

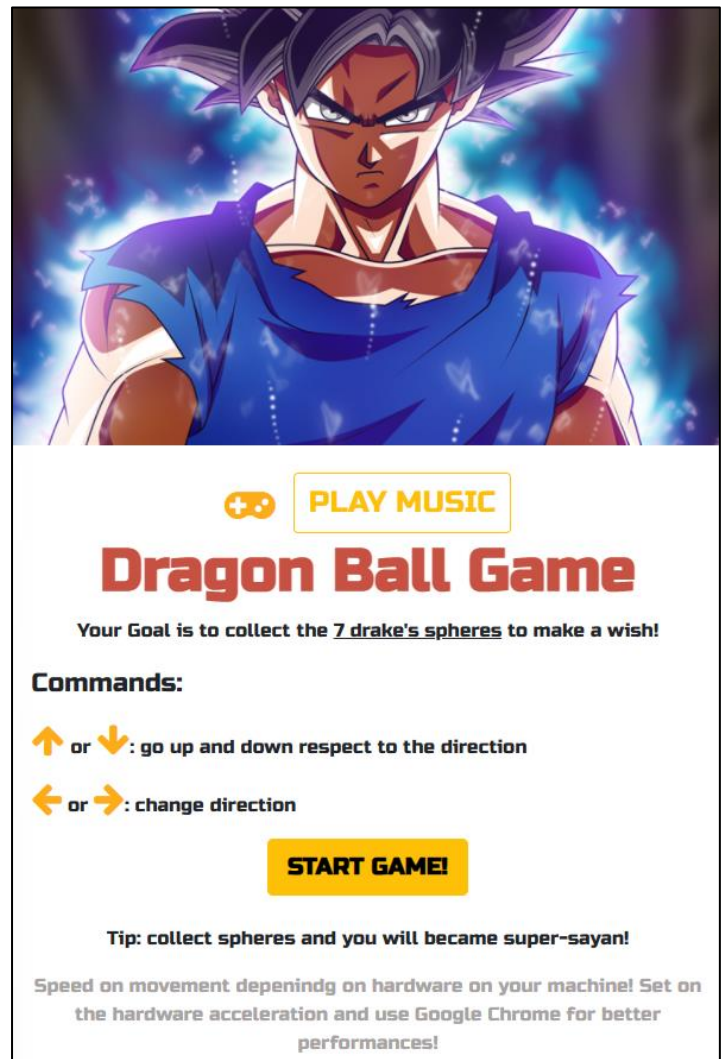
Project Report – Interactive Graphics

Salvatore Iorianni 1808542

1. Dragon Ball Game: introduction.

My final project consists in an implementation of Dragon Ball Game. This game has the goal to collect all the 7 spheres, that will make appears the dragon of wishes. The project starts with a pre-index page with 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 on the right 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 this 2 elements in order to build complicated objects. Major details about models in section 4.

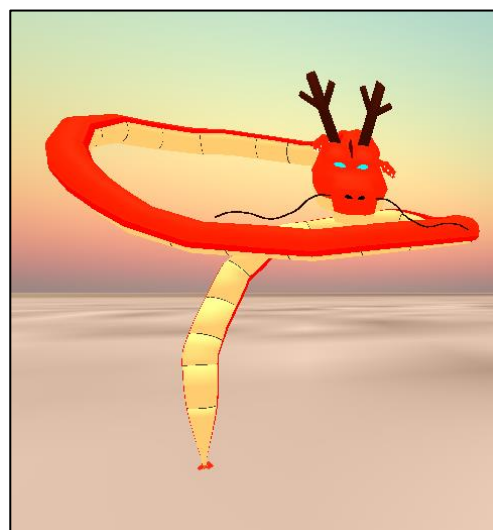
3. The game: details and technical aspects.

The game is divided into 2 phases.

In the first Phase user can control the main character, Goku, that flies on a cloud, 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 red spheres that are located randomly on the map. User can always check the number of the sphere on the board, that changes dynamically. There are other objects in the map, user cannot have conflicts with them during the movement. When user starts collecting spheres the character will grow, becoming more powerful, in fact he becomes SuperSayan after 3 spheres, and becomes SuperSayan4 after 5 spheres (more details about these aspects will be discussed later). After 7 spheres we have the second Phase of the game. In the pictures below is shown the character with his transformations.



