

PLANET WARS

ALEX BERMUDEZ

PAULA PAZ

ASIER POZO

PLANET WARS



1. Project introduction	4
○ Brief description of the game	4
○ Project Objectives	4
○ Tools used (IDE, llenguatge, llibreries, frameworks...).	4
2. Architecture and Development	5
○ Description of the source code structure	5
3. Database Creation	6
○ Creation of the database	6
○ Creation of the tables	7
4. Web outline	11
○ Folders and contents	11
○ Web photos	12
5. User Manual	14
6. Blender	15
7. Soundtrack	17
8. XML and HTML into the web	18
9. Class Diagram	21



1. Project introduction

○ Brief description of the game

Planet Wars is a survival game where you must collect resources and win intergalactic battles to survive. In future expansions, we will add more planets to obtain more and different resources.

○ Project Objectives

The primary goal of this project is to connect and integrate all the knowledge we have acquired throughout the course. Specifically, it involves linking a database with a program developed in Java.

Additionally, this project allows us to expand our expertise in database creation and management. It also involves extracting XML files from the game and then incorporating them into our website using JavaScript. Through this process, we enhance our understanding of data handling and web development while applying practical skills in various technologies.

○ Tools used (IDE, llenguatge, llibreries, frameworks...).

Language: Java, Javascript, HTML, CSS, XML, MYSQL

IDE: Visual Studio Code, Eclipse

Libraries:

- Timer
- ArrayList
- Scanner
- TimerTask
- JFrame
- JPanel
- JButton
- IOException
- Random

Other: Github



2. Architecture and Development

- Description of the source code structure

We have divided all the big classes into different files. All those files have their own attributes and methods.

We have divided our website as the main code, in four different big groups.

All the gameplay methods are done in the class battle, while the graphic interface was written in Juego.java. There are other classes that only have their attributes, the getters and setters. Those classes are also important to create the enemy and our own army.

We also have a class where we have all the methods related to the Database. That classed is called DatabaseManager. There we have the connection to the database and all the inserts and updates necessary to play the game and save all the events on our database.

Another important file is Variables.java. There we have the chances to attack, to be attacked, the costs of every weapon, evolution...

```

    <div>
        <ul style="list-style-type: none; padding-left: 0;">
            <li><span style="color: #ccc; font-size: 1em; font-weight: bold; margin-right: 0.5em; position: relative; top: -1px; left: -1px; border: 1px solid black; padding: 0 2px; border-radius: 3px; background-color: white; opacity: 0.8; z-index: 1;*>▽</span> src</li>
            <li><span style="color: #ccc; font-size: 0.8em; margin-right: 0.5em; position: relative; top: -1px; left: -1px; border: 1px solid black; padding: 0 2px; border-radius: 3px; background-color: white; opacity: 0.8; z-index: 1;*>|></span> images</li>
            <li><span style="color: red; font-size: 0.8em; margin-right: 0.5em; position: relative; top: -1px; left: -1px; border: 1px solid black; padding: 0 2px; border-radius: 3px; background-color: white; opacity: 0.8; z-index: 1;*>J</span> ArmoredShip.java</li>
            <li><span style="color: red; font-size: 0.8em; margin-right: 0.5em; position: relative; top: -1px; left: -1px; border: 1px solid black; padding: 0 2px; border-radius: 3px; background-color: white; opacity: 0.8; z-index: 1;*>J</span> Battle.java</li>
            <li><span style="color: #ccc; font-size: 0.8em; margin-right: 0.5em; position: relative; top: -1px; left: -1px; border: 1px solid black; padding: 0 2px; border-radius: 3px; background-color: white; opacity: 0.8; z-index: 1;*>Ξ</span> battleReport.xsl</li>
            <li><span style="color: red; font-size: 0.8em; margin-right: 0.5em; position: relative; top: -1px; left: -1px; border: 1px solid black; padding: 0 2px; border-radius: 3px; background-color: white; opacity: 0.8; z-index: 1;*>J</span> BattleShip.java</li>
            <li><span style="color: red; font-size: 0.8em; margin-right: 0.5em; position: relative; top: -1px; left: -1px; border: 1px solid black; padding: 0 2px; border-radius: 3px; background-color: white; opacity: 0.8; z-index: 1;*>J</span> DatabaseManager.java</li>
            <li><span style="color: red; font-size: 0.8em; margin-right: 0.5em; position: relative; top: -1px; left: -1px; border: 1px solid black; padding: 0 2px; border-radius: 3px; background-color: white; opacity: 0.8; z-index: 1;*>J</span> Defense.java</li>
            <li><span style="color: red; font-size: 0.8em; margin-right: 0.5em; position: relative; top: -1px; left: -1px; border: 1px solid black; padding: 0 2px; border-radius: 3px; background-color: white; opacity: 0.8; z-index: 1;*>J</span> HeavyHunter.java</li>
            <li><span style="color: red; font-size: 0.8em; margin-right: 0.5em; position: relative; top: -1px; left: -1px; border: 1px solid black; padding: 0 2px; border-radius: 3px; background-color: white; opacity: 0.8; z-index: 1;*>J</span> IonCannon.java</li>
            <li><span style="color: red; font-size: 0.8em; margin-right: 0.5em; position: relative; top: -1px; left: -1px; border: 1px solid black; padding: 0 2px; border-radius: 3px; background-color: white; opacity: 0.8; z-index: 1;*>J</span> Juego.java</li>
            <li><span style="color: red; font-size: 0.8em; margin-right: 0.5em; position: relative; top: -1px; left: -1px; border: 1px solid black; padding: 0 2px; border-radius: 3px; background-color: white; opacity: 0.8; z-index: 1;*>J</span> LightHunter.java</li>
            <li><span style="color: red; font-size: 0.8em; margin-right: 0.5em; position: relative; top: -1px; left: -1px; border: 1px solid black; padding: 0 2px; border-radius: 3px; background-color: white; opacity: 0.8; z-index: 1;*>J</span> Main.java</li>
            <li><span style="color: red; font-size: 0.8em; margin-right: 0.5em; position: relative; top: -1px; left: -1px; border: 1px solid black; padding: 0 2px; border-radius: 3px; background-color: white; opacity: 0.8; z-index: 1;*>J</span> MilitaryUnit.java</li>
            <li><span style="color: red; font-size: 0.8em; margin-right: 0.5em; position: relative; top: -1px; left: -1px; border: 1px solid black; padding: 0 2px; border-radius: 3px; background-color: white; opacity: 0.8; z-index: 1;*>J</span> MissileLauncher.java</li>
            <li><span style="color: red; font-size: 0.8em; margin-right: 0.5em; position: relative; top: -1px; left: -1px; border: 1px solid black; padding: 0 2px; border-radius: 3px; background-color: white; opacity: 0.8; z-index: 1;*>J</span> Planet.java</li>
            <li><span style="color: red; font-size: 0.8em; margin-right: 0.5em; position: relative; top: -1px; left: -1px; border: 1px solid black; padding: 0 2px; border-radius: 3px; background-color: white; opacity: 0.8; z-index: 1;*>J</span> PlasmaCannon.java</li>
            <li><span style="color: red; font-size: 0.8em; margin-right: 0.5em; position: relative; top: -1px; left: -1px; border: 1px solid black; padding: 0 2px; border-radius: 3px; background-color: white; opacity: 0.8; z-index: 1;*>J</span> ResourceException.java</li>
            <li><span style="color: red; font-size: 0.8em; margin-right: 0.5em; position: relative; top: -1px; left: -1px; border: 1px solid black; padding: 0 2px; border-radius: 3px; background-color: white; opacity: 0.8; z-index: 1;*>J</span> Ship.java</li>
            <li><span style="color: red; font-size: 0.8em; margin-right: 0.5em; position: relative; top: -1px; left: -1px; border: 1px solid black; padding: 0 2px; border-radius: 3px; background-color: white; opacity: 0.8; z-index: 1;*>J</span> Variables.java</li>
        </ul>
    </div>

