INTERACTIVE GRAPHICS PROJECT REPORT

Members: Caruso Emanuele, Cicero Edoardo, Giunta Daniele

OVERVIEW

We decided to develop the project as a Demo expandable game: it is about a robot, Randy, that has to walk around the spaceship in which it was built, and interact with the environment in order to reach the end of the starship.
In the following there will be a description for which regards the environment that it has been used, libraries and tools that have been added to the project, a kind of user manual describing all the technical aspects, the interactions of the user with the game and how to play it, and finally some possible future developments.
For the aim of our project we chose the Java ThreeJS library because of its good documentation that helped us a lot. In addition, to implement gravity, we used a library called Cannon.

TECHNICAL ASPECTS AND FEATURES

The game is based on the idea of letting the player control a robot that can act in a spaceship to help humans repair, manage and in general interact with objects. First of all, the robot can move of course: we chose the A-D keys as in common games because the game in a 3D game but with 1D motion control. Further in the game the player can also transform the robot, from a spider-like model to a simple rolling sphere. Randy can interact with tablets to open doors or to activate the rescue cabin through a labyrinth mini-game in which the user has to move a square to unlock passages that let it reach the goal and open the door. In addition, we put a zoom-in in the rooms with the window that is triggered by the robot position for a good view of the Earth. The interactions are not over: since we are controlling a robot in an environment that is not without possible failures, it may happen a blackout, so we implemented a flashlight on the head of the robot that can be switched on with T and controlled by the user with Q and E to make it rotate.

The development of the game started from the **robot**: we created a complex and articulated robot using spheres, boxes and cylinders to build the head (boxes), the body (sphere) and the legs (composed by 2 parts); as learned in the course, we linked all the parts to the body in order to have a better control of the whole structure. After that, we focused on the **robot animation**: the movement of course should resemble the one of the spider, so a rotation of all the legs around a central axis plus an extension, a stretch from the “relaxed” state have done the job. We also added an idle motion when the player does not push any button on the keyboard, and it is composed by 3 parts: head, body and legs movement. The sphere waves up and down, the first part of the legs attached to the body follow the sphere and, in the end, the head, through a sinus function, makes a random rotation of 360° on the body; furthermore, sometimes a random leg can have a kind of “spasm”.

The **scene setup** was built thinking of which could be the elements in a spaceship: large color-flat rooms with some devices inside as computers, tables and tablets. The spaceship has some windows too, in order to have a view of the Earth, and it let us implement a cool zoom-in. There are also special rooms, such as the rescue cabin room, the one with containers and floating cubes inside, that have been made possible by using another library (the Cannon library) and thanks to it we realized the physics of the objects, or the room in which there is a capsule that leaves the spaceship when Randy, the handyman robot, passes nearby. The physics has been mostly used to detect collisions and simulate gravity.

The spaceship has also moving objects inside, either triggered by the robot or just part of the specific room: not only we implemented the motion of pistons and of a surveillance camera at the beginning of the game, but also a rotating sphere and a complex rescue capsule that start translating and rotating when the player moves near it, and sliding doors too.

In order to make it as realistic as possible, we put textures on each object of the game: the floor, the pistons, the tablets, the doors, the surveillance cameras, and so on. Randy itself has a texture attached to its body and legs.

We introduced also an AI voice that interacts with the user in order to give him some information about what he has to do, or just to communicate if the starship is working properly or if it has some faults. For a clearer understanding, subtitles will appear at the bottom of the screen any time the AI will talk.

Finally, to help the user not to get stuck during some parts of the game, some hints will appear at the top of the screen explaining which of the buttons are needed to be pushed or simply in which of the rooms the robot has to move to go on with the game.

Functions from ThreeJS

- Geometry

The creation of all the objects in the scene starts from a proper combination of "geometry" and "material" that have been used in order to create the meshes of the objects: "geometry" defines the structure, while "material" describes the physical properties that characterize the future object. ThreeJS has a set of standard geometries that we combined together to build also complex objects and structures: in particular, we used SphereBufferGeometry, BoxBufferGeometry, CylinderBufferGeometry, TorusBufferGeometry.

