Stellaris Sandbox: Portfolio Assignment

Matthew S. Macovsky

Logan P. Traffas

CS 340-400 Winter 2021

Oregon State University

Link to HTML Interface: `http://flip3.engr.oregonstate.edu:3845/`

Contents

1 Executive Summary of Project Changes

- Change colors to use RGB hex color codes rather than a choice between a small number of colors.
- Replaced the Systems int starCount attribute with a varchar(16) type attribute that can be one of {“unary”, “binary”, “trinary”}. Because of the few options available, it makes more sense and is more immersive to use these names rather than a number.
- Changed color attributes to be varchar(7) instead of varchar(6) so that they could include the # symbol in the hex code color (ex: #FFFFFF).
- Change minimum system orbitalRadius to 0.25 in the outline.
- Add a varchar(16) planetType attribute to bodies to further characterize what kind of planet the body is if its type is ”planet”. This is set at random upon creation and is used only to stylize bodies when they are displayed in JavaScript canvases.
- Add varchar(24) star1Type, star2Type, and star3Type attributes to systems to further characterize what kind of stars it has. For unary systems, star2Type and star3Type default to NULL, and for binary systems, star3Type is NULL. These are set at random upon creation and are used only to stylize systems when they are displayed in JavaScript canvases.

2 Project Outline

In the videogame Stellaris, players control galactic empires competing for control of a galaxy and its resources. Stellaris Sandbox is a database-driven website that will allow users to design a galaxy so that a game of Stellaris can be played in it. This will provide users with the freedom to create new starting circumstances for players, lending itself especially to role-playing oriented players. Stellaris has an average of 14,000 concurrent players on PC alone and a number of active online communities, together making up a large audience with a potential interest in the tool. A custom mod would be needed in order to import the galaxies designed in Stellaris Sandbox into the game itself.

Using Stellaris Sandbox, users can create between 200 and 1000 Systems within the galaxy as well as the hyperlanes that connect them. Users can also create between 0 and 10 Bodies, such as planets, within Systems that can each contain 0 to 3 exploitable Resource deposits. AI-controlled Empires can also be created that span Systems and start off with given quantities of each Resource.

In Stellaris, there is only one galaxy, so Stellaris Sandbox will only allow design of a single galaxy at a time. Since users will be expected to create galaxies with a large number of Systems, Stellaris Sandbox will provide templates and other tools to automatically generate features which can then be edited by the user.

3 Database Outline

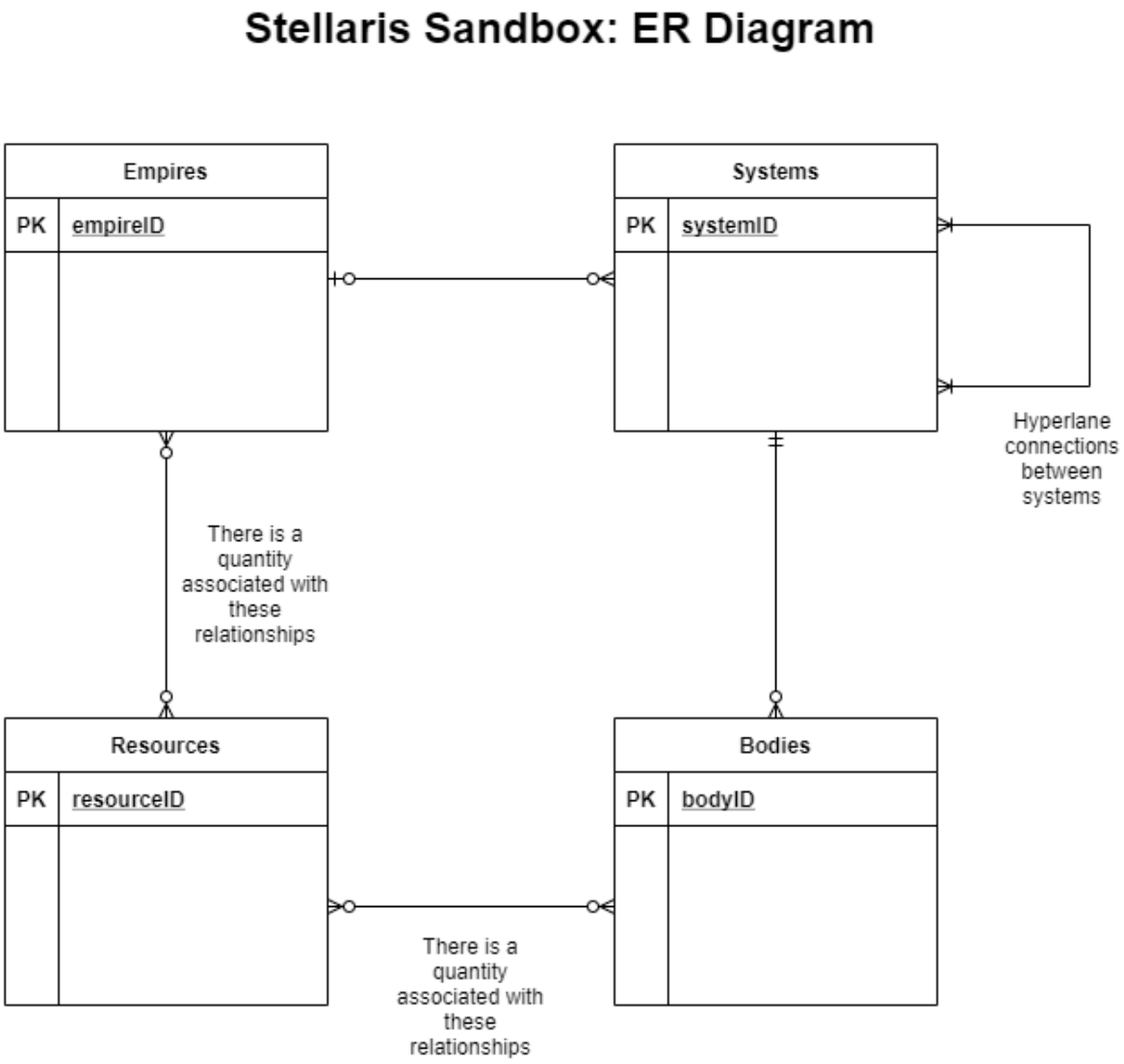
- Empires: Pre-existing, AI-controlled galactic empires
  - empireID: int, auto_increment, unique, not NULL, PK
  - name: varchar(255), not NULL
  - aggressiveness: varchar(16), not NULL, one of {“passive”, “moderate”, “aggressive”}
  - primaryColor: varchar(7), not NULL, the color Hex code
  - secondaryColor: varchar(7), not NULL, the color Hex code
  - isFallenEmpire: bool, not NULL
  - Relationship: 1:M between Empires and Systems, implemented with empireID as a FK within Systems. This relationship represents which empire owns the system.
  - Relationship: M:M between Empires and Resources, implemented in a separate table with empireID and resourceID as FKs and an int resourceQuantity. This represents resource stockpiles owned by an empire. Owner: Logan Traffas
  - Owner: Matthew Macovsky
- Systems: Star systems connected by hyperlanes and controlled by empires
  - systemID: int, auto_increment, unique, not NULL, PK
  - name: varchar(255), not NULL
  - type: varchar(16), not NULL, one of {“unary”, “binary”, “trinary”}
  - star1Type: varchar(24), default NULL, one of {”class b”, ”class a”, ”class f”, ”class g”, ”class k”, ”class m”, ”class m red giant”, ”class t brown dwarf”}
  - star2Type: varchar(24), default NULL, one of {”class b”, ”class a”, ”class f”, ”class g”, ”class k”, ”class m”, ”class m red giant”, ”class t brown dwarf”}
  - star3Type: varchar(24), default NULL, one of {”class b”, ”class a”, ”class f”, ”class g”, ”class k”, ”class m”, ”class m red giant”, ”class t brown dwarf”}
  - orbitalRadius: float, not NULL, between 0.25 and 1.0
  - theta: float, not NULL, between 0 and 360. The theta and orbitalRadius attributes together indicate the positions of Systems in the galaxy using polar coordinates.
  - empireID: int, FK
  - Relationship: M:M between Systems, implemented in a separate table with system1 and system2 as FKs. This relationship consists of the hyperlane connections between systems. Matthew Macovsky and Logan Traffas will work together on this relationship in the database.
  - Relationship: 1:M between Systems and Bodies, implemented with systemID as a FK within Bodies. Owner: Logan Traffas

    - Indirect Relationship: M:M between Systems and Bodies, since Systems contain Bodies which are in turn related to Resources. Does not require explicit implementation in the database.
    - Owner: Logan Traffas

- Bodies: Astronomical bodies that reside in systems (namely planets and asteroids)
    - bodyID: int, auto_increment, unique, not NULL, PK
    - name: varchar(255), not NULL
    - type: varchar(16), not NULL, one of {"planet", "asteroid"}
    - planetType: varchar(16), default NULL, one of {"arid", "desert", "savannah", "alpine", "arctic", "tundra", "continental", "ocean", "tropical"}
    - orbitalRadius: float, not NULL, between 0.1 and 1.0
    - theta: float, not NULL, between 0 and 360. The theta and orbitalRadius attributes together indicate the positions of Bodies in Systems using polar coordinates.
    - systemID: int, not NULL, FK
    - Relationship: M:M between Bodies and Resources, implemented in a separate table with bodyID and resourceID as FKs and an int resourceQuantity. This represents resource deposits on a body that can produce that many of that resource per month. Owner: Matthew Macovsky
    - Owner: Logan Traffas

- Resources: Resource deposits on bodies that can be exploited by empires
    - resourceID: int, auto_increment, unique, not NULL, PK
    - name: varchar(255), not NULL
    - baseMarketValue: float, greater than 0
    - color: varchar(7), not NULL, the color Hex code
    - Owner: Matthew Macovsky

4 Entity-Relationship Diagram



5 Schema

```
Empires(
        empireID,
        name,
        aggressiveness,
        primaryColor,
        secondaryColor,
        isFallenEmpire)

Systems(
        systemID,
        name,
        type,
        star1Type,
        star2Type,
        star3Type,
        orbitalRadius,
        theta,
        empireID)

Bodies(
        bodyID,
        name,
        type,
        planetType,
        orbitalRadius,
        theta,
        systemID)

Resources(
        resourceID,
        name,
        baseMarketValue,
        color)

Hyperlanes(
        system1ID,
        system2ID)

ResourceStock(
        empireID,
        resourceID,
        quantity)

ResourceDeposits(
        bodyID,
        resourceID,
        quantity)
```

6 Screen Captures

6.1 Home Page

Figure 1: Home page, hub slide.

