

Computer Graphics

Final Project

StarLord

Luis Eduardo Barajas Pérez A01370934

Daniel Charua A01017419

# Ideas

**First Idea: Flappy Bird 3D**

Objective/Idea of the project

The project will be a kind of flappy bird but instead of being 2D and in a phone controlled by taps would be in 3D controlled with keys. The main camera would be like mounted on the back of the bird and you would have to pass thourgh pipes and other kinds of obstacles for accumulate points, in order to make a top score. Another difference will be that the bird in the traditional 2D game only goes up and down, while in this version it will move up, down, left, right and diagonally.

How the user will interact with the project

As mentioned before, the user will interact with the project by controlling the main character of the game, in order to survive. The controls will be with the arrows of the keyboard of a computer.

What controls/shaders/additional libraries you are going to use or need

- SFX Class

- Tween

- Assets, objects of pipes obstacles and bird

References to similar projects

- https://www.cooljuegos.com/juego-en-linea/flappy-3d/

- https://www.kongregate.com/games/hamster18/flappy-bird-3d

- https://www.minijuegos.com/juego/flappy-bird-3d

**Second Idea: Starfox**

Objective/Idea of the project

The project will be very similar to StarFox game. This is a very popular first shooter 90's game. Our project will have also this functionality, gamebility, and graphics.

How the user will interact with the project

The user will interact with the mouse and keyboard. The mouse will be to move his field of view and click to shoot. The keyboard keys will be to move and jump around.

What controls/shaders/additional libraries you are going to use or need

- Threejs-fps-controls

- Threejs Audio

- Shaders

- Lights

- PointerLockControls

- SFX Class

- Tween

- Assets, objects of map

- References to similar projects

- Starfox 64

**Third Idea: Augmented Reality Showroom**

Objective/Idea of the project

The project will be, as the name suggests, an augmented reality showroom to display any type of object a company may be interested in displaying. Cars, furniture, etc.

How the user will interact with the project

The user will interact with the mouse and keyboard. The mouse will be to move around the object and the keys in keyboard to navigates thorugh objects

What controls/shaders/additional libraries you are going to use or need

- SFX Class

- Tween

- Assets, objects of for the elements of each showroom

References to similar projects

- Hyundai AR Showroom

- Raumplus AR Showroom

# Selected Idea

The selected idea is the Starfox game because we and the teacher believe that it has a good level of complexity, it uses the required things by the specifications and it will be fun to make and play.

The player will control the arrowing spaceship piloted by captain star fox as he navigates through space avoiding enemies and objects in the journey.

# Objective

The goal of this web application using the JavaScript Framework Three.js, is to learn how to implement computer graphics using a web technology. And to use all the topics seen in class, while making it fun to make and play.

# Requirements

### Functionals

The functional requirements for this project will be:

1. **Lobby:** The player will enter to the lobby at first glance
2. **High Score Lobby:**The player will be able to see the highscore at the lobby
3. **Play Button:** The player will be able to click the play button to start a new game
4. **Horizontal Movement:** The player will be able to move in the horizontal axis the spaceship with the arrow keys
5. **Shoot:** The player will be able to shoot pressing the click button in the mouse
6. **Rotation Movement:** The spaceship will rotate always following the mouse movement from the player
7. **HUD:** The game will always display to the player the hud (heads-up display) having the current score of the game, the live that it has remaining and a hint to the user to press R if it wants to restart the game
8. **R to restart game**: The player will be able to press R whenever it wants to restart the current game.
9. **Enemy death**: The enemy will die if anything of the following happens:
   1. A bullet shot by player hits it
   2. Leaves the camera’s view ponit.
   3. It collides with the player
10. **Score will increase:** The score will increment if a bullet that the player shot collides an enemy or the spaceship of the player collides with the Score Powerup.
11. **Player’s health will decrease:** The health of the player will decrease if anything of the following happens:
    1. A bullet shot by enemy hits it
    2. It collides with enemy or asteroid
12. **Player’s health will increase:** The health of the player will increase if it collides with the Med Kit Powerup.
13. **Saved High score:** The score of the match will be saved as high score if it is more than the current high score saved.

### Non Functionals

The non functional requirements for this project will be:

1. **Web performance:** The web game will not see a decrease in performance although it is a very resource consuming project to the web browser because of all assets and calculations it has to load and make.
2. **Game Usability:** The usability of the game will be intuitive for the player and follow good design standard
3. **Game Availability:** The game will be available at all the time letting the user play how many times it wants

# How to match the requirements

The architecture implemented in the project is based on JavaScript objects and procedures to make it more reliable and user friendly with the use of asynchronous calls to load all the heavy models. The necessary technologies for this will be: HTML/CSS and JavaScript with two frameworks: Threejs and jQuery.

### Functional

1. **Lobby:** A lobby was created for the player to start the experience with the web game
2. **High Score Lobby:** The last saved highscore is displayed at the top right corner
3. **Play Button:** A play button is displayed at the center of the screen for the user to click when it is ready to start playing
4. **Horizontal Movement:** An event listener is implemented so the window listens when the user clicks the right or left arrow of the keyboard.
5. **Shoot:** An event listener is implemented so the window listens when the user clicks the mouse left button.
6. **Rotation Movement:** An event listener is implemented so the window listens for the movement of the mouse inside the viewport. then with this information it does the calculations for the rotation of player’s spaceship.
7. **HUD:** A HUD is displayed at the top left corner with all the information described, It is displayed all the time when the player is in the game
8. **R to restart game**: An event listener is implemented so the window listens when the user presses the R key in the keyboard and restarts the game.
9. **Enemy death**: A collider is added to each enemy so in each frame it’s validating if it collides with anything that can be responsible for its death
   1. A bullet shot by player
   2. It collides with the player

Also at each frame it’s validated that the enemy is still at the view sight.

