to-use API that works across a multitude of browsers.
- *jquery-ui.min.js*: jQuery UI is a curated set of user interface interactions, effects, widgets, and themes built on top of the jQuery JavaScript Library

3 Technical aspects

The project is composed by the following js files:

- *Snowball.js*: this is the class in which we create a part of our main character. In particular we use this to manage the creation of the snowman's head containing a set of parameters like the diagonals, that through a specific function allow the control of the character.
- *Tree.js*: it creates the hierarchical model of the tree that is composed by a trunk and a foliage.
- *TreeGenerator.js*: this file allows us to manage the creation of a set of trees calling the previous class (Tree.js). This will receive in input the number of trees to generate in the environment, in particular it will assign to them a random position along the ski run.

- *prova.js*: this is our main class in which we manage the particle effects, the hierarchical model of the snowman, lights, textures and shadows, skybox, the animations, the meshes, and the user interaction.

and by the following two html files:

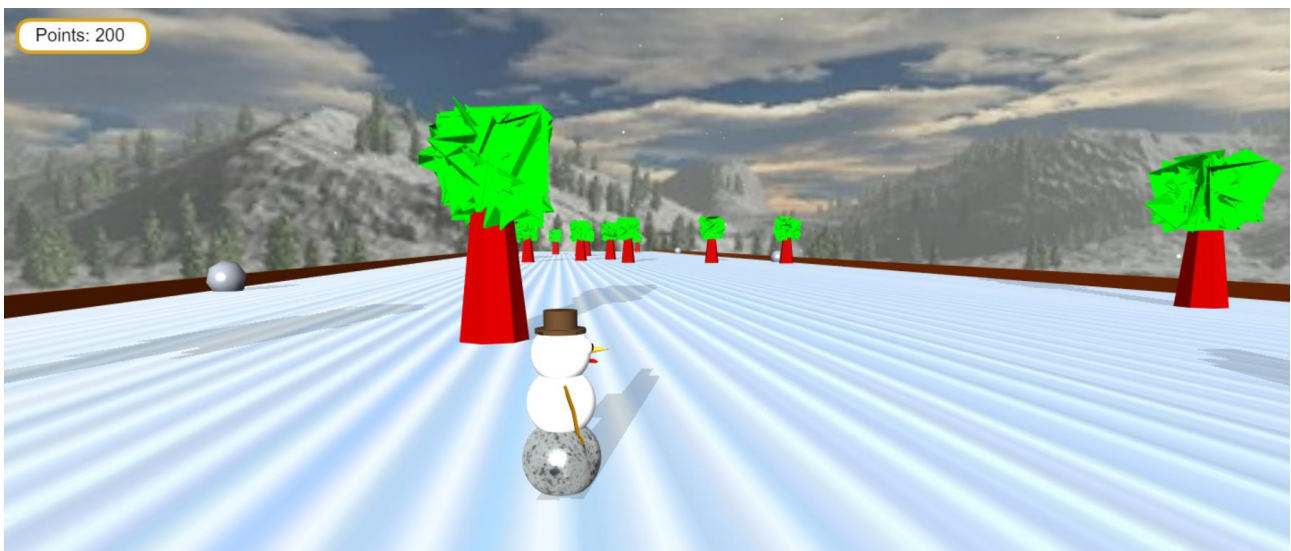
- *index.html*: this class manages the imports used in *prova.js* and all the styles and fonts for the popups that are created in the initial and final phase of the game or when the snowman crashes.
- *mainpage.html*: this is the page in which the user can read the game's instructions and its description. He/she can also select the level's difficulty through a set of buttons differentiated by colors (green, yellow, red and black).

Furthermore, we have imported three Babylon.js models (*flag.babylon*, *hat.babylon*, *tongue.babylon* and their respective manifests) [5] in order to use them in the game's scene, in particular the last two are used in the creation of the snowman's hierarchical model.

3.1 Lights and Textures

We used two lights and three textures, let's see how.

For what concerning the lights we used a hemispheric light that simulates the ambient environment light and a directional light that is used for the projection of the shadows present in the scene.

