

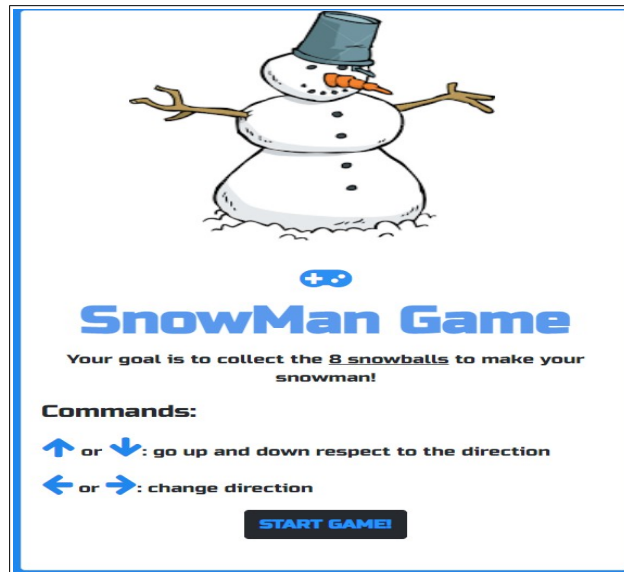
## Project Report – Interactive Graphics

### Attilio Borea - 1816730

#### 1. SnowMan Game: introduction.

My final project consists in an implementation of SnowMan Game. Game's goal is to collect all the 8 snowballs, that will make appears the snowman. The project starts with an index page which contains the commands, then user can start the game by clicking "start game". Now he can control the player and enjoy the game.

In the picture below is shown the index page.



#### 2. Framework.

ThreeJS is the library that I used to develop mainly the project. ThreeJS is a Javascript 3D library that allows to easily implements lights, textures, models and everything is necessary to build a 3D application, unlike native WebGL implementation. Mainly ThreeJS's functins used is this project are related to the creation of the geometries, the materials and the meshing between these 2 elements in order to build complicated objects. Major details about models in section 4.

#### 3. The game: details and technical aspects.

The game in divided in 2 phases. In the first Phase user can controls the main character, Snowboarder, that goes on a snowboard, in fact he can move the character by pressing the arrows on the keyboard, sky is implemented with Sky function of ThreeJS with related parameters and Sun. The main goal is to collect all the snowballs that are located randomly on the map. User can always check the percentage of the snowman on the scoreboard, that changes dynamically. There are other objects in the map, user cannot have conflicts with them during the movement. When user start collecting snowballs the character will start to make the snowman, infact after 3 snowballs collected the snowman will begin to appear, and after 6

snowballs the snowman will grow up (more details about this aspects will be discussed later). After 8 snowballs we have the second Phase of the game. In the pictures below is shown the snowman with his growth.



In the second phase user complete the tasks to complete the game,the incomplete snowman will start rotating and will disappears and in the meanwhile it start snowing! That's because the snowman is complete and it will appears from the floor ! After this the game is ended but the user will be still able to go around the map. Picture below shows the completed snowman.



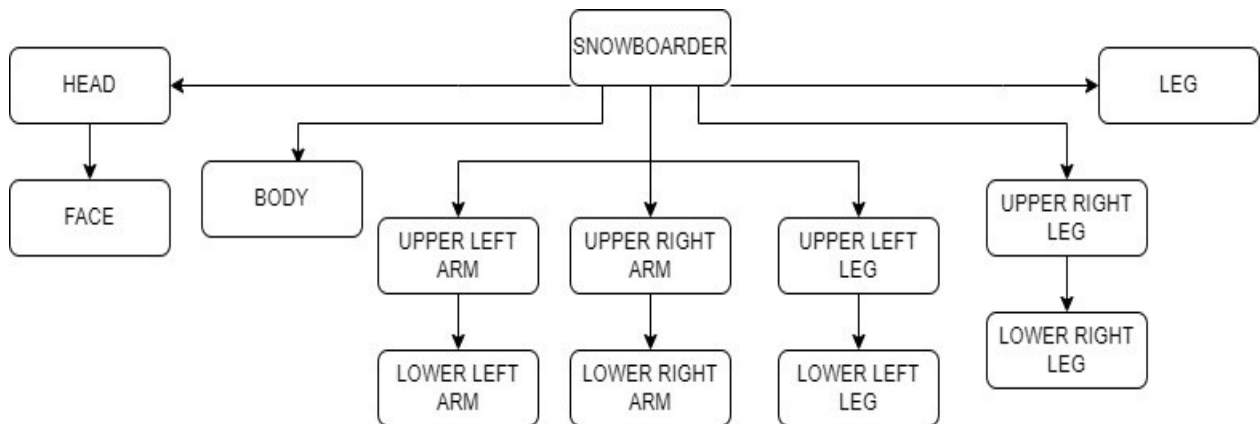
#### **4. Hierarchical Models.**

In order to satisfy all the requirements I build different hierarchical models. A hierarchical model is a more complex model, but the creation of hierarchy between the components of the main object allows to perform better animation