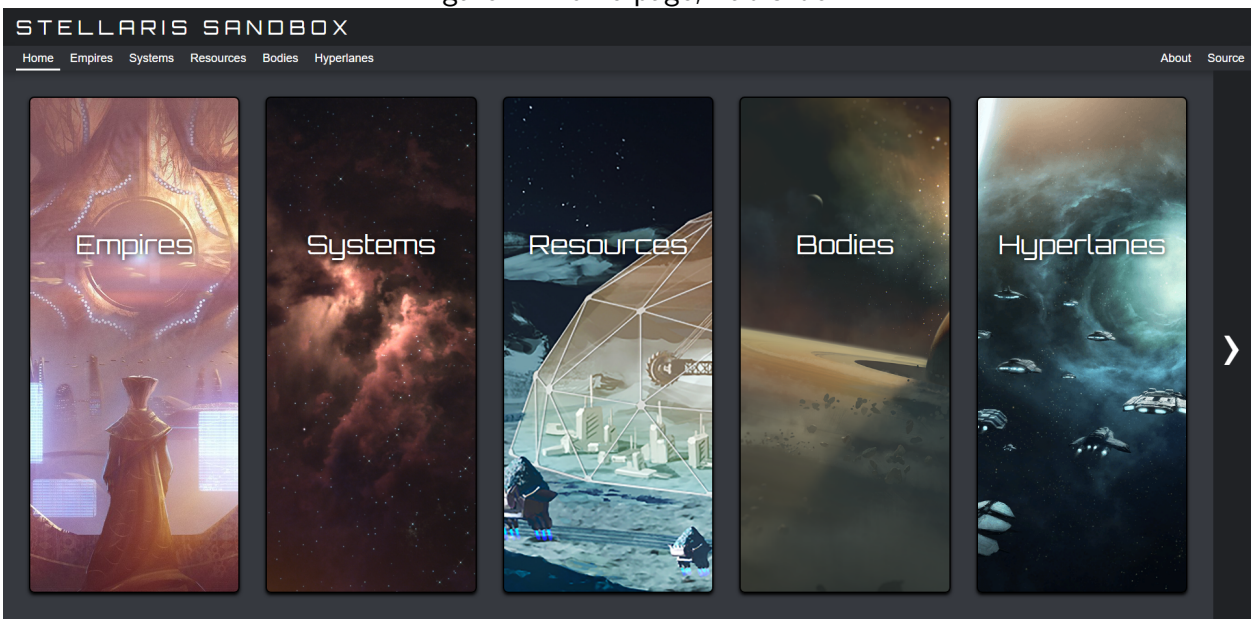
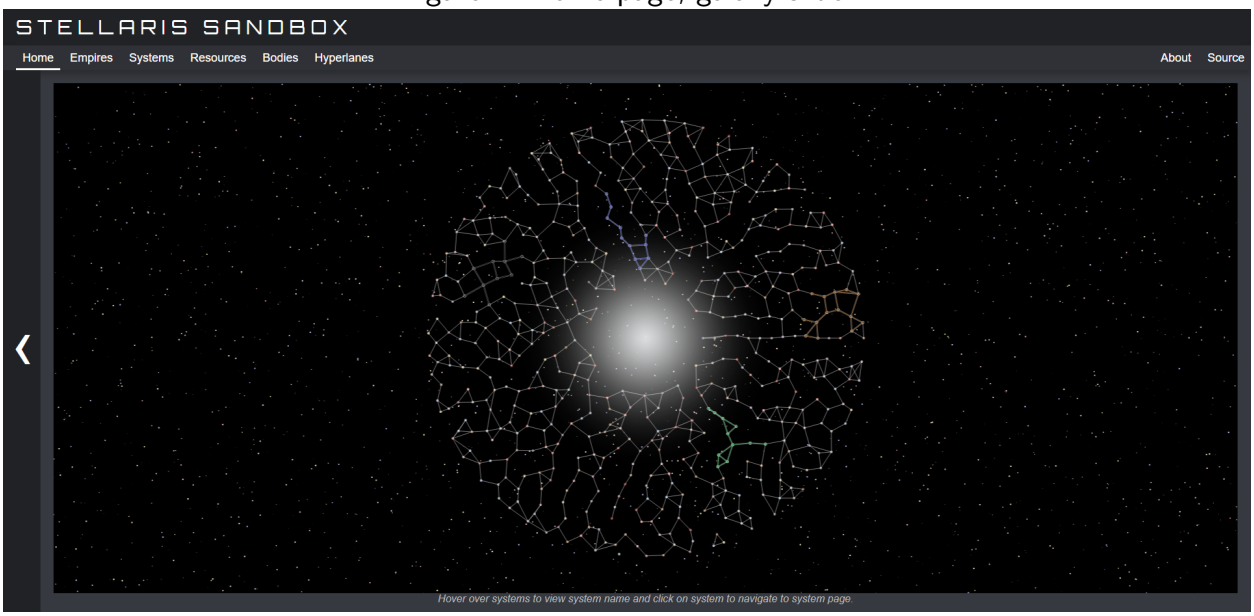


Figure 2: Home page, galaxy slide.



6.2 Empires

Figure 3: BROWSE Empires page.

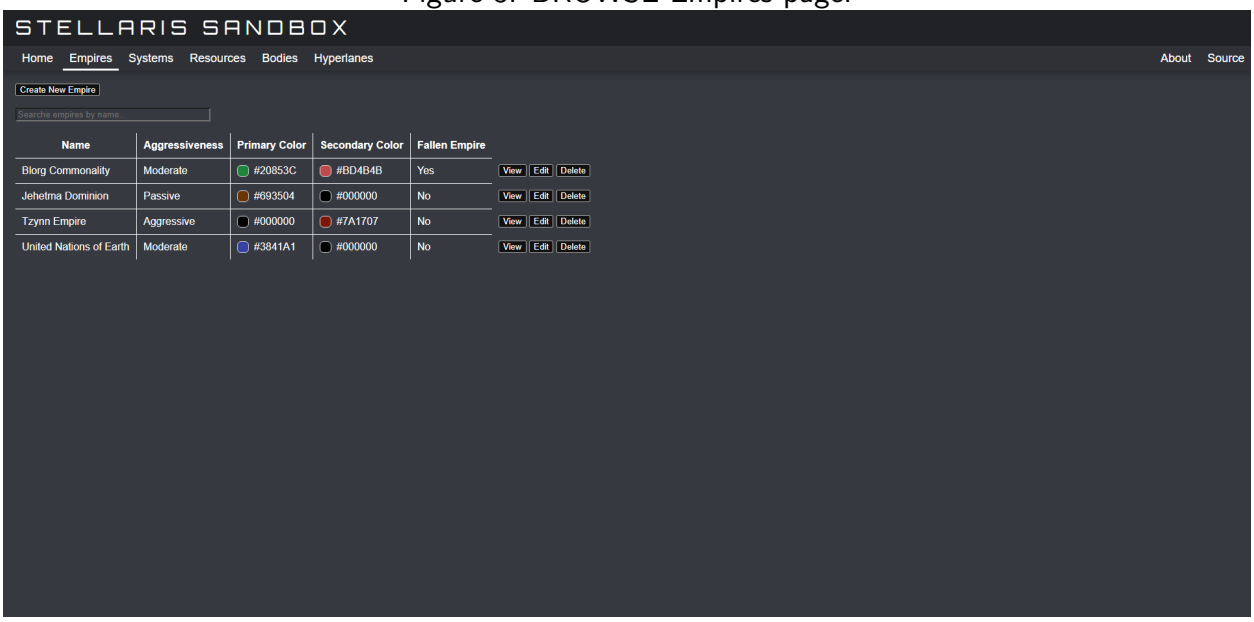


Figure 4: DELETE Empire page.