```

```

    <div>
        <ul style="list-style-type: none; padding-left: 0;">
            <li><span style="color: #ccc; font-size: 1em; font-weight: bold; margin-right: 0.5em; position: relative; top: -1px; left: -1px; border: 1px solid black; padding: 0 2px; border-radius: 3px; background-color: white; opacity: 0.8; z-index: 1;*>▽</span> webProjecte</li>
            <li><span style="color: #ccc; font-size: 0.8em; margin-right: 0.5em; position: relative; top: -1px; left: -1px; border: 1px solid black; padding: 0 2px; border-radius: 3px; background-color: white; opacity: 0.8; z-index: 1;*>|></span> About Us</li>
            <li><span style="color: #ccc; font-size: 0.8em; margin-right: 0.5em; position: relative; top: -1px; left: -1px; border: 1px solid black; padding: 0 2px; border-radius: 3px; background-color: white; opacity: 0.8; z-index: 1;*>▽</span> Batallas</li>
            <li><span style="color: red; font-size: 0.8em; margin-right: 0.5em; position: relative; top: -1px; left: -1px; border: 1px solid black; padding: 0 2px; border-radius: 3px; background-color: white; opacity: 0.8; z-index: 1;*>#</span> batallas.css</li>
            <li><span style="color: orange; font-size: 0.8em; margin-right: 0.5em; position: relative; top: -1px; left: -1px; border: 1px solid black; padding: 0 2px; border-radius: 3px; background-color: white; opacity: 0.8; z-index: 1;*>◇</span> batallas.html</li>
            <li><span style="color: red; font-size: 0.8em; margin-right: 0.5em; position: relative; top: -1px; left: -1px; border: 1px solid black; padding: 0 2px; border-radius: 3px; background-color: white; opacity: 0.8; z-index: 1;*>JS</span> batallas.js</li>
            <li><span style="color: orange; font-size: 0.8em; margin-right: 0.5em; position: relative; top: -1px; left: -1px; border: 1px solid black; padding: 0 2px; border-radius: 3px; background-color: white; opacity: 0.8; z-index: 1;*>◇</span> battle1.html</li>
            <li><span style="color: orange; font-size: 0.8em; margin-right: 0.5em; position: relative; top: -1px; left: -1px; border: 1px solid black; padding: 0 2px; border-radius: 3px; background-color: white; opacity: 0.8; z-index: 1;*>◇</span> battle2.html</li>
            <li><span style="color: orange; font-size: 0.8em; margin-right: 0.5em; position: relative; top: -1px; left: -1px; border: 1px solid black; padding: 0 2px; border-radius: 3px; background-color: white; opacity: 0.8; z-index: 1;*>◇</span> battle3.html</li>
            <li><span style="color: orange; font-size: 0.8em; margin-right: 0.5em; position: relative; top: -1px; left: -1px; border: 1px solid black; padding: 0 2px; border-radius: 3px; background-color: white; opacity: 0.8; z-index: 1;*>◇</span> battle5.html</li>
            <li><span style="color: blue; font-size: 0.8em; margin-right: 0.5em; position: relative; top: -1px; left: -1px; border: 1px solid black; padding: 0 2px; border-radius: 3px; background-color: white; opacity: 0.8; z-index: 1;*>☒</span> images4.png</li>
            <li><span style="color: #ccc; font-size: 0.8em; margin-right: 0.5em; position: relative; top: -1px; left: -1px; border: 1px solid black; padding: 0 2px; border-radius: 3px; background-color: white; opacity: 0.8; z-index: 1;*>|></span> Principal</li>
            <li><span style="color: #ccc; font-size: 0.8em; margin-right: 0.5em; position: relative; top: -1px; left: -1px; border: 1px solid black; padding: 0 2px; border-radius: 3px; background-color: white; opacity: 0.8; z-index: 1;*>|></span> Tutorial</li>
        </ul>
    </div>

```



3. Database Creation

○ Creation of the database

The first step to create a database is to download and install MYSQL Workbench

MySQL Community Downloads

Login Now or Sign Up for a free account.

