

Phoenix space adventure documentation

Isabella Claire Adriani

September 2019

Contents

1	Introduction	1
2	Used libraries, tools and models	1
3	Technical aspects	4
3.1	Phoenix	4
3.2	Blooming	4
3.3	God rays	4
3.4	Fire	4
3.5	Explosions	5
3.6	balls	5
4	Interactions	5

1 Introduction

This game was developed by AngelWings, of which I am the only member, for the Interactive Graphics course's final project. The goal is to gather as many eggs as possible. In first scene there are many balls, each of them contains eggs. To get the eggs you must enter a ball of you choosing first. Once enter the ball the scene will change and obstacles will appear. The user must avoid the obstacles. The first obstacles are meteorites and asteroids. Then once entered the black hole, the crystals must be avoided. At the end of the journey in the chosen ball, a chest of eggs will appear. Collect the eggs and exit the ball. Repeat the process, if wanted, for all the 500 balls.

I used a combination of basic WebGL and Three.js. The animations are all done in WebGL.

2 Used libraries, tools and models

Libraries:

- Three.js (I got all the files from the three.js-master folder which I downloaded from the Three.js official site. Some of the libraries in the folder are used by the following):

- lib/GLTFLoader.js
- lib/Reflector.js
- lib/Refractor.js
- lib/EffectComposer.js
- lib/RenderPass.js
- lib/ShaderPass.js
- lib/UnrealBloomPass.js
- lib/LuminosityHighPassShader.js
- lib/CopyShader.js
- lib/Fire.js
- lib/GodRaysShader.js

Models:

- models/phoenix_bird
- models/meteorite
- models/asteroid_01
- models/meteorite_2
- models/meteorite_3
- models/low_poly_crystals
- models/dragon_chest

Skyboxes:

- models/skyboxes/jupiter.jpg
- models/skyboxes/galaxy/GalaxyTex_PositiveX.png
- models/skyboxes/galaxy/GalaxyTex_NegativeX.png
- models/skyboxes/galaxy/GalaxyTex_NegativeY.png
- models/skyboxes/galaxy/GalaxyTex_PositiveZ.png

Textures:

- models/textures/circle.png
- models/textures/spark1.png

- models/textures/colourscaenew.jpg
- models/textures/night_sky.jpg

Fonts:

- models/fonts/digital-dream/DIGITALDREAMFAT.ttf
- models/fonts/questrial/Questrial-Regular.ttf
- models/fonts/behtix/Behtix/Behtix.ttf
- models/fonts/los-angeles/Los-Angeles.ttf
- models/fonts/trackers/Trackers.ttf

Music and sound effects:

- models/sound.effects/423805__tyops__game-theme-4.wav
- models/sound.effects/173859__jivatma07__j1game-over-mono.wav
- models/sound.effects/235968__tommccann__explosion-01.wav
- models/sound.effects/394424__inspectorj__bamboo-swing-c9.wav
- models/sound.effects/270332__littlerobotsoundfactory__hit-03.wav
- models/sound.effects/233558__waveplay-old__softer-car-screch.wav
- models/sound.effects/322176__liamg-sfx__background-fire.wav

Keyboard buttons images:

- models/keyboard_buttons/W1.png
- models/keyboard_buttons/S1.png
- models/keyboard_buttons/A1.png
- models/keyboard_buttons/D1.png
- models/keyboard_buttons/R1.png
- models/keyboard_buttons/F13.png
- models/keyboard_buttons/B1.png
- models/keyboard_buttons/K1.png
- models/keyboard_buttons/Up-Arrow1.png
- models/keyboard_buttons/Down-Arrow1.png
- models/keyboard_buttons/C1.png
- models/keyboard_buttons/T1.png
- models/keyboard_buttons/M1.png

Some pieces of the code are inspired by Three.js examples: particles, blooming, god rays, fire and reflection.

3 Technical aspects

3.1 Phoenix

The phoenix is managed by its own file. The model has been imported without animations. In the file there are all the points and functions to execute the animations. I calculated all the points so that, through linear interpolation, the movement is as fluid and as realistic as possible. Note that the game and model itself are heavy, therefore this may slow down the movement so much that it might not be as fluid. A suggestion would be to charge the computer and not have many programs, if any, running at the same time. The phoenix has 87 bones. I made most of them move. Thanks to the blooming effect the phoenix illuminates itself. During the flight animation, I found the right points only for one wing; the other just mirrors them.

3.2 Blooming

In the full version there is selective blooming. It's most notable in the black hole scene. This post processing, inspired by three.js examples, makes object irradiate light.

3.3 God rays

When entering and when exiting a ball the god rays will appear from the chosen ball. To exit the ball just collect the eggs. This is done to transition between moving in and out of a ball. This post processing was also inspired by three.js examples.

3.4 Fire

The fire is inspired by three.js examples. It's very present in the entire game. Both the loading and the game over texts go on fire. I created the fire on a plane and then I used the text as texture for the fire.

For the commands and the tutorial page there is an ice ball with two sources. To make them I made fire with blue flames. The two sources form W for wings.

When the phoenix has no more lives the phoenix goes on fire. I created a cube and located it at the phoenix position and then I put the fire in the cube.

Last but not least, the phoenix spits fire when the user holds "k". I created a cylinder with the fire in it. I positioned the fire in front of the phoenix's beak. When "k" is released the fire is removed from the scene. Still, when the user interacts with the phoenix by moving it, the fire is also moved. This way, when the fire is added to the scene again the fire starts from right position and rotation. When a meteorite or a crystal is hit by the fire for a certain amount of time it will explode.

3.5 Explosions

When an obstacle explodes it will be removed from the scene and particles will appear in the same position. Then each particle will move in a random direction towards the exterior. The same algorithm is used in the page to start game, but the position is the origin, where the user is.

3.6 balls

Balls are positioned and moved randomly in space. The texture of the balls reflect the texture of the scene's background.

4 Interactions

The first interaction would be to choose the version of the game: lite or full. The full version as opposed to the lite version blooms the objects including the phoenix, and it plays music and sound effects. The version can be changed later from the pause page.

Then it is possible to choose the difficulty of the game: easy, medium or hard. Each determines a different number of obstacles and their speed. Hit the play button and the game will start.

In the first phase, in which you have to choose a ball to enter, the user can move freely by holding the following keys: forward ("w"), backwards ("s"), left ("a"), right ("d"), up ("r") and down ("f"). It is also possible to change the velocity of the phoenix with up and down arrows.

Once the user enters a new ball the scene will change and the user will be able to go left ("a") and right ("d") to avoid the obstacles. This is also the case for the black hole scene. This last scene appears as soon as the user arrives at the black hole entrance. When the phoenix gets hit by an obstacle it will lose a life. When there are no more lives it's game over. The phoenix will go on fire.

By holding "k" the phoenix spits fire. This happens in the first phase, in the ball and in the black hole. In last two though if an obstacle is hit by this fire, after a given time, the obstacle will explode. Note that after a number of explosions the fire will not work anymore. So it's better to use only if necessary.

At any given moment during the game it is possible to access the commands, tutorial, pause and exit pages by pressing the buttons below on the left and right side, or from the pause page, or from the keys "c", "t", "p" and "u".

When exiting the game the phoenix will bow and it is possible to start a new game by pressing the button below.

In any moment the music and sound effects can be muted or unmuted with "m". This can also be done from the pause page. From this page music and sound effects volumes can be changed.

Also if "b" is pressed, the full screen will be activated or deactivated. Note this may not work if f11 is pressed. This is because f11 is OS function therefore I can't control it.