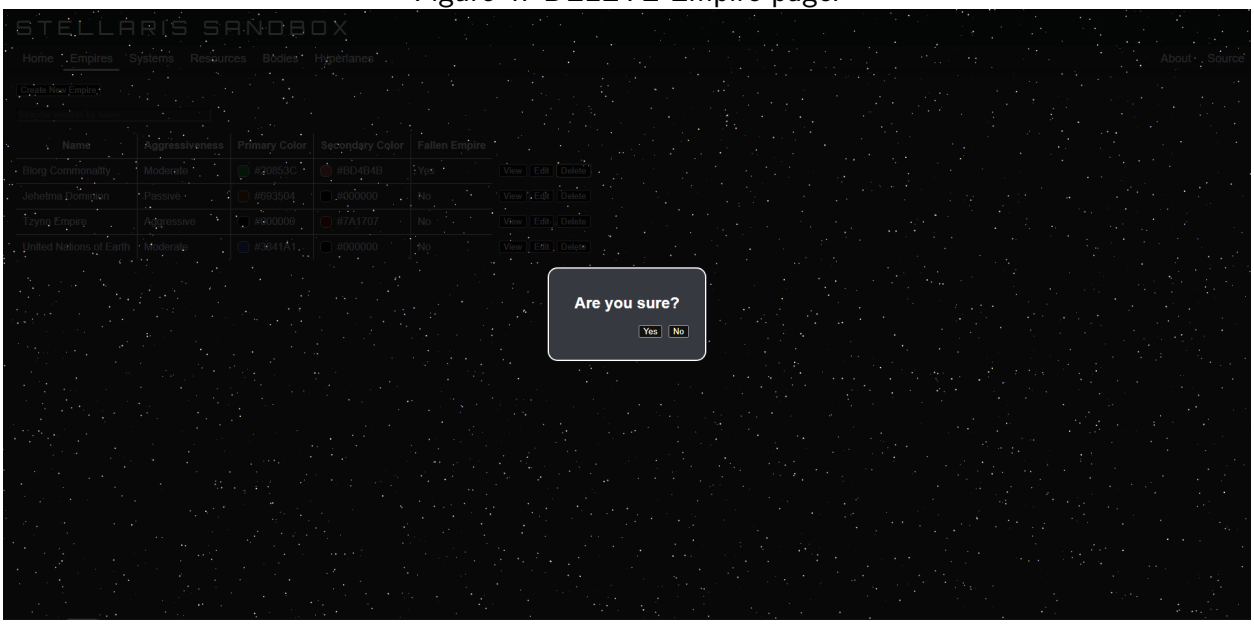


Figure 5: CREATE Empire page.

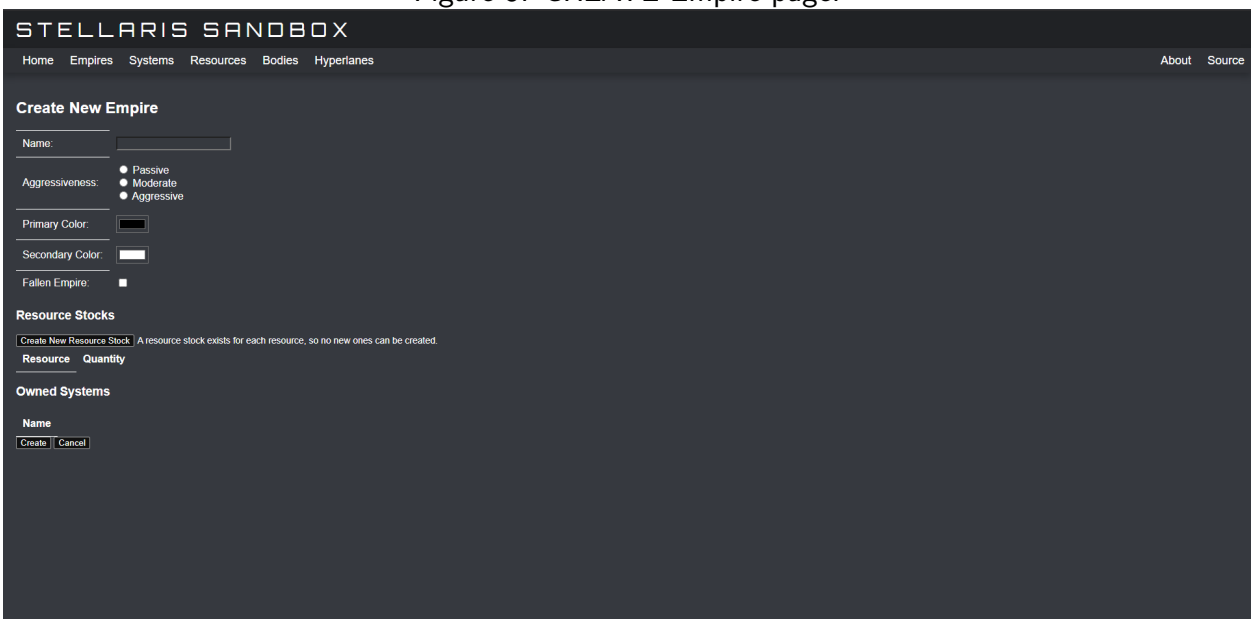


Figure 6: READ Empire page.

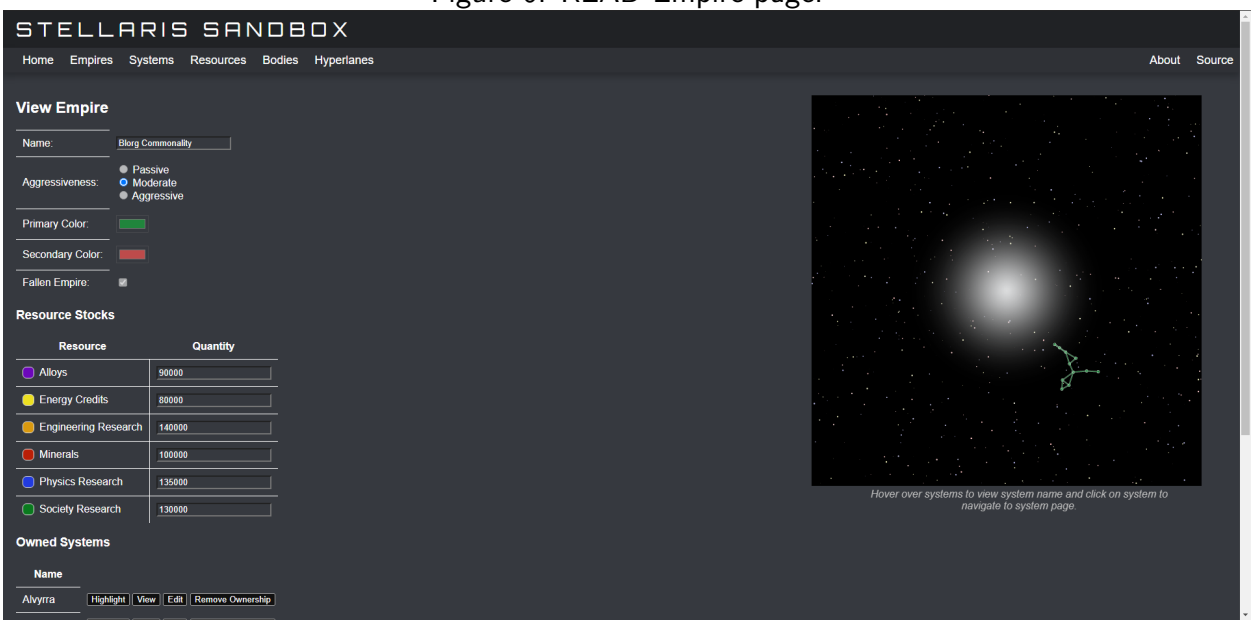
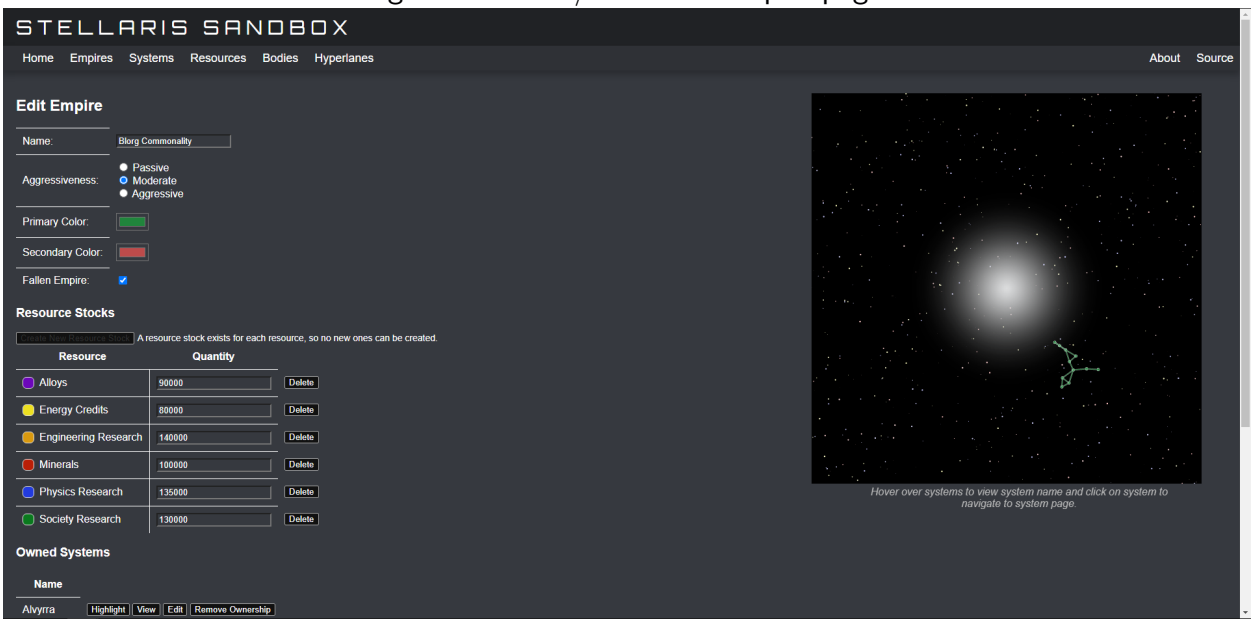


Figure 7: READ/UPDATE Empire page.



6.3 Systems

Figure 8: BROWSE Systems page.