An Oracle Web Account provides you with the following advantages:

- Fast access to MySQL software downloads
- Download technical White Papers and Presentations
- Post messages in the MySQL Discussion Forums
- Report and track bugs in the MySQL bug system

[Login »
using my Oracle Web account](#) [Sign Up »
for an Oracle Web account](#)

MySQL.com is using Oracle SSO for authentication. If you already have an Oracle Web account, click the Login link. Otherwise, you can signup for a free account by clicking the Sign Up link and following the instructions.

[No thanks, just start my download.](#)

After this I entered MYSQL Workbench with a username and password and I entered the following commands to create and use the database.

```
SQL File 5* x
CREATE DATABASE PlanetWars;
USE PlanetWars;
```

PLANET WARS



○ Creation of the tables

To create the tables we have used the following commands:

```
3 • CREATE TABLE Planet_stats (
4     planet_id INT PRIMARY KEY,
5     name VARCHAR(100),
6     resource_metal_amount INT,
7     resource_deuterion_amount INT,
8     technology_defense_level INT,
9     technology_attack_level INT,
10    battles_counter INT,
11    missile_launcher_remaining INT,
12    ion_cannon_remaining INT,
13    plasma_canon_remaining INT,
14    light_hunter_remaining INT,
15    heavy_hunter_remaining INT,
16    battleship_remaining INT,
17    armored_ship_remaining INT
18 );
```

```
30 • CREATE TABLE Battle_log (
31     planet_id INT NOT NULL,
32     num_battle INT NOT NULL,
33     num_line INT NOT NULL,
34     log_entry TEXT,
35     PRIMARY KEY (planet_id, num_battle, num_line),
36     FOREIGN KEY (planet_id, num_battle) REFERENCES Battle_stats(planet_id, num_battle)
37 );
```

```
39 • CREATE TABLE Planet_battle_defense (
40     planet_id INT NOT NULL,
41     num_battle INT NOT NULL,
42     missile_launcher_built INT,
43     missile_launcher_destroyed INT,
44     ion_cannon_built INT,
45     ion_cannon_destroyed INT,
46     plasma_canon_built INT,
47     plasma_canon_destroyed INT,
48     PRIMARY KEY (planet_id, num_battle),
49     FOREIGN KEY (planet_id, num_battle) REFERENCES Battle_stats(planet_id, num_battle)
50 );
```

PLANET WARS



```
52 • Ⓜ CREATE TABLE Planet_battle_army (
53     planet_id INT NOT NULL,
54     num_battle INT NOT NULL,
55     light_hunter_built INT,
56     light_hunter_destroyed INT,
57     heavy_hunter_built INT,
58     heavy_hunter_destroyed INT,
59     battleship_built INT,
60     battleship_destroyed INT,
61     armored_ship_built INT,
62     armored_ship_destroyed INT,
63     PRIMARY KEY (planet_id, num_battle),
64     FOREIGN KEY (planet_id, num_battle) REFERENCES Battle_stats(planet_id, num_battle)
65 );
```

```
67 • Ⓜ CREATE TABLE Enemy_army (
68     planet_id INT NOT NULL,
69     num_battle INT NOT NULL,
70     light_hunter_threat INT,
71     light_hunter_destroyed INT,
72     heavy_hunter_threat INT,
73     heavy_hunter_destroyed INT,
74     battleship_threat INT,
75     battleship_destroyed INT,
76     armored_ship_threat INT,
77     armored_ship_destroyed INT,
78     PRIMARY KEY (planet_id, num_battle),
79     FOREIGN KEY (planet_id, num_battle) REFERENCES Battle_stats(planet_id, num_battle)
80 );
```

PLANET WARS



To insert data from the Java program, I did the following:

First, I created a separate file called DatabaseManager.java in which I created all the inserts needed to insert data into each of the tables.

```
public class DatabaseManager {  
    private static final String URL = "jdbc:mysql://localhost:3306/planetwars";  
    private static final String USER = "root";  
    private static final String PASSWORD = "1234";  
  
    public static Connection getConnection() throws SQLException {  
        return DriverManager.getConnection(URL, USER, PASSWORD);  
    }  
  
    // Inserta un registro en Planet_stats  
    public static void insertPlanetStats(  
        int planet_id, String name, int resource_metal_amount, int resource_deuterion_amount,  
        int technology_defense_level, int technology_attack_level, int battles_counter,  
        int missile_launcher_remaining, int ion_cannon_remaining, int plasma_cannon_remaining,  
        int light_hunter_remaining, int heavy_hunter_remaining, int battleship_remaining, int armored_ship_remaining  
    ) {  
        String sql = "INSERT INTO Planet_stats (planet_id, name, resource_metal_amount, resource_deuterion_amount, technology_defense_level, technology_attack_level, battles_counter, missile_launcher_remaining, ion_cannon_remaining, plasma_cannon_remaining, light_hunter_remaining, heavy_hunter_remaining, battleship_remaining, armored_ship_remaining);  
        try (Connection conn = getConnection();  
             PreparedStatement stmt = conn.prepareStatement(sql)) {  
            stmt.setInt(1, planet_id);  
            stmt.setString(2, name);  
            stmt.setInt(3, resource_metal_amount);  
        }  
    }  
}
```