- Material

We did some tests to decide which one among all the materials could be a good one, considering the light reflection effect on the surface and the impact on the performance of using many objects with that particular material. The cheapest one that we found was the BasicMaterial effect, but it was not possible to implement light reflection with it; it was impossible also with the Normal Basic Material one, so in the end we chose Phong Material, that implements light reflection very well as we saw in the course.

- Light

Light reflection was possible because we added light sources in every room and those are the common lights we can find on the ceiling of every building, but to make them more "sci-fi" we built a tube with cylinder mesh, we attached the spotlight to it and placed it on the ceiling. Talking about the lights we used a different color for the one in the elevator room and it makes that room appear to be a bit more blue, and also a blinking light in the dark room was implemented; the whole game has also an AmbientLight effect for two reasons: to represent everything less dark, and because of the fact that the textures attached to the objects are a bit darker than they appear out of the game.

- Music and audio sounds

To make the player a bit more inside the situation of being alone in a spaceship we added a kind of sad music through Audio and AudioLoader functions, and it starts after the user clicks on the Start button. The capsule's door and the sliding doors have a sound effect too, with appropriate triggers of course, and there's also an AI voice that guides the player and gives some information to him. We also added the sound of the metallic legs of Randy when he walks in the starship during the spyder mode. Finally, it is possible to listen to a kind of “final music” that denotes the end of our game.

Performance

As it has been said before for the material, not only we tried to make a game as light as it could be done to be run for any browser, but also as fluid as possible. We used dat.GUI to change quickly the variables related to resolution scale or shadows to see the immediate results on the game and we found a good compromise. The creation of the shadows influences the performances a lot, so we decided not to let WebGLRenderer render them. A good way to make the game playable also on old machines is the resolution reduction effect, that makes the game appear a bit more “pixellized” of course, but really more fluid, and it is possible for the player to modify that value on the top right of the screen.
Another thing about performances is the “room spawn”: the rooms are all generated once the first time/at the beginning, but then only the rooms located before and after the one in which the robot is are shown: in this way we saw an improvement on the performance at the cost of a bit more time that is needed to be spent to load all the objects at the beginning.

INTERACTIONS AND HOW TO PLAY THE GAME

The user is capable of course to **interact** with the robot and the surrounding environment in different ways; in fact, it is possible to:

1. make the robot move;
2. interact with tablets;
3. play a mini-game;
4. change from “spider mode” to “sphere mode”;
5. switch-on/off a torch, and make it rotate.

1. Inside the game it is possible to move the robot by simply pushing the buttons “**A**” and “**D**” on the keyboard, to move leftward and rightward respectively.
The robot can be moved both in “*sphere mode*” and in “*spider mode*”, therefore only bidirectionally. We payed attention to the fact that when the robot moves, it must not pass through objects: for example, when a door is closed the robot has to stop in front of it, and this is what we implemented.
The velocity of the movement of the robot changes with respect to the robot mode: in fact, when the “sphere mode” is running, the robot will move quicker than the “spider-mode” one, because, as it can be imagined, a sphere runs faster than a spider does, and so this is what we reproduced.
There is another aspect that has to be mentioned, even though it is not an interaction with the game, and in particular with the robot. We decided to make the robot itself “dynamic” even when it does not move: in fact, as it has been said in the previous paragraphs, its head starts to move by drawing an “eight” shape in the air anytime neither “**A**” nor “**D**” buttons are pushed, and it happens in both sphere mode and spider mode; instead, only when spider mode occurs and it does not move, it is possible to see that its legs start to make some movements; it is finally possible to notice that the body of the robot oscillates up and down when it does not move.

2. In the starship where Randy is, there are a lot of tablets distributed among the rooms: it is possible to interact with them by pushing the button “**F**” on the keyboard, as also an hint on the screen will suggest to do.
Some of the tablets are used to open and close the sliding doors of the starship, some others permits Randy to escape from dangerous situations and so on. While a certain number of doors slides automatically when Randy gets closer to them, in some cases just pushing the button on the keyboard will not be sufficient to unlock the doors.