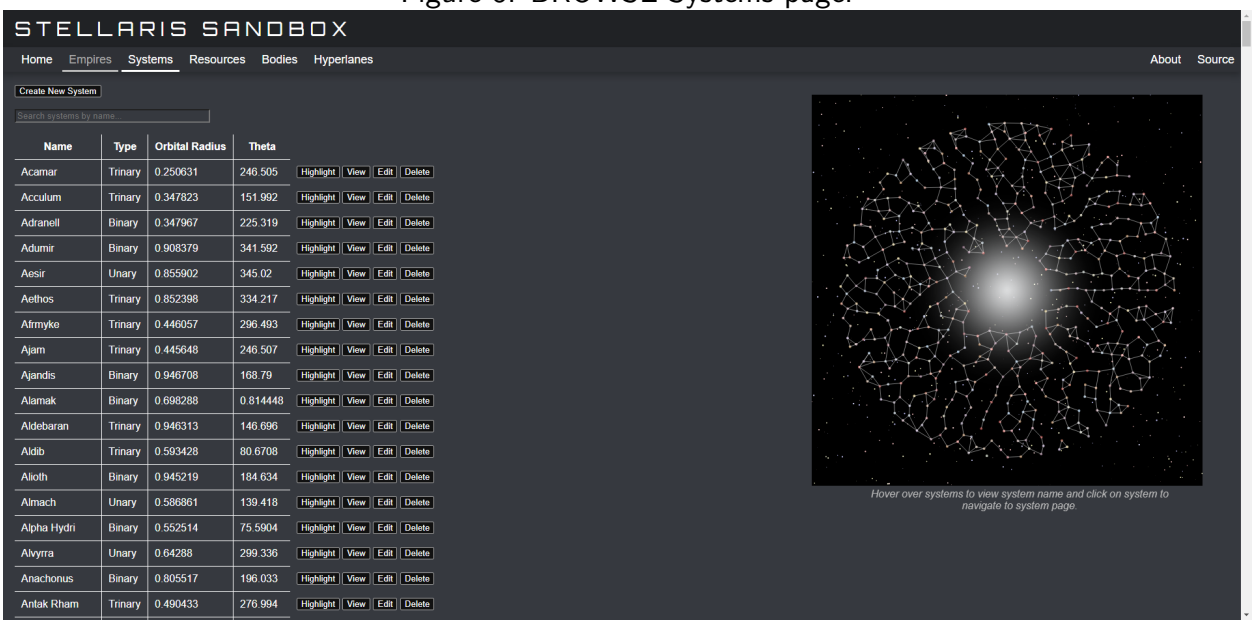


Figure 9: DELETE System page.

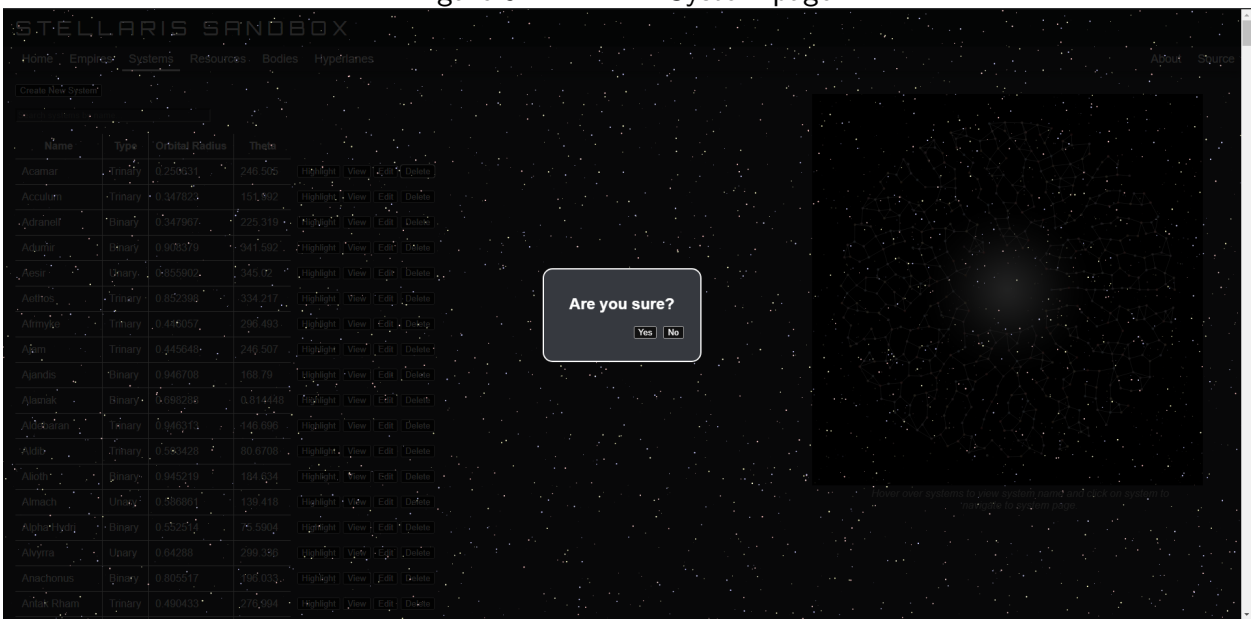


Figure 10: CREATE System page.

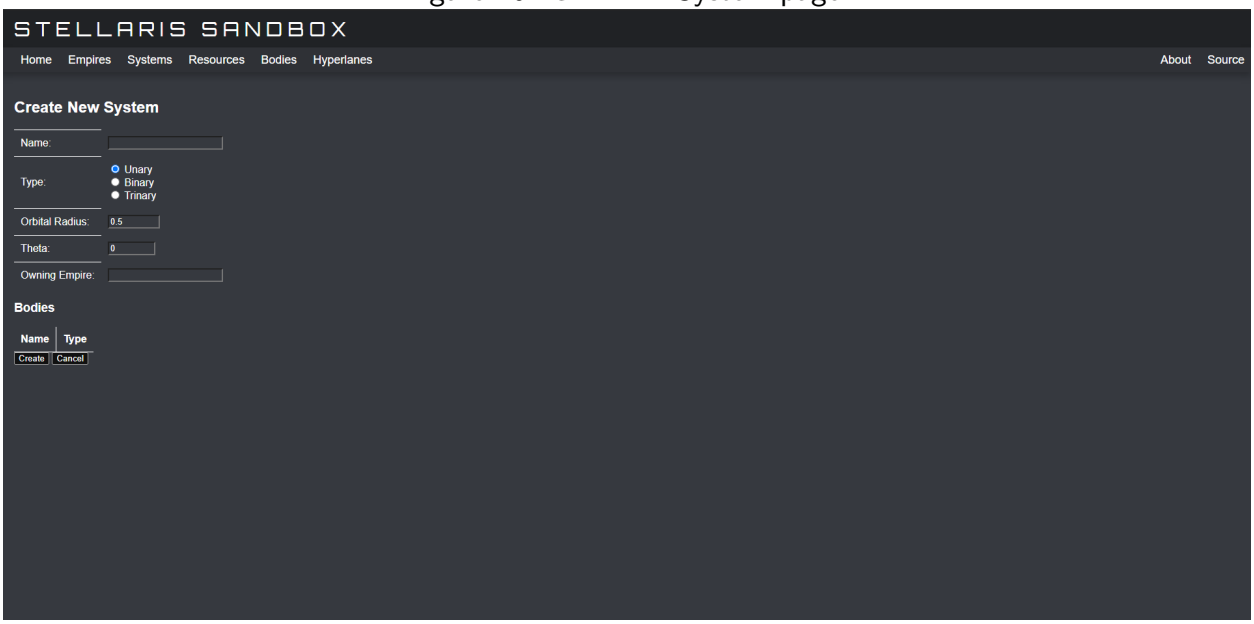


Figure 11: READ System page.

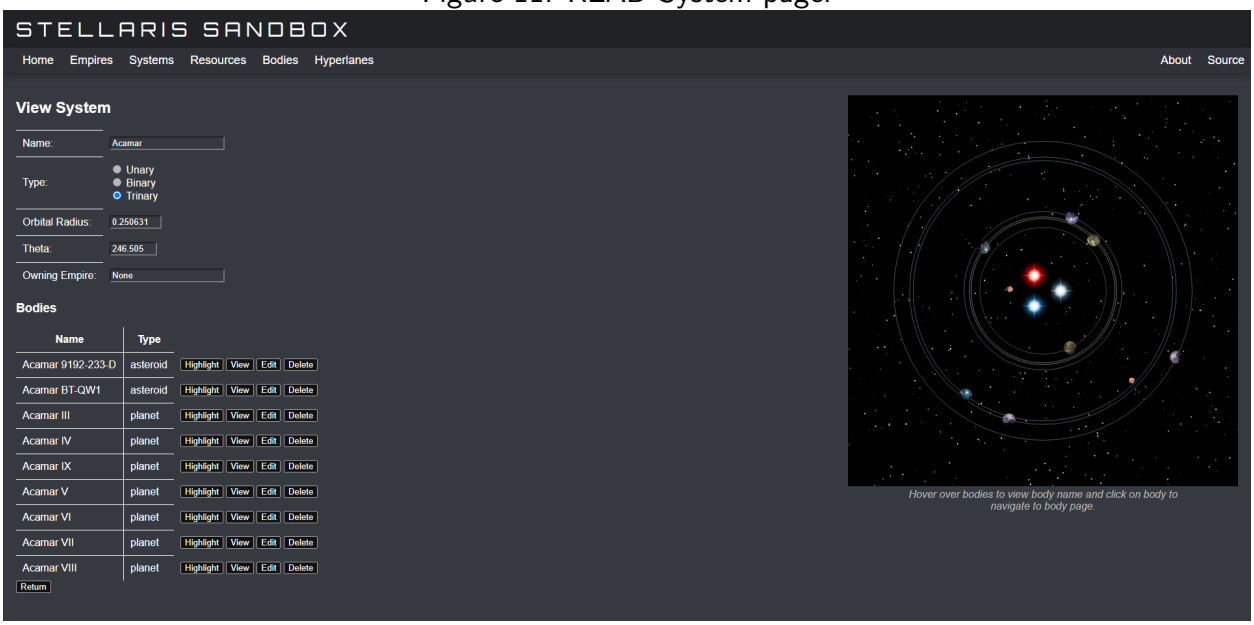
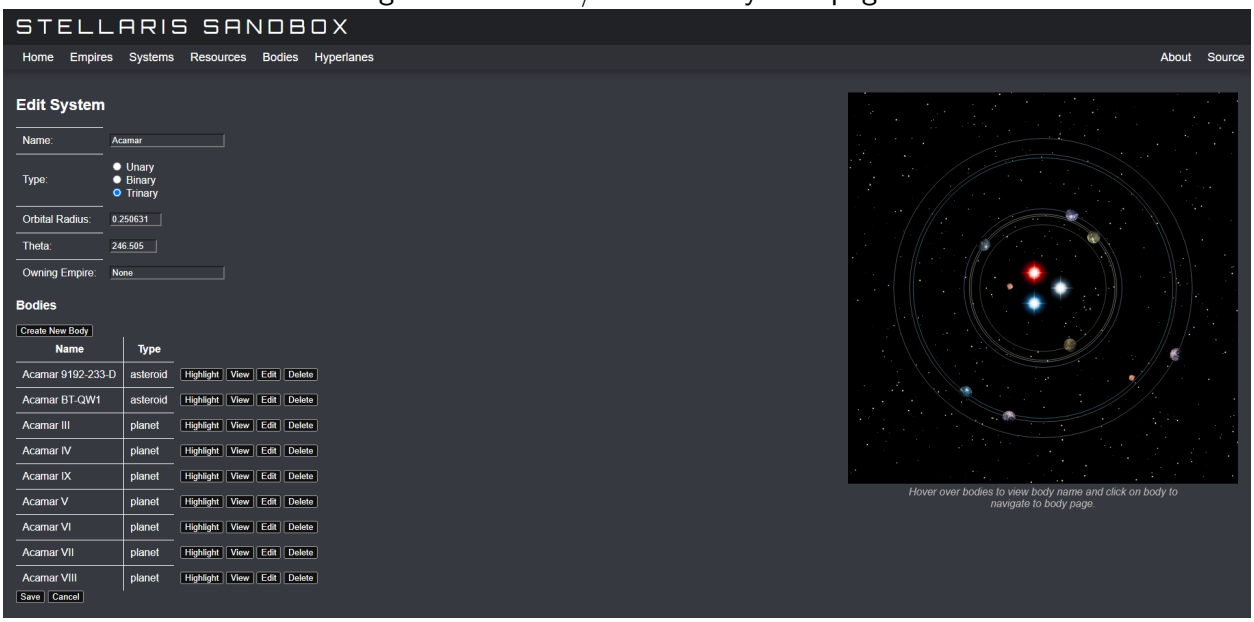


