SPACE ENCYCLOPEDIA

Lina Margaryan, Arevik Shahinyan, Ani Baghramyan

INTRODUCTION

The Space Encyclopedia is an interactive Java application with a graphical user interface (GUI) that allows users to explore stars, planets, and galaxies. It combines robust object-oriented design with user-friendly visual navigation to create an engaging experience for learning about the universe.

PROJECT OVERVIEW

- A space-themed encyclopedia with a Graphical User Interface
- Built in Java, using object-oriented design
- Users can browse, search, view, and learn about celestial bodies
- Aimed at students and enthusiasts interested in astronomy
- Designed to be modular, intuitive, and expandable

MAIN FEATURES

BROWSE PANEL

- Browse all celestial objects by category (stars, planets, galaxies)

SEARCH BAR

- Type to find objects by name

DETAILS PANEL

- See deatailed information about objects

DATA STORAGE

- Load/save data from a file automatically

INTERACTIVE GUI

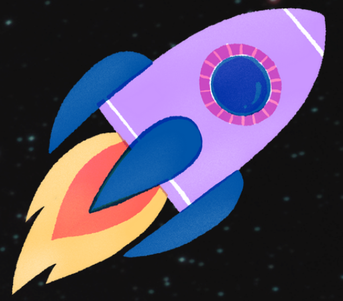
- Engaging space-themed GUI with custom panels




SYSTEM ARCHITECTURE

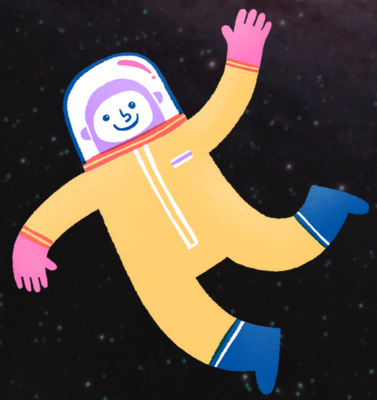
DATA HANDLING

DataManager – reads/writes to file, stores and searches objects



CORE CLASSES

- CelestialBody (abstract) – shared properties (name, mass, distance, etc.)
- Star, Planet, Galaxy – extend CelestialBody with specialized data

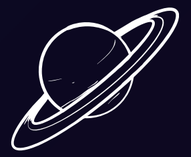


USER INTERFACE

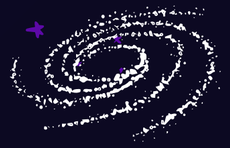
GUISpaceEncyclopedia - Main coordinator and GUI window initializer

```java
public CelestialBody(String name, double distanceFromEarth, double mass, double temperature, String type) {
    if (name == null || name.trim().isEmpty())
        throw new IllegalArgumentException("Name cannot be null or empty.");
    if (distanceFromEarth < 0)
        throw new IllegalArgumentException("Distance from Earth cannot be negative.");
    if (mass <= 0)
        throw new IllegalArgumentException("Mass must be positive.");
    if (temperature < 0)
        throw new IllegalArgumentException("Temperature cannot be negative.");
    if (type == null || type.trim().isEmpty())
        throw new IllegalArgumentException("Type cannot be null or empty.");

    this.name = name;
    this.distanceFromEarth = distanceFromEarth;
    this.mass = mass;
    this.temperature = temperature;
    this.type = type;
}
```

```java
public String calculateLifeStage() {
    double massInKg = getMass();  // mass in kg
    double massInSolarMasses = massInKg / SUN_MASS_KG;

    if (massInSolarMasses < 0.08)
        return "Brown Dwarf (Failed Star)";
    else if (massInSolarMasses < 0.5)
        return "Low-Mass Main Sequence Star";
    else if (massInSolarMasses <= 8)
        return "Main Sequence Star";
    else
        return "High-Mass Main Sequence Star";
}
```



```java
public String moonCategory() {
  if (numberOfMoons == 0)
    return "Moonless";
  else if (numberOfMoons <= 2)
    return "Few Moons";
  else if (numberOfMoons <= 10)
    return "Multi-Mooned";
  else
    return "Satellite Swarm";
}
```



```java
public double estimateStarDensity() {
    double radiusLY = diameter / 2.0; // diameter in light-years
    double volume = (4.0 / 3.0) * Math.PI * Math.pow(radiusLY, 3); // in cubic light-years
    if (volume <= 0)
        throw new ArithmeticException("Galaxy volume must be greater than zero to estimate star density.");
    return numberOfStars / volume;
}
/*
```

```
    private void loadData() {
        if (!new File(DATA_FILE).exists()) {
            System.out.println("Creating and loading sample celestial objects...");
            createSampleData();
            saveData();
            return;
        }

        try (BufferedReader reader = new BufferedReader(new FileReader(DATA_FILE))) {
            String line;
            while ((line = reader.readLine()) != null) {
                CelestialBody body = parseLine(line);
                if (body != null) {
                    if (body instanceof Star) {
                        addStar((Star) body);
                    } else if (body instanceof Planet) {
                        addPlanet((Planet) body);
                    } else if (body instanceof Galaxy) {
                        addGalaxy((Galaxy) body);
                    }
                }
            }
        } catch (IOException e) {
        System.out.println("Error loading data from " + DATA_FILE + ": " + e.getMessage());
    }
    }
```

GUI CLASSES

BUTTONS

- MenuButtons - GUI buttons for navigation (Search, Browse, Exit, etc.)

MENU PANEL

MenuPanel - The main entry screen of the application

PANELS

- SearchPanel - UI panel for searching celestial bodies
- BrowsePanel - UI for listing all celestial bodies
- DetailsPanel - Displays object information and facts


THANK
YOU