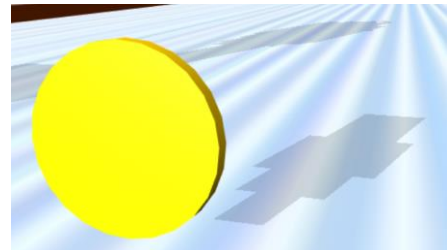
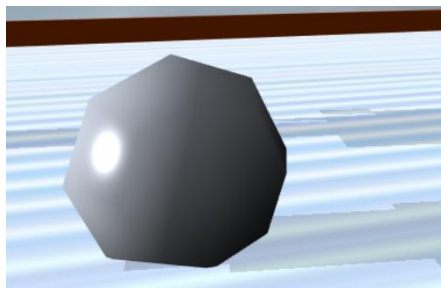
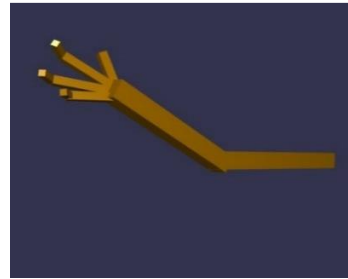


For the textures instead, they are applied in the scene in the following way: the first one is applied on the ski run's ground in order to give it a snowy effect, the second one is on the lowest part of our snowman and is applied to emphasize its rotation, the last one is used in the sky box to have an effect of a cloudy place with a snowy ground and some mountains in the background.



3.2 Models

The models present in our game are various and of different type. Some of them are imported as Babylon's models but most of them are created by hands. The hand crafted ones are in particular the rocks, based on a sphere with low segments to underline the edgy shape of them; the coins that are simply cylinders with a high number of tessellation in order to present them as smooth as possible; the fences are based on boxes and they are generated on the edges of the ski run; the carrot that is the nose of the snowman, based on a cone; the eyes that have been created through a sphere and a black torus that represents the pupil; the snowman's arms formed by a set of boxes; and the trees.



The other models are the flags, the tongue and the hat that are imported through the Babylon method ImportMesh. For what concerning the flags we have used the clone method that allows us to have multiple instances of the same model without the needing of its recreation from scratch.

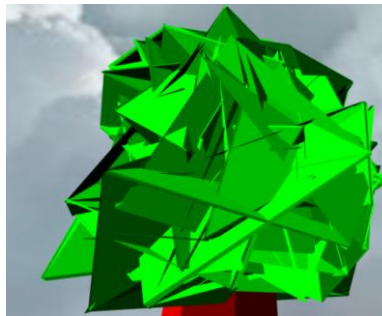


3.2.1. Hierarchical Structure

As mentioned above, there are some models that are formed by more than only one mesh or component, they are in particular the trees and the main character, the snowman.

The trees, as mentioned above, are composed by a foliage [\[2\]](#) that is a sphere and the trunk that is a cylinder; in this case the hierarchy is very simple in fact it is developed in only one level, from the trunk to

the foliage. An interesting thing is how the foliage is created, starting from a sphere in fact we have developed a function that transforms it, through a sort of tessellation, in a more realistic set of foliage rather than a simple green sphere, this foliage is generated for each tree so it will be different for each of them.



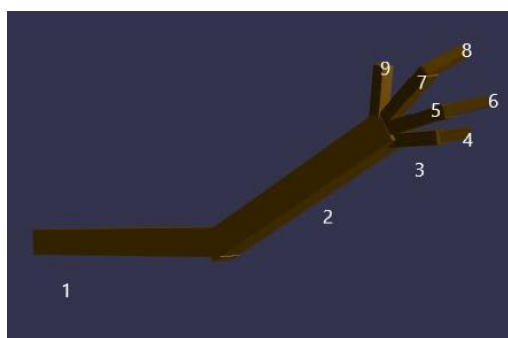
The Snowman instead is the more complex one, it is formed by 28 components, let's see all of them and how they are connected.



The central part of the snowman is the body, it is formed by three spheres, each one on a different level from the other. The first level that is the father is the head, the second level is the central part of the body, and the third level is the lowest ball. Let's see in detail the head. It is formed by the eyes and their pupils, the nose, the tongue, and the hat; they are all second level elements (so children) of the head itself, except the pupils that have as father the eyes posing themselves on a third level with respect to the head.

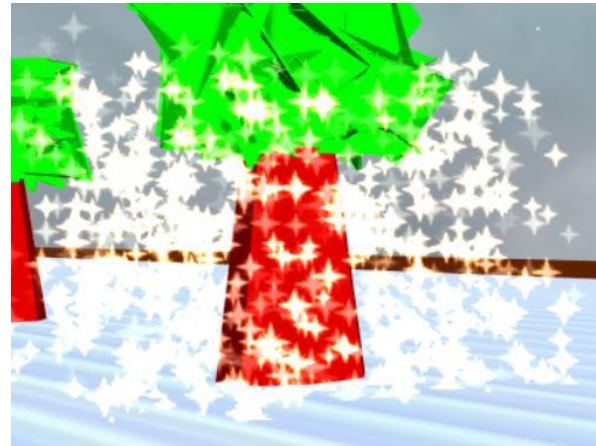
Finally, we have the arms; they are composed by a lot of boxes here structured: upper arm, lower arm and four fingers; three of these four fingers have also another phalanx, for a total of 9 components for each arm.