3. In order to access from the first room to the second one, the robot has to pass over a door, but it is closed; to unlock it, it is necessary to play a short **minigame** that will appear on the tablet near the locked door; from now to the end of the minigame, the control of the user passes from Randy to a 2D-square, which represents the robot itself stuck in a maze.
As already said, the minigame is a 2D maze: the square is located to a home position and, by collecting some coloured dots that will open (or close) some doors, it will be possible to reach the winning position, which will finally unlock the door to the next room.
In particular, it is possible to move the square by simply pushing the **W**,**A**,**S**,**D keys** on the keyboard; the square can touch and “slide” on the walls of the maze without passing through it. It may happen also that during the minigame

you could get stuck in some parts of the maze: in that case it is possible to restart the minigame from the beginning by pushing the **space bar** on the keyboard.
During the minigame session it is possible to see some legs of the robot moving as if they were typing on the tablet, it’s the robot itself that is interacting with the tablet.

4. The initial mode of the robot is the spider mode. During the travel of Randy, it will be able to switch its mode from the spider mode to the sphere mode, and the action is made possible by pushing the button “**1**” on the keyboard: Randy will put its legs inside its body, and it will be able to roll from a room to another.
To come back to the spider mode status, it is sufficient to push again the same button.

5. At a certain time of the game, Randy will have to walk (or roll) inside a dark room, with damaged lights that do not work properly, so it will be difficult for the user go ahead through the dark: some hints will appear on the screen in order to help the user. In fact, Randy is a very special robot, and it is full of resources: thanks to this fact, it is capable to transform its front and back “eye” into a flashlight.
Going into details, the user can push the button “**T**” to switch on and off the torch: after having done it, two beams of light will start from the to lenses of the head of the robot, so it will be possible not only to illuminate the space in front of Randy, but also its back.
Furthermore, the torch (Randy’s head) can rotate of 360 degrees; as the hint on the screen will suggest it, it is possible to perform this particular action by pushing the buttons “**Q**” and “**E**”, to rotate the torch from left to right and vice versa respectively.

While walking through the corridor of the starship, Randy will get into a particular room full of flying boxes that were originally inside the containers: apparently, this room seems not to have a particular meaning, except for the fact that we wanted to show that there is no gravity in the whole spaceship, and that Randy can “walk” on the floor thanks to its special spyder legs that generate a magnetic field. To make it possible we had to use a particular library called **Cannon** that let us represent the physics of the objects, as well as the change in directions of the boxes of this room when Randy impacts them or when they hit the walls.

In order to give a possible story to our game, we thought as Randy as the last survivor of the starship: in fact, the entire crew decided to come back to the Earth, forgetting of Randy. This scenario can be clearly seen in a particular room, in which when Randy will get in, a rescue pod will leave that room, and will start to travel to the Earth; Randy will be able to see this sad scene through the window of the room, and the user too, thanks to the zoom-in towards the window that we implemented.

Proceeding in the game, an alarm will start to ring warning Randy that the starship is entering into the gravitational field of the Earth; to avoid Randy to be hurt, the AI of the starship will tell him to move towards the savage room, to enter in the rescue cabin and to stay in it until the spaceship will have re-established its gravity with respect to the Earth's one. An animation will start, and when the starship will be out of danger, the user will be able to control Randy again.

Due to the fact that at this point the spaceship is affected by the Earth's gravity, from now on Randy will be able to switch between the spyder mode and the sphere mode just by pushing the button "**1**" on the keyboard. During this mode some functionalities are limited: for example, Randy can use the flashlight only if it is not moving, or can roll only if the light of the torch is off (by the way, some hints will appear on the screen to help the user in these particular situations).

Once Randy and the starship are off the hook, the user will have to navigate (for the last time) through the corridor and reach the last sliding door (that must be unlocked by using the tablet close to it). At this point, the user will see Randy walking into the next room, but the camera will not follow him: in fact, the door will close, and while a final music will start to play, the screen will become darker and darker as to declare the end of the game. Finally, credits will appear on the screen.

FUTURE DEVELOPMENTS

The game has been thought to be a short Demo of what it could be a full game: in fact, it is possible to extend it just by imagining how many different scenarios and kind of "levels" can be built around the story of Randy.
For example, by passing to the next level, Randy could take control of the starship and travel to the Earth in order to take over the human race (due to the fact that the crew forgot him in the spaceship), so there can be a level in which the user has to drive the starship by avoiding asteroids and martians; another scenario can be the one in which the poor Randy decides to take the starship back to the Earth in order to make it repaired and ready for the next travel.