Figure 12: READ/UPDATE System page.



6.4 Bodies

Figure 13: BROWSE Bodies page.

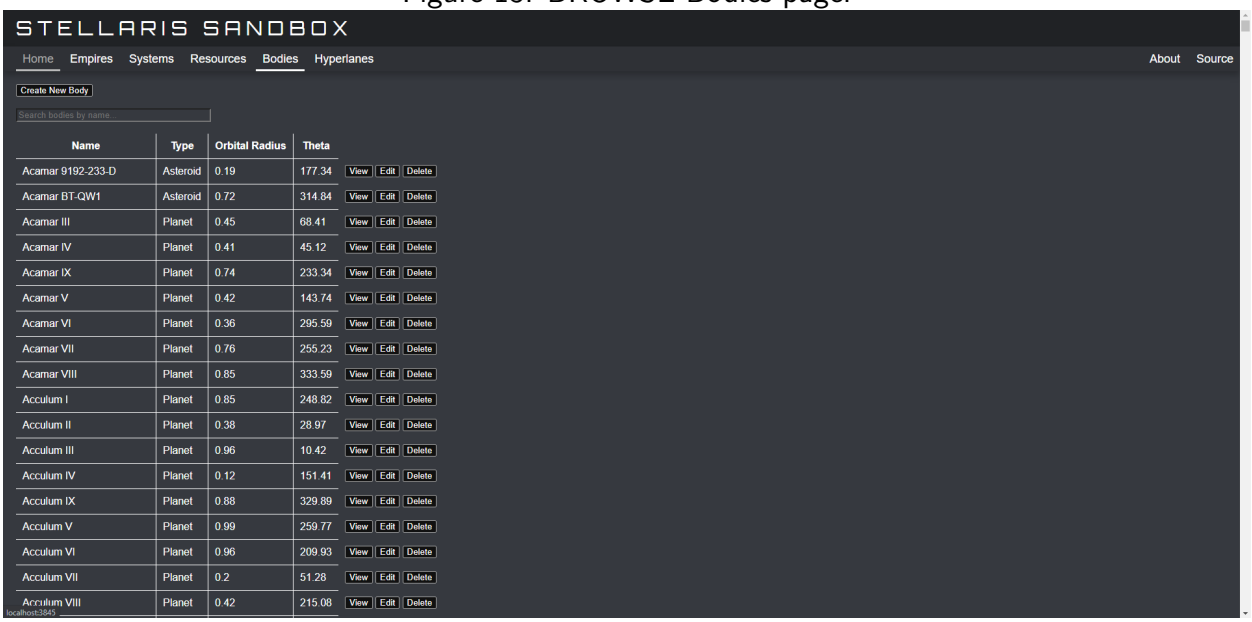


Figure 14: DELETE Body page.

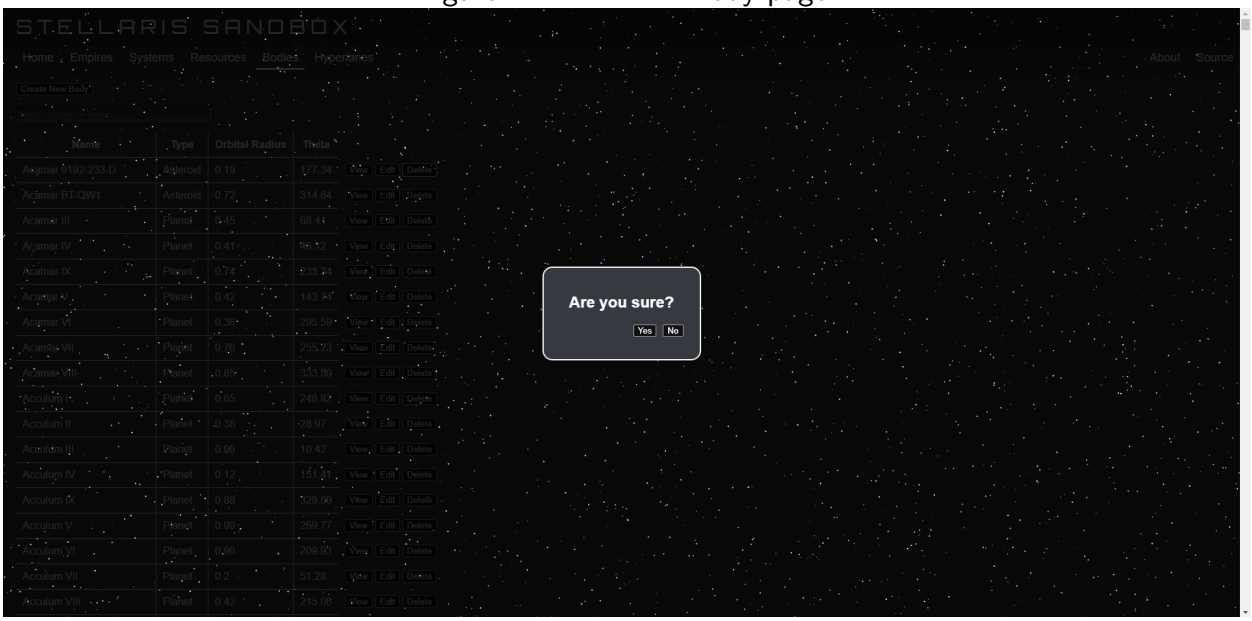


Figure 15: CREATE Body page.

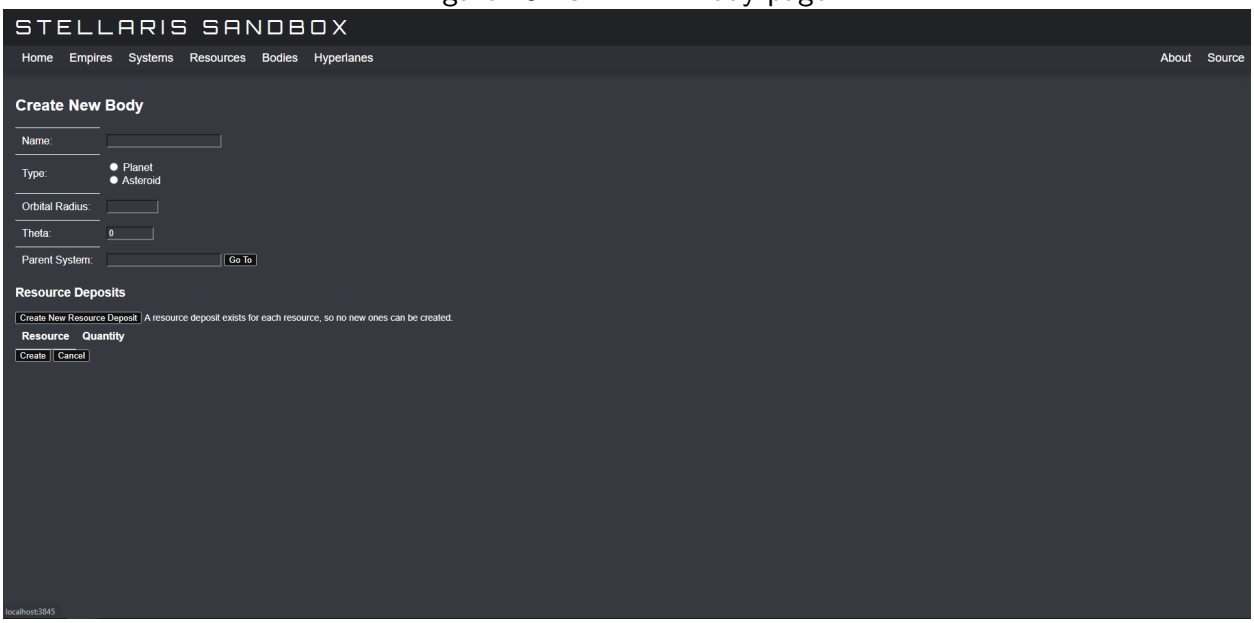


Figure 16: READ Body page.