Here you can see an example of an insert into the database

```
Run | Debug | Run main | Debug main  
public static void main(String[] args) throws ResourceException {  
    DatabaseManager.insertPlanetStats(  
        planet_id:1, name:"Planeta Tierra", resource_metal_amount:10000, resour..5000, 2, 3, 1,  
        missile_launcher_remaining:5, ion_cannon_remai..2, 1, 10, 4, 2, 1  
    );  
    // Insertar estadísticas de la batalla  
    //DatabaseManager.insertBattleStats(planetAddress, numBattle, metalAcquired, deuteriumAcquired);  
  
    // Insertar log de la batalla (puedes llamar esto en un bucle para cada línea del log)  
    //DatabaseManager.insertBattleLog(planetAddress, numBattle, 1, "La batalla comenzó...");  
  
    // Insertar defensas construidas/destruídas  
    //DatabaseManager.insertPlanetBattleDefense(planetAddress, numBattle, missileLauncherBuilt, missileLauncherDestroyed, ionCannonBuilt, ionCannonDestroyed, plasmaCannonBuilt, plasmaCannonDestroyed);  
  
    // Insertar unidades destruidas del ejército del planeta  
    //DatabaseManager.insertPlanetBattleArmy(planetAddress, numBattle, lightHunterDestroyed, heavyHunterDestroyed, battleShipDestroyed, armoredShipDestroyed);  
  
    // Insertar unidades destruidas del ejército enemigo  
    //DatabaseManager.insertEnemyArmy(planetAddress, numBattle, lightHunterDestroyed, heavyHunterDestroyed, battleShipDestroyed, armoredShipDestroyed);
```

By running that test insert and doing a select, you can see how the data has been inserted into the database:

The screenshot shows the MySQL Workbench interface. At the top, there is a command line with the following text:
1 • use planetwars;
2 select *from planet_stats;
Below the command line is a result grid. The grid has the following columns: planet_id, name, resource_metal_amount, resource_deuterion_amount, technology_defense_level, technology_attack_level, battles_counter, missile_launcher_remaining, ion_cannon_remaining, plasma_cannon_remaining, light_hunter_remaining, heavy_hunter_remaining, battleship_remaining, armored_ship_remaining. There is one row of data:
1 Planeta Tierra 10000 5000 2 NULL 3 NULL
The 'name' column contains 'NULL' because the value was not explicitly provided in the insert statement.

	planet_id	name	resource_metal_amount	resource_deuterion_amount	technology_defense_level	technology_attack_level	battles_counter	missile_launcher_remaining	ion_cannon_remaining	plasma_cannon_remaining	light_hunter_remaining	heavy_hunter_remaining	battleship_remaining	armored_ship_remaining
▶	1	Planeta Tierra	10000	5000	2	3	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
*														

PLANET WARS



Here is a more specific example of an insert. When we create the planet, we insert all the necessary data to create it into the database.

```
//se inicia todo en 0
int light_hunter_destroyed = 0;
int heavy_hunter_destroyed = 0;
int battleship_destroyed = 0;
int armored_ship_destroyed = 0;

int missile_launcher_built = 0;
int missile_launcher_destroyed = 0;
int ion_cannon_built = 0;
int ion_cannon_destroyed = 0;
int plasma_canon_built= 0;
int plasma_canon_destroyed = 0;

DatabaseManager.insertPlanetBattleArmy(
    planet.getIdPlanet(),
    battleNumber,
    light_hunter_destroyed,
    heavy_hunter_destroyed,
    battleship_destroyed,
    armored_ship_destroyed
);

DatabaseManager.insertPlanetBattleDefense(
    planet.getIdPlanet(),
    battleNumber,
    missile_launcher_built,
    missile_launcher_destroyed,
    ion_cannon_built,
    ion_cannon_destroyed,
    plasma_canon_built,
    plasma_canon_destroyed
):
```

When the technology defense is upgraded, the database is updated with the new data

```
// Actualizar "Mejoras" de Defensa
public void upgradeTechnologyDefense(){
    int precioAntiguoDefensa = this.getUpgradeAttackTechnologyDeuteriumCost();

    if (this.getDeuterium() > this.getUpgradeDefenseTechnologyDeuteriumCost()) {
        this.setTechnologyDefense(this.getTechnologyDefense() + 1); // Sube 1 punto de defensa

        // Restar material que tienes
        this.setDeuterium(getDeuterium() - this.getUpgradeDefenseTechnologyDeuteriumCost());

        // Actualizar precio de mejora de defensa
        this.setUpgradeDefenseTechnologyDeuteriumCost(
            (int) (this.getUpgradeDefenseTechnologyDeuteriumCost() * 1.6 )); //60%
        System.out.println("PRECIO MEJORA DEFENSA ANTES = " + precioAntiguoDefensa + " AHORA " + this.getUpgradeDefenseTechnologyDeuteriumCost());
        System.out.println("DETERIUM ACTUAL = " + this.getDeuterium());
        DatabaseManager.updatePlanetStats(
            this.getIdPlanet(),
            name:"PlanetName",
            this.getMetal(),
            this.getDeuterium(),
            this.getTechnologyDefense(),
            this.getTechnologyAttack(),
            battles_counter:0,
            this.getArmy()[4].size(),
            this.getArmy()[5].size(),
            this.getArmy()[6].size(),
            this.getArmy()[0].size(),
            this.getArmy()[1].size(),
            this.getArmy()[2].size(),
            this.getArmy()[3].size()
        );
    } else {
        // Si no tienes suficiente material salta el error
        System.out.println("\nNo tienes Deuterium suficiente!\n");
    }
}
```



4. Web outline

○ Folders and contents

In our web we have four main folders:

Nombre	Fecha de modificación	Tipo	Tamaño
📁 About Us	16/05/2025 14:18	Carpetas de archivos	1 KB
📁 Batallas	16/05/2025 15:35	Carpetas de archivos	1 KB
📁 Principal	16/05/2025 14:18	Carpetas de archivos	1 KB
📁 Tutorial	16/05/2025 14:18	Carpetas de archivos	1 KB

Each folder has its html, css, javascript and image if it's necessary.