##### **a. Hierarchical Model – Main Character(Snowboarder).**

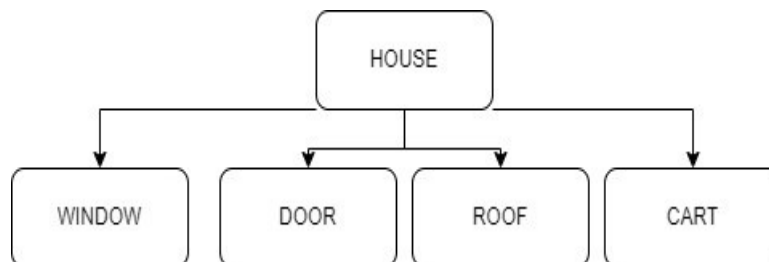
The main Character is composed by these elements: face, hat, body, rightHand, leftHand, upperLeftArm, lowerLeftArm, upperRightArm, lowerRightArm, leg, upperRightLeg, upperLeftLeg,lowerRightLeg and

lowerLeftLeg. There are also 3D models like the snowboots and the hand which I decide to import from Sketchfab This was made in order to make players move on the snowboard and to let him doing animation during the collection of the snowballs.



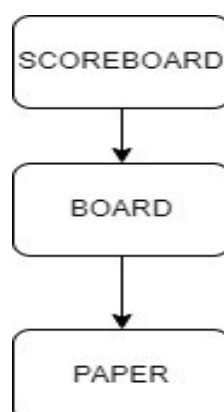
### b. Hierarchical Model – House

The house is made by several elements. It's composed by the structure(called House in the code), the Roof, the Door, the Window and the Cart which is imported by Sketchfab .



### c. Hierarchical Model – Scoreboard

The scoreboard allows the user to double check the completion percentage of the snowman. In order to implement this feature, I decide to import a model from Sketchfab, that is the structure and I have added interactive objects that are modified during the game. In fact, this hierarchical model is composed by the Scoreboard, the BoardPaper(is the white paper on the scoreboard) and the textGeometry, that is made from a loader that load the font, and the value of the text is modified instantly when the user collect new snowballs.



## **5. Lights.**

Lights are fundamental for the developing of this project. In fact, I used different kind of lights for different situations. First of all an Ambient soft white light, created by function `THREE.AmbientLight`, then I implemented a `directionalLight`, created by function `THREE.DirectionalLight`, in order to provide the lights over the window.

## **6. Textures.**

In the Texture folder are shown all the texture that is used in order to give more realistic effects to the object. Textures are for: hat, arms, body, paper, face, snow of the snowballs, floor, head, legs, house and roof. Textures are used in forms of images with the map with the material, but even in forms of simply colors.

## **7. Animations.**

I can synthesize this section by dividing for each animated object.

### **a. Main Character**

Main character can be moved by pressing arrows on keyboard, generating an event, this is computed inside the function `KeyPressed(e)`. Animation is computed by translations/rotation following the chosen direction, character will be stop during movement because he stands on a snowboard that does the movement. Collisions are avoided when user would go on or back by var collisions that can be true or false if the collision is present or not, this is made by the function `.intersectsBox()` of ThreeJS. In case of changing direction I check the direction and I rotate the main character(Snowboarder) to the correct position. Main character has others animations. In fact the collection of snowballs determines changes on the environment of the game. After that an user collects 3 snowballs, main character will start to make the snowman which it will start to appear in front of the house. After 6 snowballs he will continue to make the snowman which will keep on growing. I compute this animation by translations and rotations of `OldSnowMan` and `OldSnowMan1` and after the animation the snowman will appear almost completed. Then at the end, when the user has collected all the snowballs, the finished Snowman appears.

### **b. Snowballs**

Snowballs rotate all the time to give to the user a better experience, that's made in the function `animate` by a simple rotation incremental.

### **c. Snowman**

When the user takes all the snowballs the old snowman (which is uncompleted) will start rotating on y-Axes until it will disappear, that's made in the animate function by increasing rotation value until it reaches value of 12, in that case the flag of animation will be changed, letting space to the completed Snowman animation which is the last animation of the game.

### **d. Snow**

In the meanwhile the completed snowman appears, after collected all the 8 snowballs, user can realize that it will start snowing! This animation is done by two functions, one for the snowflakes and the other for the translation of these snowflakes.

## **8. User Interaction.**

This project is really dynamic and let to the users interact actively. In fact, user can think about the growth of the snowman and can collect the snowballs in order to complete it, in fact collecting snowballs will change dynamically the snowman, so he can choose to change snowman's appearance or not make it by not collecting the spheres and just go around in the map. User can check always on the Scoreboard the percentage of the snowman's completion (There are 8 snowballs and the percentage will grow ten to ten until he reaches 7 snowballs, the 8 snowball will let you complete your snowman), this was really complicated to be implemented, in fact the information about the collection is given by given a text-object that changes dynamically, in the picture below is shown the Scoreboard. User has the completely control of the character, can go around the map (collision are not allowed). User can completely change his viewpoint by use mouse or trackpad, can zoom in and out and change the rotation of the camera.



## **9. Final considerations.**

This project was focused to give the most possible user interaction and gives to the user the possibility to do all the choices that he wants. It integrates different complex-hierarchical models and uses as best the animations with an integration of threeJS, that helps in development time, making easy and faster the steps of development of a 3D Application.