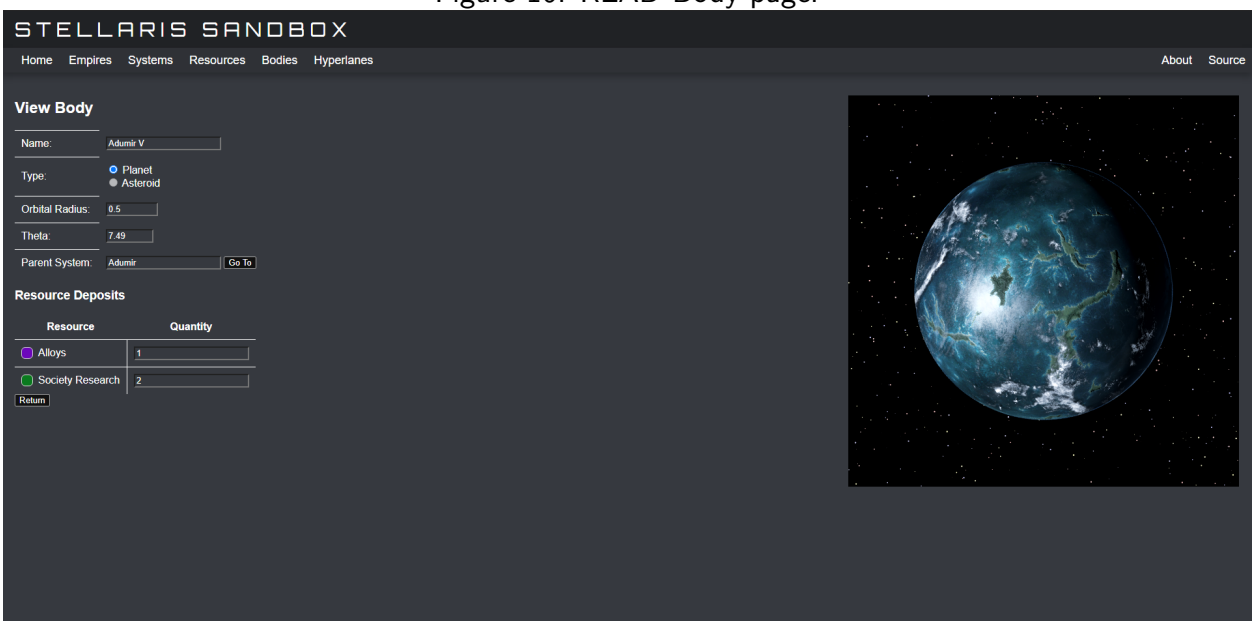
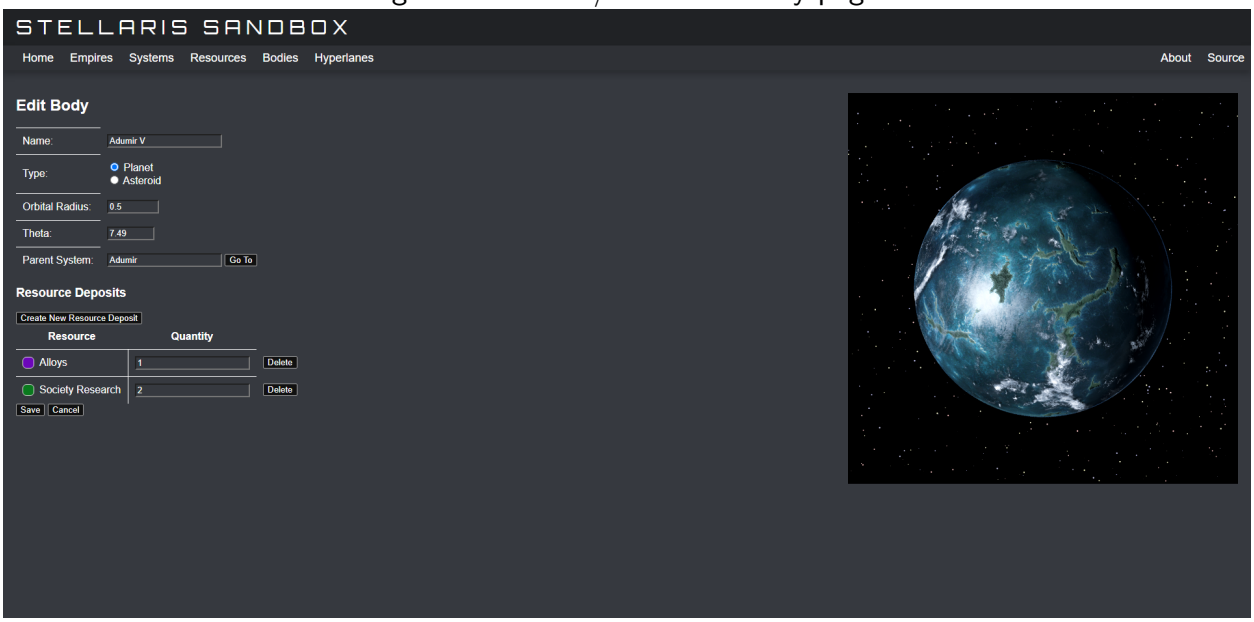


Figure 17: READ/UPDATE Body page.



6.5 Resources

Figure 18: BROWSE Resources page.

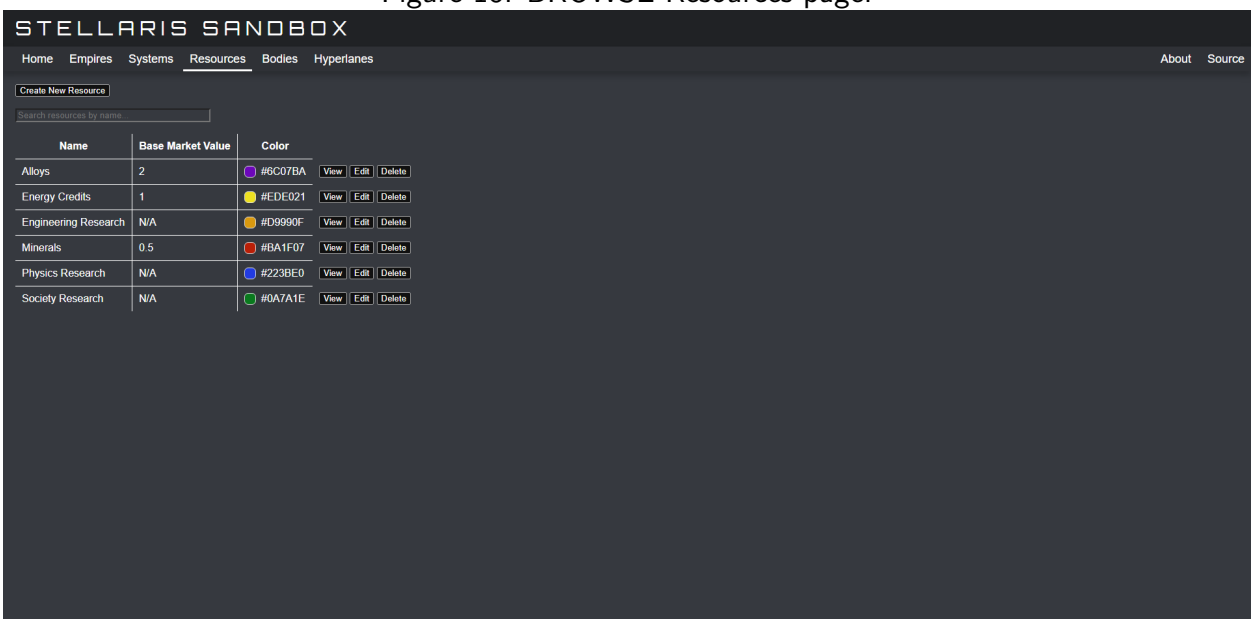


Figure 19: DELETE Resource page.