1. **Score will increase:** At each frame is validated if any collider of a bullet shot by the player has collided with a enemy’s collider or if the player’s spaceship collider has intersected with a Score powerup collider.
2. **Player’s health will decrease:** At each frame the game objects colliders are checked form collision with the player, if a bullet, enemy or an asteroid collides with the prayer points are deducted from the live variable
3. **Player’s health will increase:** At each frame the Med Kit Powerup colliders are checked form collision with the player if there is a collision points are increased from the live variable with the Med Kit Powerup.
4. **Saved High score:** At the end of the game the score is checked with the high score, if the current score is higher than the saved high score, the new high score is saved in the local web storage replacing the old one, this high score is load every time the player opens the game.

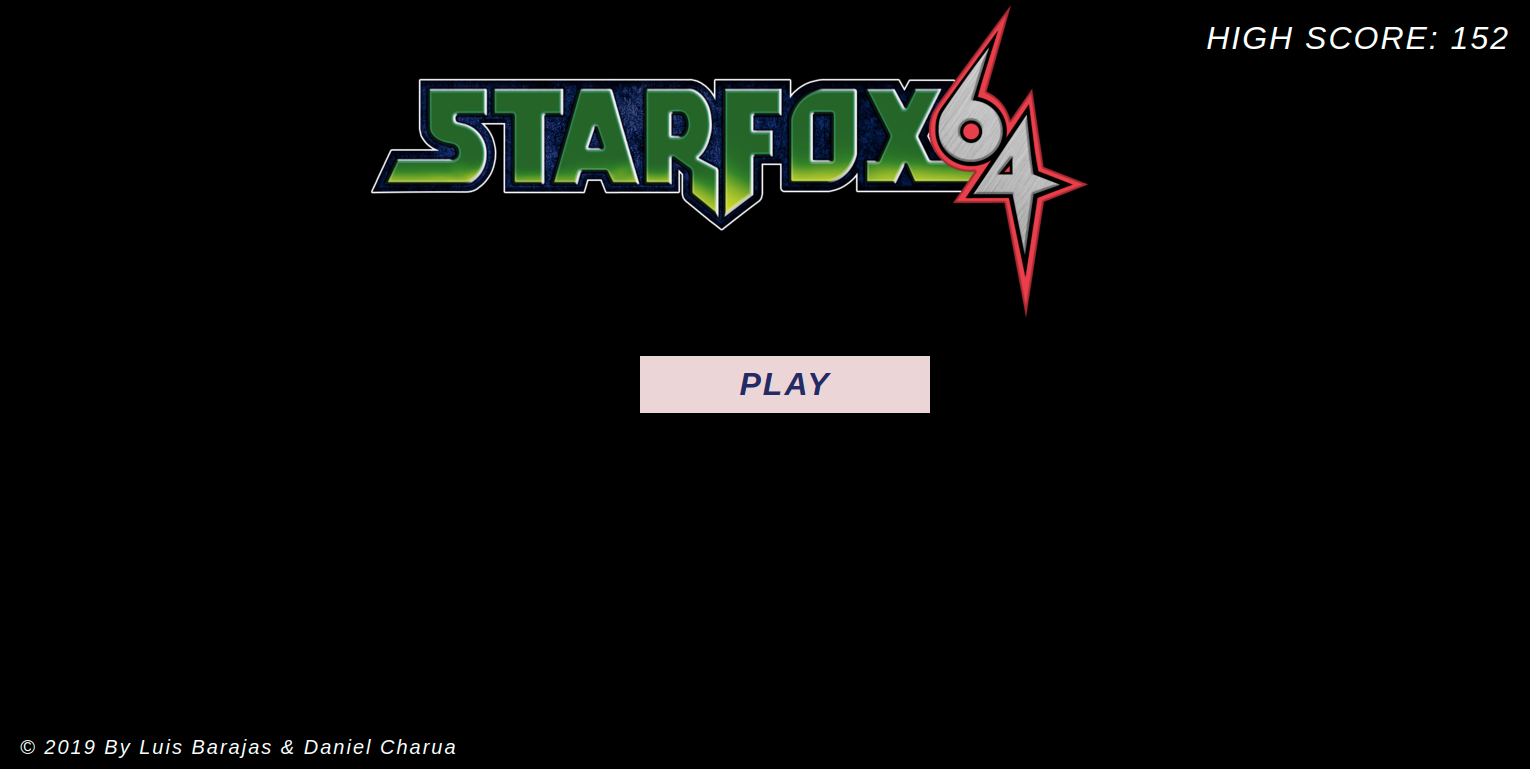
Non Functional

1. **Web performance:** To keep stable performance medium size assets were selected, additional game objects are removed as soon as they disappear from the scene, and deleted from the data structure that holds them.
2. **Game Usability:** Controls were made with a very easy mouse pointer that shoots precisely where the mouse is pointing, this was made possible by calculating the necessary geometry from a given (x, y) coordinate.
3. **Game Availability:** since there is no online playing the game did not require a server to host connections, It can be run by any computer via localhost, or it can be uploaded to the cloud with autoscaling for commercial release.

# Game images

splash



Lobby 

Game play



# Assets

The assets were taken from the following link https://www.models-resource.com/nintendo\_64/starfox64/

The main player is the arrowing spaceship



Wolf II