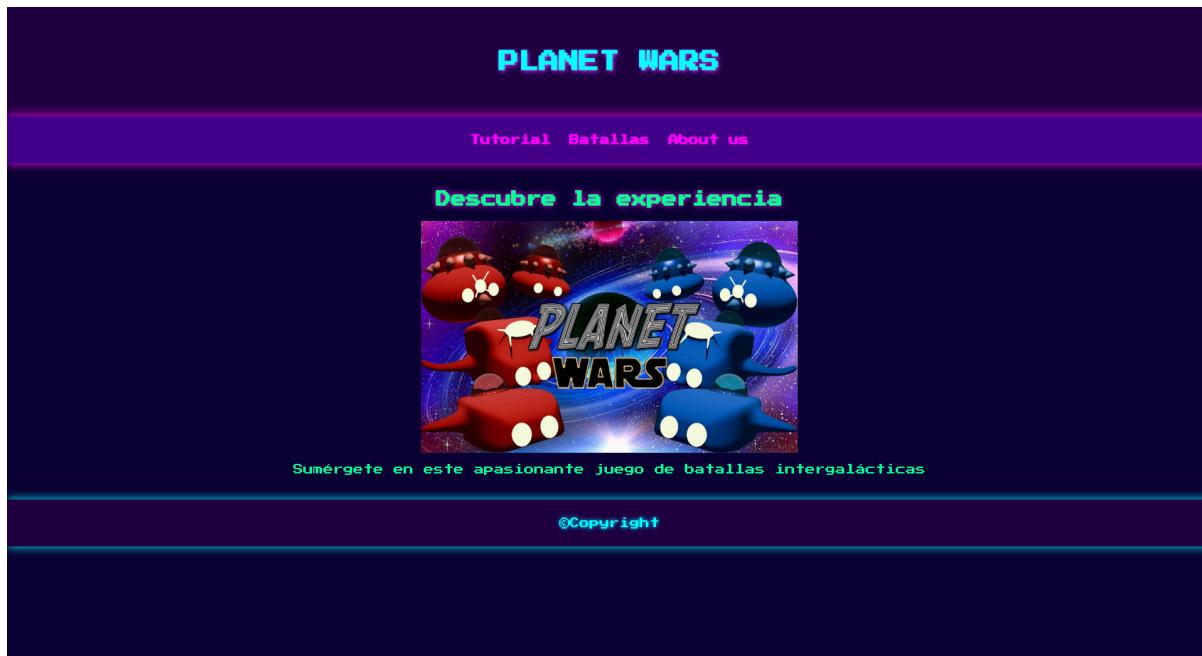
Nombre	Fecha de modificación	Tipo	Tamaño
# batallas.css	16/05/2025 14:18	Archivo de origen ...	3 KB
batallas.html	16/05/2025 15:36	Opera GX Web Do...	2 KB
batallas.js	16/05/2025 16:00	Archivo de origen ...	1 KB
images4.png	16/05/2025 14:18	Archivo PNG	1.317 KB

PLANET WARS



○ Web photos

In the main page we have a big title with an animation. Below that there is a menu with all the different options we have and some pictures of the game and the developers.



In tutorial we have the information needed to start playing the game





In about us we have information about our team and our user images

Sobre Nosotros

[Inicio](#) [Tutorial](#) [Batallas](#)

Los creadores de Space Wars

os retro y la cultura arcade. Nuestra misión es crear los shooters espaciales de los años 80 y 90.



Asier

In battle we can choose a number to see the review of that battle

Reporte de Batallas

[Inicio](#) [Tutorial](#) [About Us](#)

Consulta una batalla

Introduce el número de batalla

Ver Batalla

Detalles de la batalla

Introduce un número de batalla para ver el reporte

PLANET WARS



5. User Manual

To start playing, you need to install the game's [repository](#) from GitHub.

Once downloaded, you'll see that it contains several files. It's very useful to read the **README** file to get all the detailed information about the game.

One of the main files is the game's database, which you'll need to import to your computer.

Once everything is set up, you can run the file and enjoy the game!



PLANET WARS



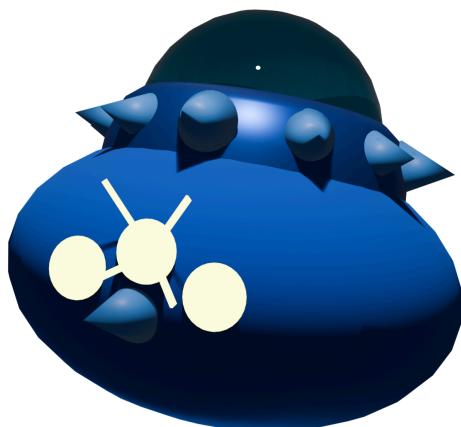
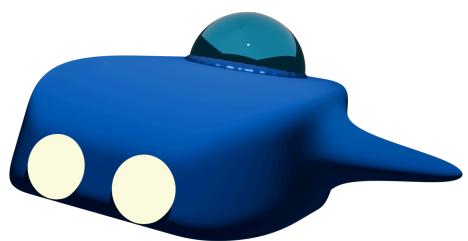
6. Blender

We used the [Blender](#) tool to create the spaceship models for the video game.