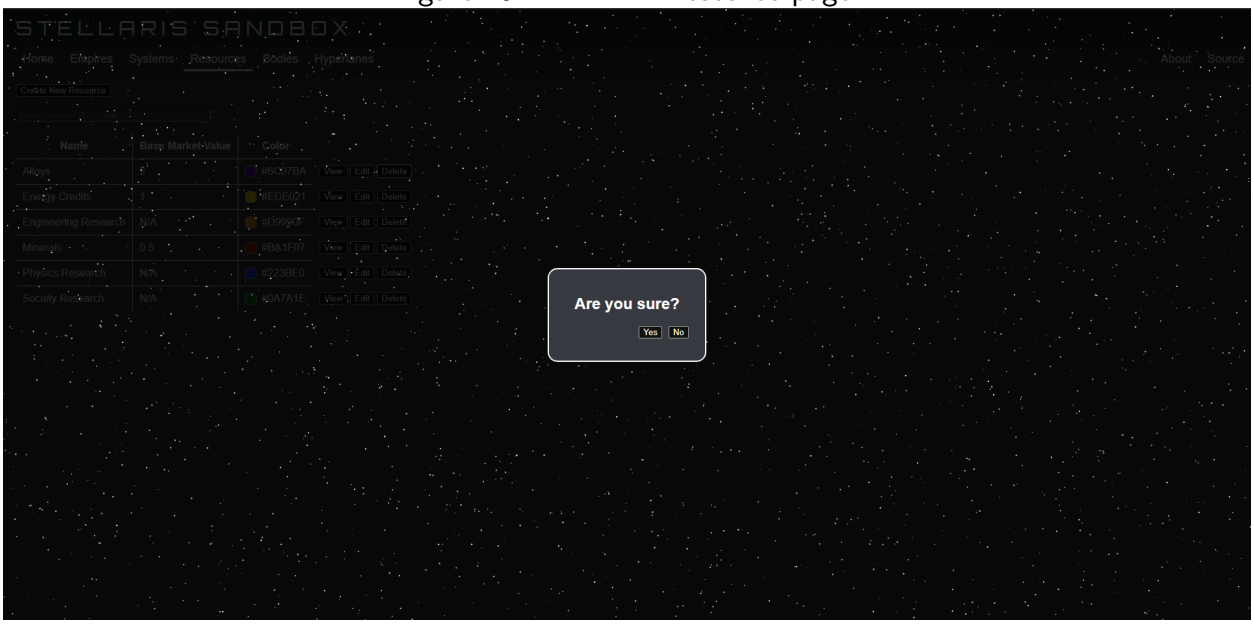


Figure 20: CREATE Resource page.

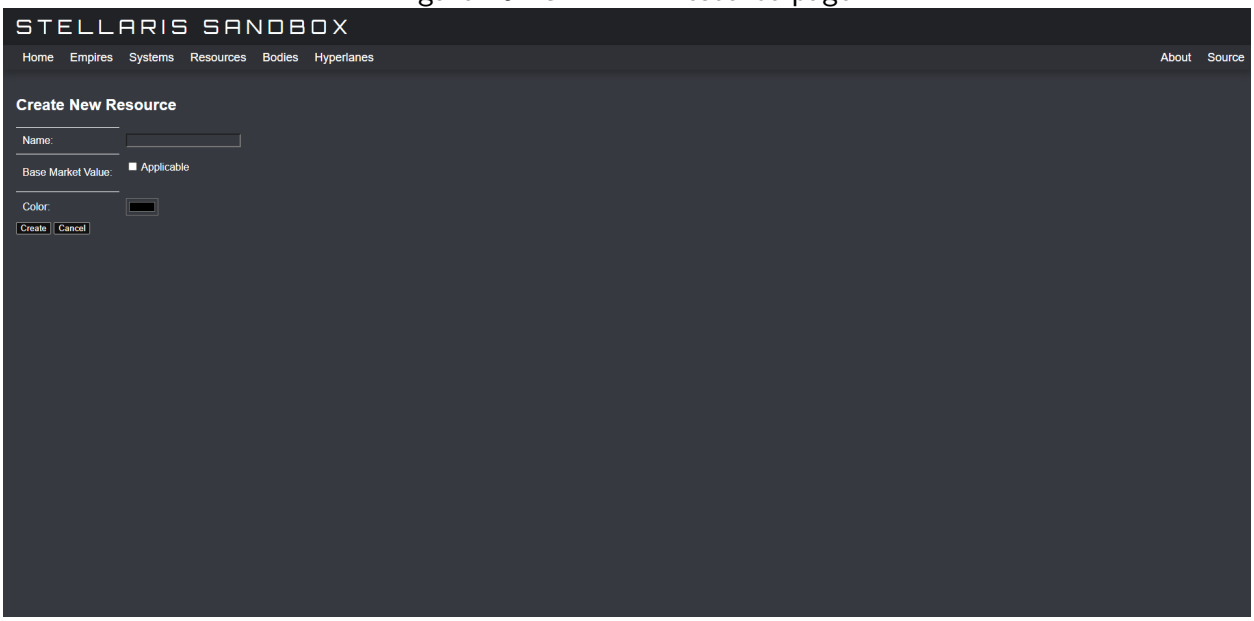


Figure 21: READ Resource page.

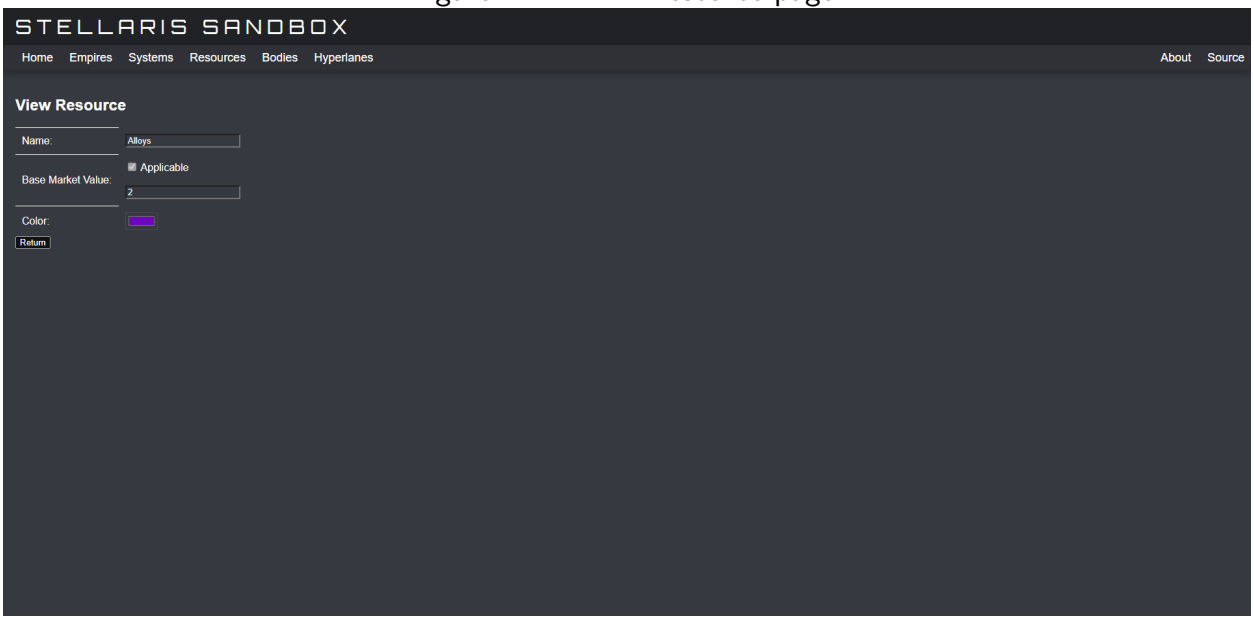
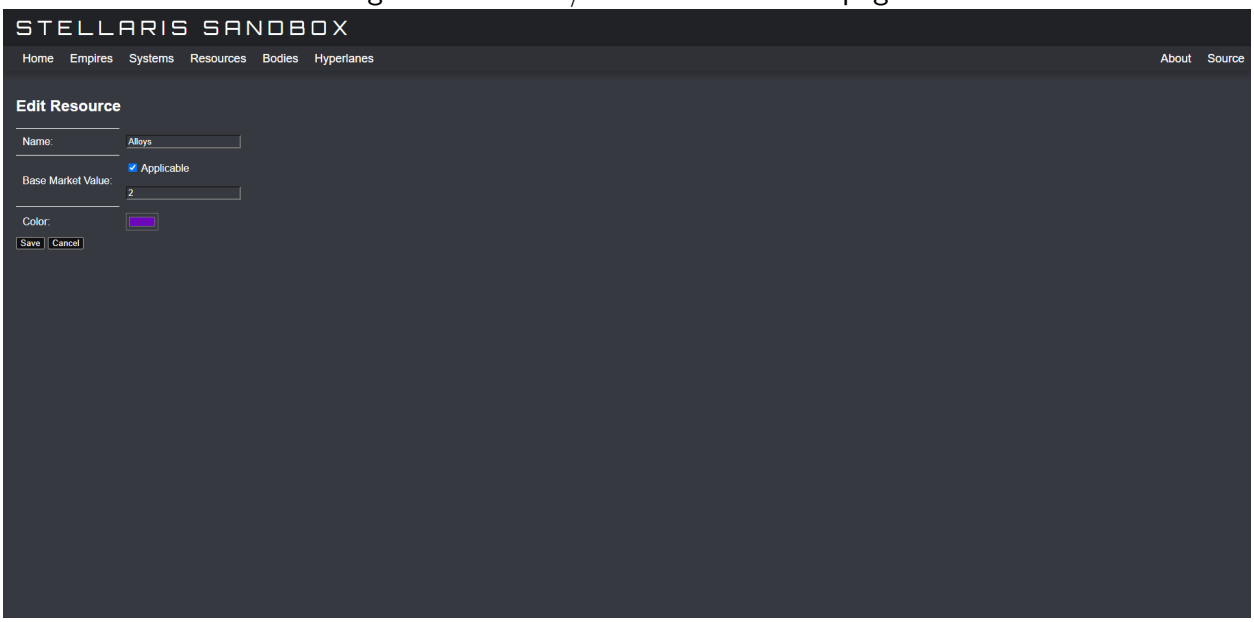


Figure 22: READ/UPDATE Resource page.



6.6 Hyperlanes

Figure 23: BROWSE/READ Hyperlanes page.