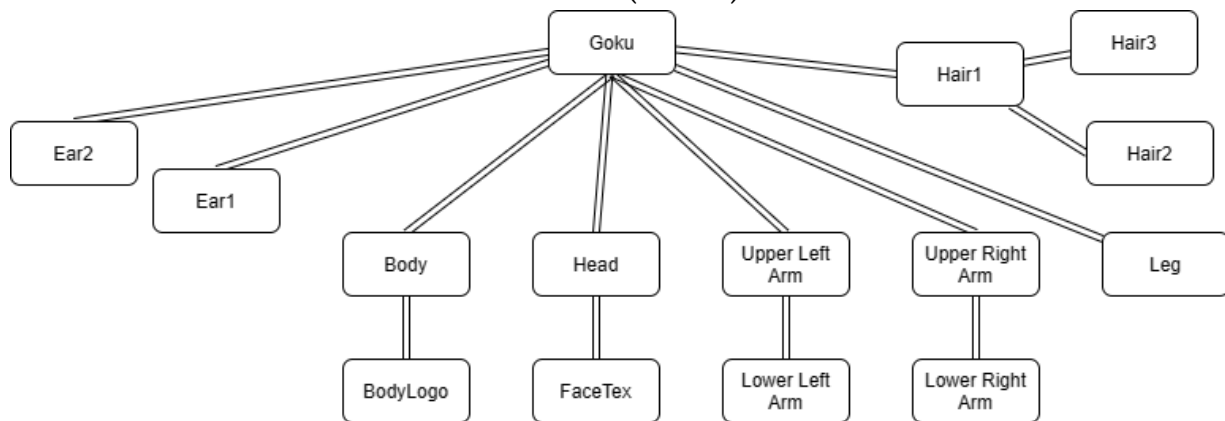
In the second phase user completes the tasks to complete the game, the house will start rotating and will disappear! That's because that is the house of the magic dragon Shenron, that will appear from the floor and will make true the wish of the user! After this the game is ended but the user will be still able to go around the map. Picture on the right shows Shenron Dragon.



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

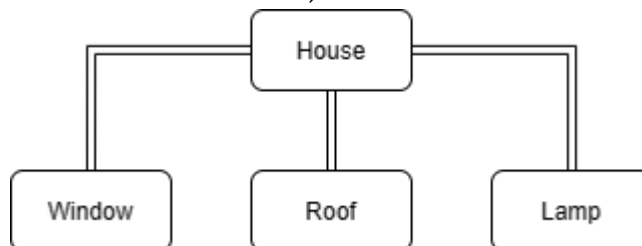
a. Hierarchical Model – Main Character(Goku).



The main Character is composed by these elements: face, hair(3), ears(2), body, bodyLogo(that is the logo on the shirt of the character), upperLeftArm, lowerLeftArm, upperRightArm, lowerRightArm, Leg. This was made in order to make players move on the cloud and to let him doing animation during the research of the spheres.

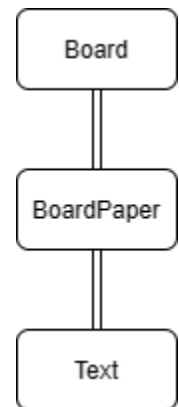
b. Hierarchical Model – House

The Big House is made by several elements in order to perform an animation at the end of the game, where it will start rotating until it will disappear. It's composed by the structure(called House in the code), the Roof, the Door and the StreetLamp.



c. Hierarchical Model – Board

The board allows the user to double check how many spheres he has already got. 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 Board, the BoardPaper (is the white paper on the board) 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 spheres.



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 lighted, created by function `THREE.AmbientLight`, then I implemented different pointLights, created by function `THREE.PointLight`, in order to give the lights from the StreetLamp to the near environment. Last kind of light that I implemented is a `directionalLight`, created by function `THREE.DirectionalLight`, in order to provide a better illumination when the dragon spawn.

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: cloud, Arms, Paper, Logo of the body, body, ears, face, fire of the spheres, floor, hair (blond, black and red), 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 synthetize 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 KeyPrssed(e). Animation is computed by translations/rotation following the chosen direction, character will be stop during movement because he stands on a cloud 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(Goku) to the correct position.

Main character has others animations. In fact the collect of speheres determines changes on the player. After that an user collects 3 sphere, main character will change color of hair, becoming a Super Sayan and after 5 spheres he will stand up right arm, I compute this animation by translations and rotations of upperRightArm and lowerRightArm and after the animation the character will change again color of hair, that will be red.

b. Spheres

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

c. House

When the user takes all the spheres the house will start roating 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 Dragon Animation.

d. Dragon

This is the last animation in the project and it's the end of the game. The dragon will appears from the floor, this is made by an increase of object.position.y until it reaches the sky. An alert will appear, letting know to the user that he can make a wish.

8. User Interaction.

This project is really dynamic and let to the users interact actively. In fact, user can think about the growth of his character and can collect the

spheres in order to improve itself, in fact collecting spheres will change dynamically the character, so he can choose to change itself or staying as it is by not collecting the spheres and just flying in the map. User can check always on the Board the number of spheres that he collected, 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 Board.



User has the completely control of the character, can fly around the map (collision are not allowed). User can see how the reactions to lights are, in fact if the characters stay near the Street Lamp he will be illuminated more. 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.