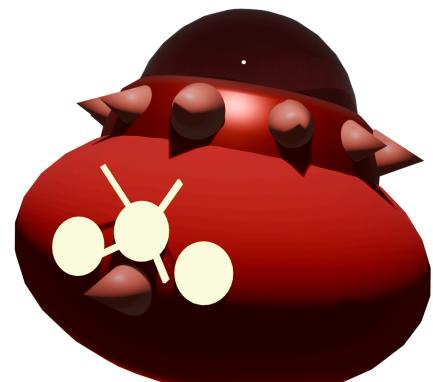
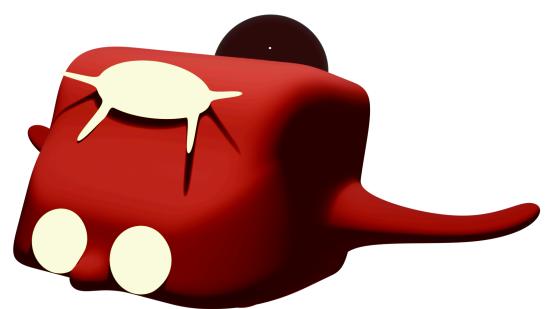
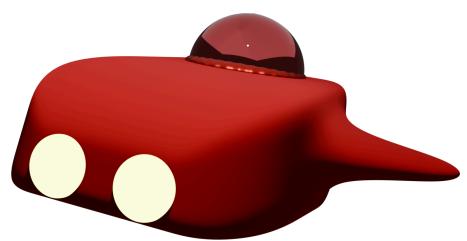
Using its built-in resources (squares, spheres, triangles),

and with the help of some modifiers and materials, we created these spaceships:

[Planet Army]



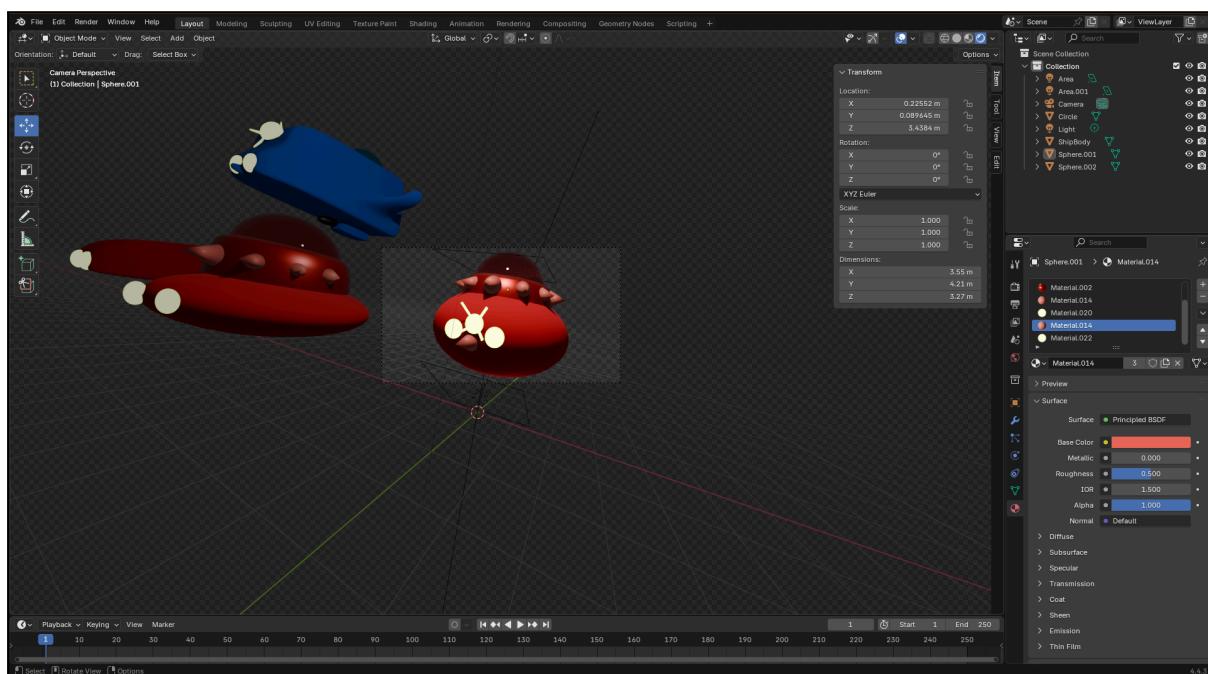
[Enemy Army]



PLANET WARS



Here you can see the Blender creation environment:



PLANET WARS

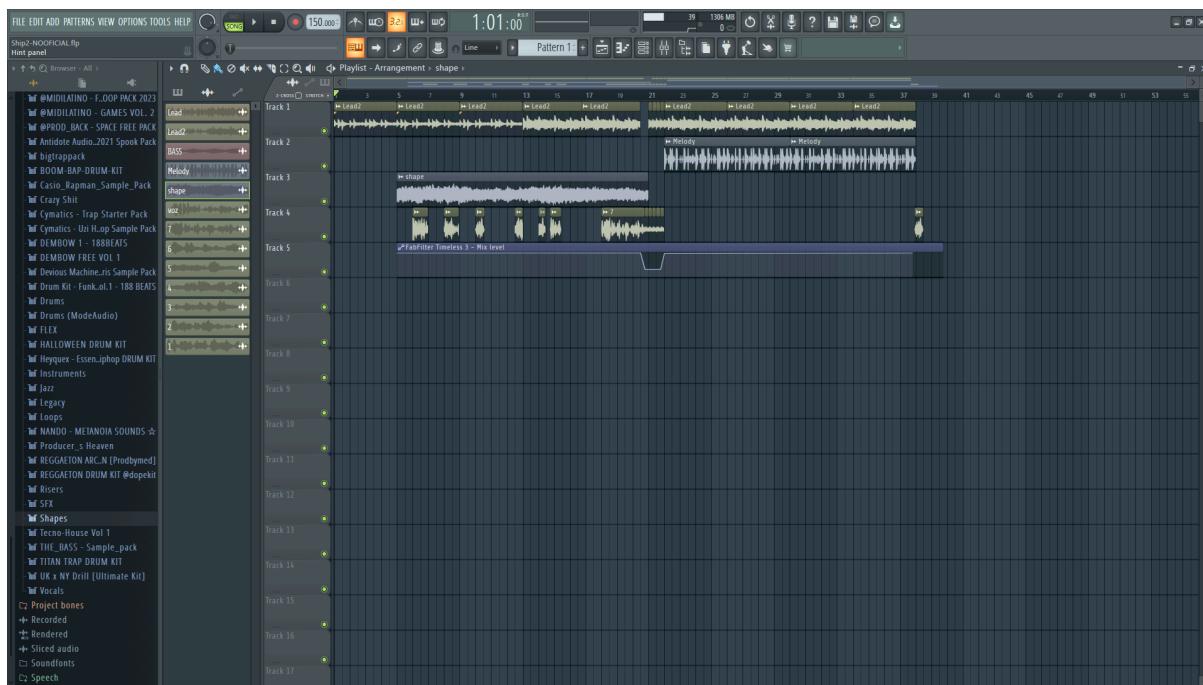


7. Soundtrack

To create the soundtrack for both the video game and the trailer, we used FL Studio 20.

This allowed us to produce music that gives an “alien” and “galactic” vibe.

Here you can see its interface:



You can listen to the instrumental track in our GitHub [repository](#), inside the **M01** folder.



8. XML and HTML into the web

In our program, we have an xsl file that converts some information from xml to html

At first, we needed to create an xsl file to edit the xml:

```

<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:output method="html" encoding="UTF-8"/>
  <xsl:template match="/battleReport">
    <html>
      <head>
        <title>Battle Report <xsl:value-of select="battleNumber"/></title>
      </head>
      <body>
        <h1>Battle Report <xsl:value-of select="battleNumber"/></h1>
        <h2>Statistics</h2>
        <table border="1">
          <tr>
            <th>Unit</th>
            <th>Planet Units</th>
            <th>Planet Drops</th>
            <th>Enemy Units</th>
            <th>Enemy Drops</th>
          </tr>
          <tr>
            <td>Light Hunter</td>
            <td><xsl:value-of select="statistics/planetUnits/lightHunter/@units"/></td>
            <td><xsl:value-of select="statistics/planetUnits/lightHunter/@drops"/></td>
          </tr>
        </table>
      </body>
    </html>
  </xsl:template>
</xsl:stylesheet>

```

Then, we generated the xml file and we transformed it into an html file.

```

// Generar archivo XML
generateBattleReportXML(battleCounter, planetWins, metalGained, deuteriumGained, planetArmyReport.toString(), enemyArmyReport.toString());

// Transformar XML a HTML
String xmlFileName = "battle" + battleCounter + ".xml";
String xslFileName = "battleReport.xsl";
String htmlFileName = "battle" + battleCounter + ".html";
transformXMLtoHTML(xmlFileName, xslFileName, htmlFileName);

private void generateBattleReportXML(int battleNumber, boolean planetWins, int metalGained, int deuteriumGained, String planetArmyReport, String enemyArmyReport) {
String xmlFileName = "battle" + battleNumber + ".xml";

```

PLANET WARS



Here you can see what appears on each file that is shown on the website. There are resources won, armies, generation of the xml...

```
try {
    // Crear el documento XML
    DocumentBuilderFactory docFactory = DocumentBuilderFactory.newInstance();
    DocumentBuilder docBuilder = docFactory.newDocumentBuilder();
    Document doc = docBuilder.newDocument();

    // Elemento raíz
    Element rootElement = doc.createElement(tagName:"battleReport");
    doc.appendChild(rootElement);

    // Número de batalla
    Element battleNumberElement = doc.createElement(tagName:"battleNumber");
    battleNumberElement.appendChild(doc.createTextNode(String.valueOf(battleNumber)));
    rootElement.appendChild(battleNumberElement);

    // Resultado de la batalla
    Element resultElement = doc.createElement(tagName:"result");
    resultElement.appendChild(doc.createTextNode(planetWins ? "win" : "lose"));
    rootElement.appendChild(resultElement);

    // Recursos ganados
    Element resourcesElement = doc.createElement(tagName:"resources");
    Element metalElement = doc.createElement(tagName:"metal");
    metalElement.appendChild(doc.createTextNode(String.valueOf(metalGained)));
    Element deuteriumElement = doc.createElement(tagName:"deuterium");
    deuteriumElement.appendChild(doc.createTextNode(String.valueOf(deuteriumGained)));
    resourcesElement.appendChild(metalElement);
    resourcesElement.appendChild(deuteriumElement);
    rootElement.appendChild(resourcesElement);

    // Ejército del planeta
    Element planetArmyElement = doc.createElement(tagName:"planetArmy");
    Element planetArmyDescription = doc.createElement(tagName:"description");
    planetArmyDescription.appendChild(doc.createTextNode(data:"El ejército del planeta está compuesto por unidades de ataque y defensa."));
    planetArmyElement.appendChild(planetArmyDescription);
```