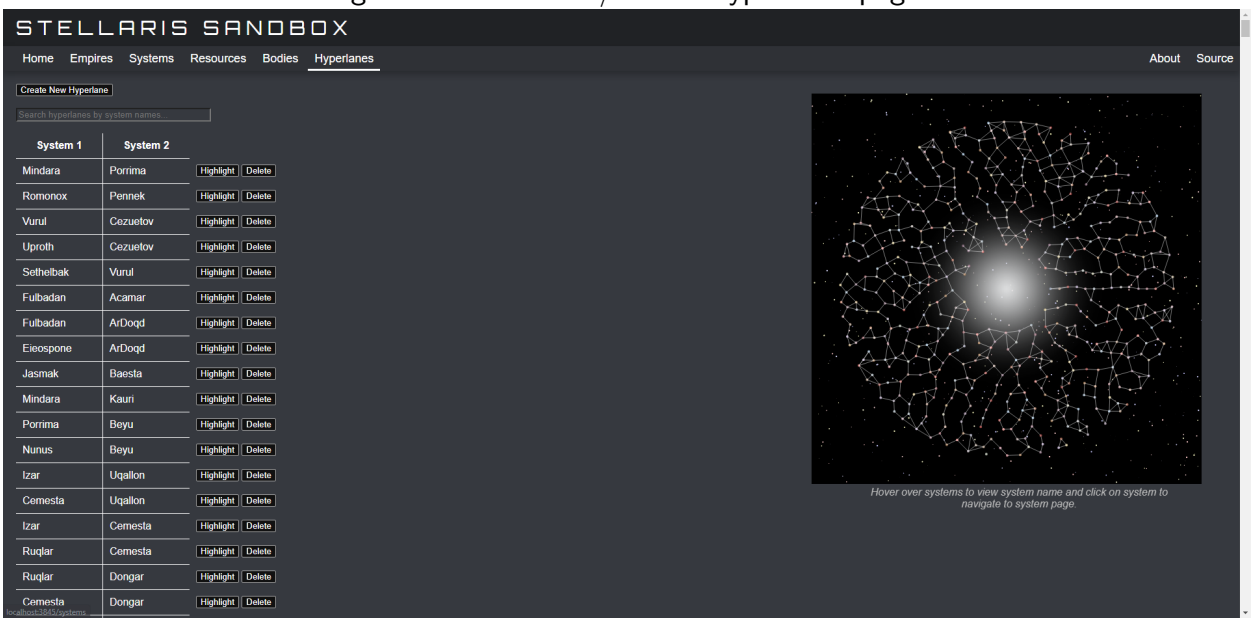


Figure 24: DELETE Hyperlane page.

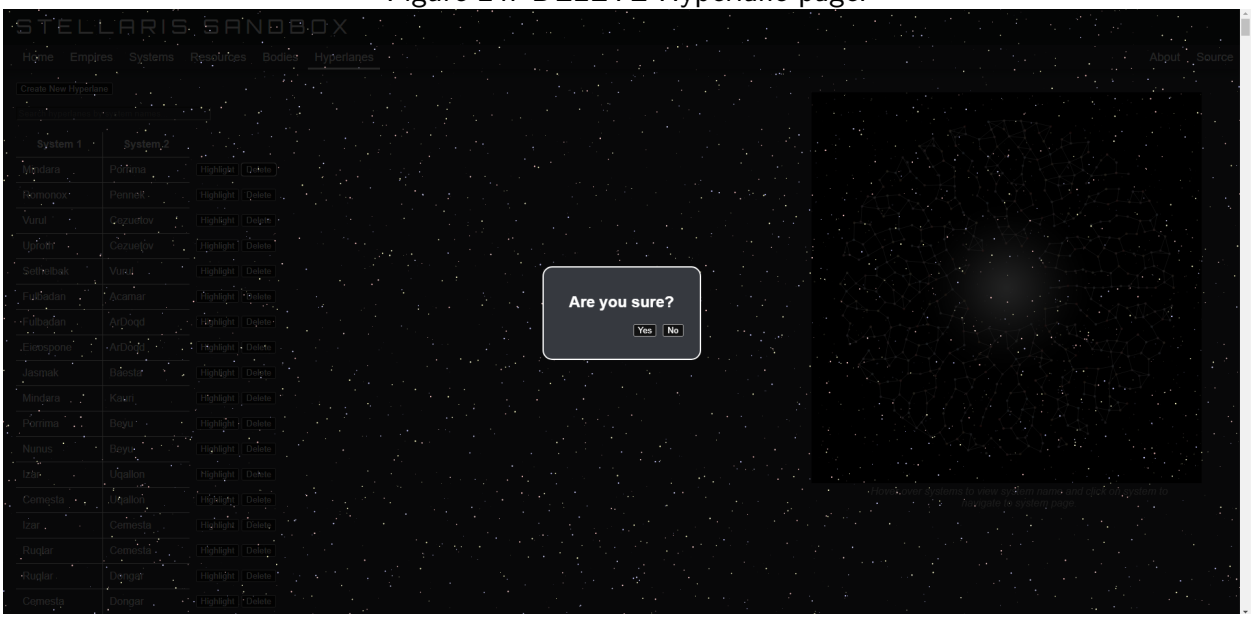
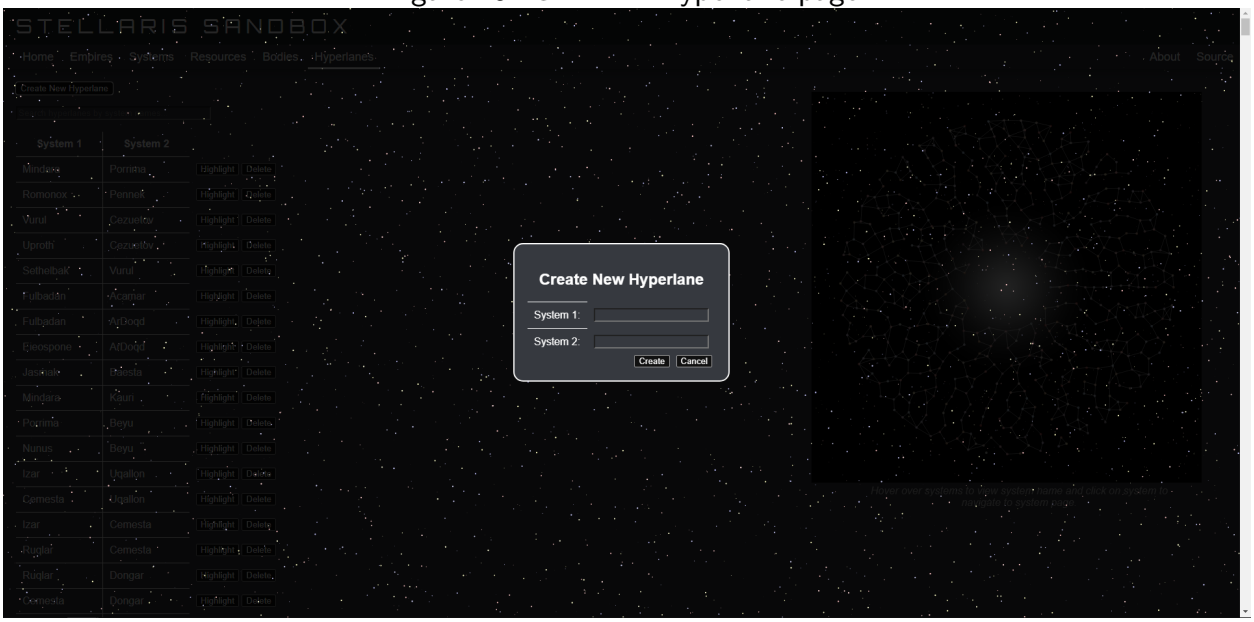


Figure 25: CREATE Hyperlane page.



6.7 Miscellany

Figure 26: About page.

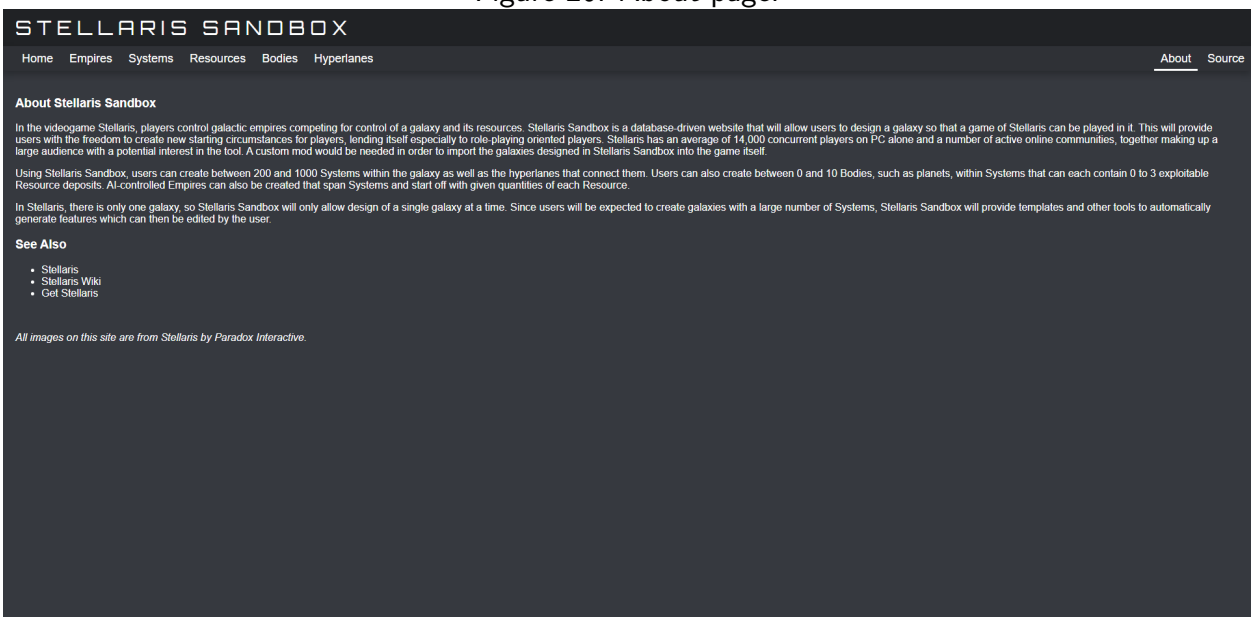


Figure 27: 404 page.