Its hierarchy starts from the upper arm that is children of the central body ball (so a third level hierarchy), from the upper arm we pass to its child, the lower arm and from this lower arm we have four children that are the fingers. Three among these fingers are then parents of their phalanxes reaching six levels hierarchy.



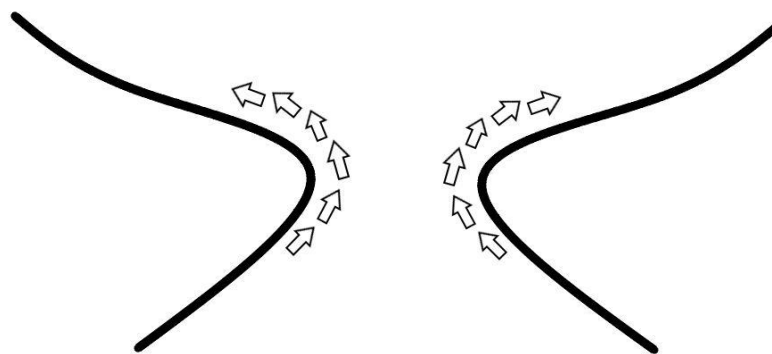
3.3 Animations

In the project are presented different animations. A couple of them are imported in order to develop the particle effects, in particular they are the snow that falls during the run, and the shining star that appear when the snowman hits an obstacle or the fence.



The others are all done by hand and they are related to the rocks, to the coins and to the snowman. On the rocks we have applied a rotation in diagonal that simulates their traversing of the ski run from one side to another and a set of 8 frames used in order to simulate a bouncing during their movement, while on the coins we have a rotation on the y axis. For what concerning the snowman we have a rotation of the lowest body part that changes in diagonal any time that the space bar is pressed, then the other parts of the body rotate gradually from right to left and from left to right when the snowman changes its direction.

All these rotations are accompanied by a set of 12 frames that are activated every time that the user press the space bar in order to represent this movements:



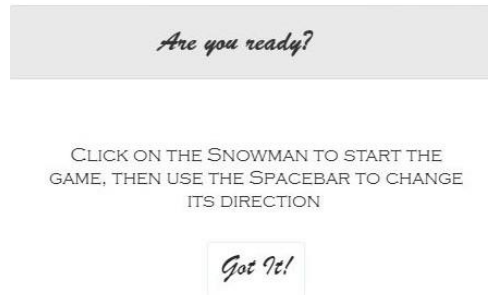
Finally, we have an animation related to arms, they develop an up to down and vice versa specular movement in order to simulate the behavior of the human's arms during the run.

4 User interactions

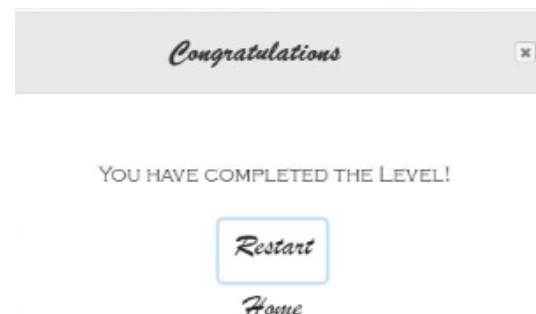
As mentioned above the user can interact with the game in different ways; first of all it can manage the difficulty of the level choosing which button to press in the game's main page



while during the level it will select a button in order to make the initial popup disappear.



Near to this the user must click on the snowman's body in order to start the run of our character, if this is not done and the user clicks on the scenario he will be free to navigate on the ski run with the arrows and with the mouse, he can also rotate the camera in order to reach some elements and see them in a more detailed way. Finally, other user interaction will be the space bar pressing for changing the character's direction and the clickable button that will appear when the snowman will hit some obstacle, or when it reaches the finishing line.



5 References

- [1]. <https://www.doc.babylonjs.com/>
- [2]. <http://www.pixelcodr.com/tutos/trees/trees.html>
- [3]. <https://codepen.io/herrberk/pen/GarMvo>
- [4]. <https://jquery.com/>
- [5]. <https://clara.io/library>