This piece of code prints the xml file with a pretty format.

```
// Escribir el contenido en el archivo XML con formato bonito
TransformerFactory transformerFactory = TransformerFactory.newInstance();
Transformer transformer = transformerFactory.newTransformer();
transformer.setOutputProperty(OutputKeys.INDENT, value:"yes");
transformer.setOutputProperty(name:"{http://xml.apache.org/xslt}indent-amount", value:"4");
DOMSource source = new DOMSource(doc);
StreamResult result = new StreamResult(new File(xmlFileName));

transformer.transform(source, result);

System.out.println("Archivo XML generado: " + xmlFileName);
catch (Exception e) {
    e.printStackTrace();
```

PLANET WARS



And this other piece of code transforms the XML to HTML using all three xml, xsl and html files.

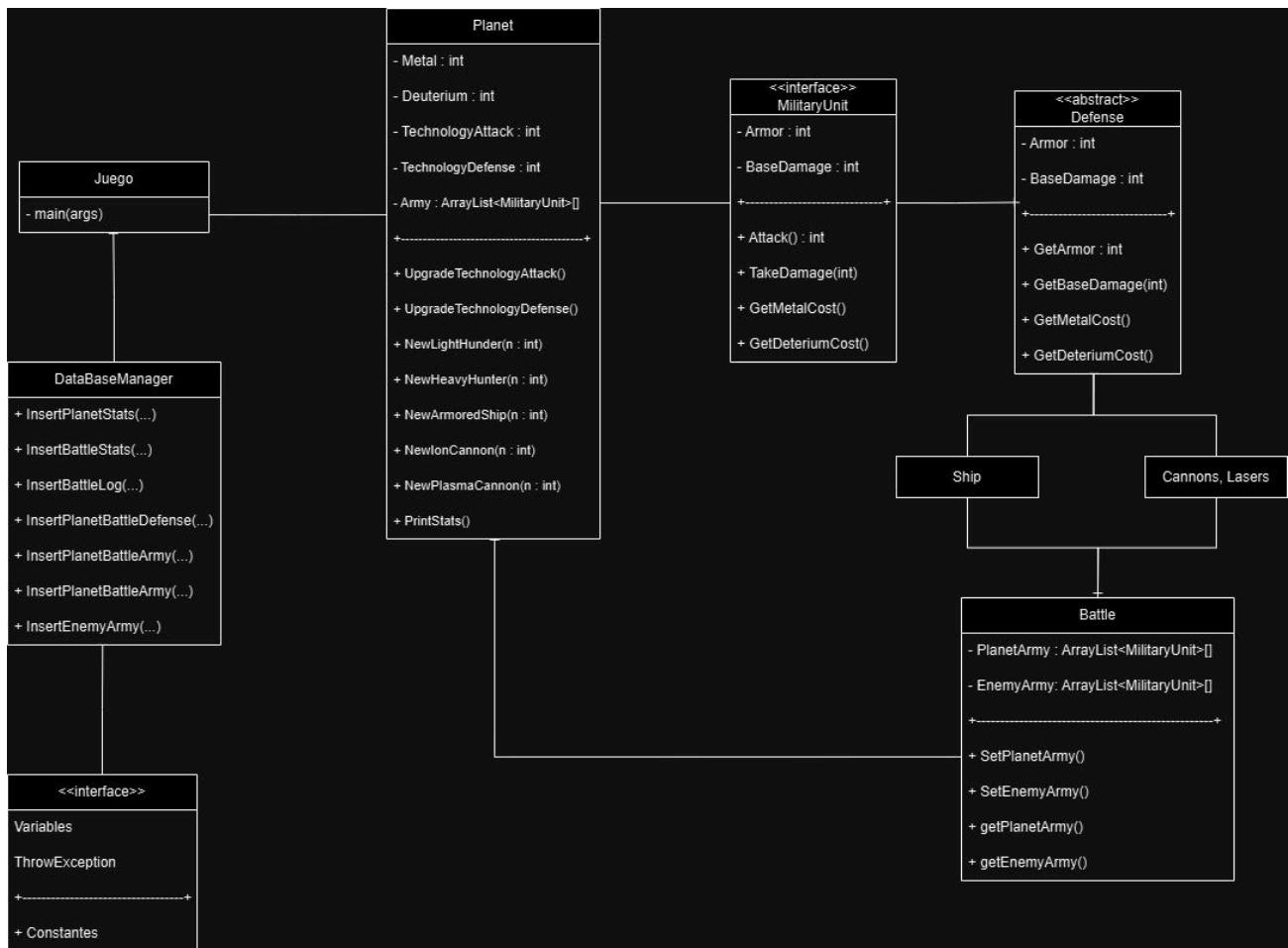
```
private void transformXMLToHTML(String xmlFileName, String xslFileName, String htmlFileName) {  
    try {  
        // Ruta de salida para los archivos HTML  
        String outputPath = "webProjecte/Batallas/" + htmlFileName;  
  
        // Ruta del archivo XSL (ajusta según la ubicación real)  
        String xslPath = "src/" + xslFileName;  
  
        TransformerFactory factory = TransformerFactory.newInstance();  
        Transformer transformer = factory.newTransformer(new StreamSource(xslPath));  
        transformer.transform(new StreamSource(xmlFileName), new StreamResult(outputPath));  
  
        System.out.println("Archivo HTML generado: " + outputPath);  
    } catch (Exception e) {  
        e.printStackTrace();  
    }  
}
```

PLANET WARS



9. Class Diagram

These are the main classes in our project. All of them have their own attributes and methods that let you enjoy the game.



PLANET WARS

