SFML SPRITE -for the interface design of our Game
Below are the explanations:.

What is SFML?

SFML (Simple and Fast Multimedia Library) is a library used to create graphics, windows, and handle input/output for game or application development. It lets us display images, shapes, text, and more.

```
1    #include <SFML/Graphics.hpp>
```

In your code, the SFML sprite design is used for **background images** and **text displays**. Here's an explanation of each related function and how it connects to the graphics.

Function: `loadBackground`

This function is responsible for loading an image file (like `.jpg` or `.png`) to use as a **background** for different parts of your application.

```
// Function to load and scale a background image
sf::Sprite loadBackground(const string& filePath, sf::Texture& texture, float scaleX, float scaleY) {
    if (!texture.loadFromFile(filePath)) {
        cout << "Error: Could not load " << filePath << endl; // Error loading background image
    }
    sf::Sprite sprite;
    sprite.setTexture(texture); // Set the texture for the sprite (background)
    sprite.setScale(scaleX, scaleY); // Scale the sprite to fit the window size
    return sprite;
}
```

Key Points:

- **`sf::Texture`**: Holds the image data loaded from a file.
- **`sf::Sprite`**: A graphical object that displays the image on the screen.
- **`setScale(scaleX, scaleY)`**: Changes the size of the image (e.g., to fit the window).

Example of `loadBackground` Usage

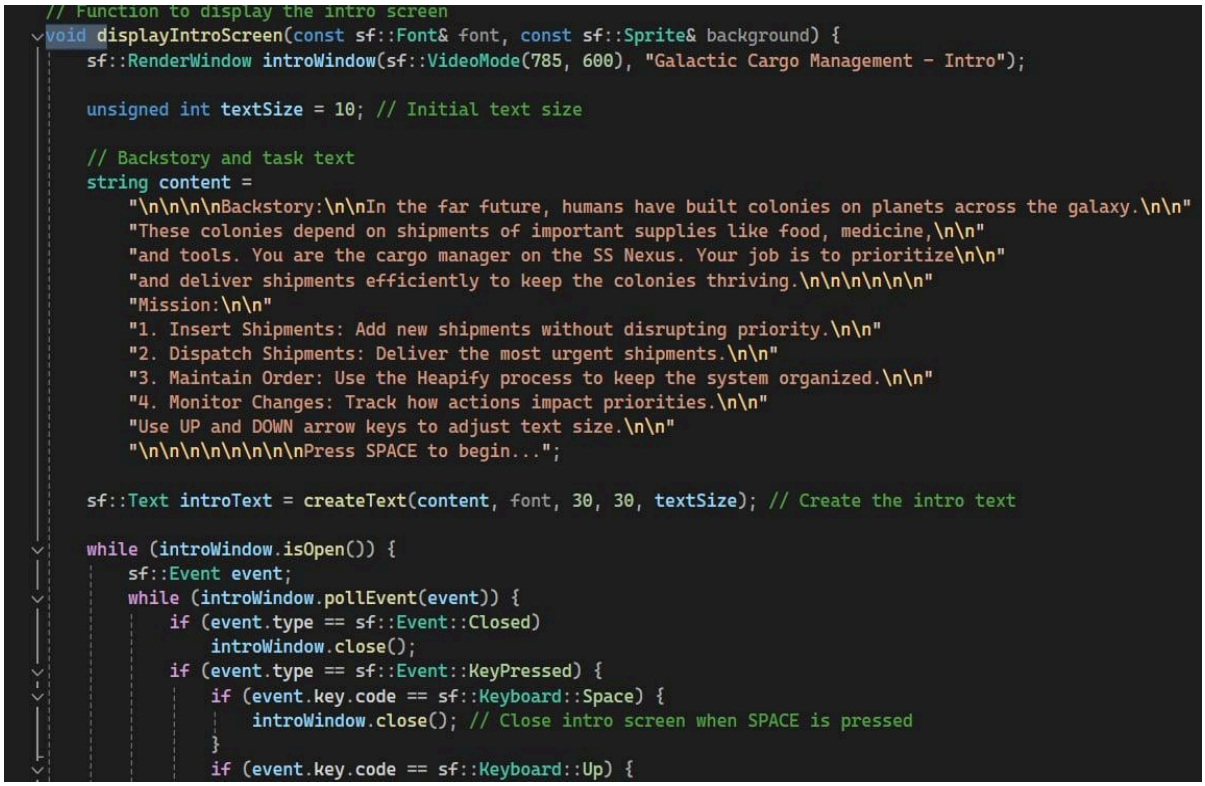
```
// Load background images for various screens
sf::Texture introTexture, menuTexture, insertTexture, dispatchTexture, viewTexture;
sf::Sprite introBackground = loadBackground("intro.jpg", introTexture, 0.5f, 0.6f);
sf::Sprite menuBackground = loadBackground("startup.jpg", menuTexture, 0.5f, 0.6f); // Adjust the scale to fit your window size
sf::Sprite insertBackground = loadBackground("insert.png", insertTexture, 1.7f, 1.5f);
sf::Sprite dispatchBackground = loadBackground("dispatch.png", dispatchTexture, 1.6f, 1.6f);
sf::Sprite viewBackground = loadBackground("view.png", viewTexture, 1.9f, 1.9f);
```

- **`"intro.jpg"`**: The file path for the image.
- **`introTexture`**: Stores the image data.
- **`introBackground`**: A sprite that will display the scaled image (half size on X-axis, 60% on Y-axis).

Later, `introBackground` is drawn using `window.draw(introBackground)`.

Function: `displayIntroScreen`

This function shows the **intro screen** with a background image and some text. Here's how it works:



```
// Function to display the intro screen
void displayIntroScreen(const sf::Font& font, const sf::Sprite& background) {
    sf::RenderWindow introWindow(sf::VideoMode(785, 600), "Galactic Cargo Management - Intro");

    unsigned int textSize = 10; // Initial text size

    // Backstory and task text
    string content =
        "\n\n\n\nBackstory:\n\nIn the far future, humans have built colonies on planets across the galaxy.\n\n"
        "These colonies depend on shipments of important supplies like food, medicine,\n\n"
        "and tools. You are the cargo manager on the SS Nexus. Your job is to prioritize\n\n"
        "and deliver shipments efficiently to keep the colonies thriving.\n\n\n\n\n\n"
        "Mission:\n\n"
        "1. Insert Shipments: Add new shipments without disrupting priority.\n\n"
        "2. Dispatch Shipments: Deliver the most urgent shipments.\n\n"
        "3. Maintain Order: Use the Heapify process to keep the system organized.\n\n"
        "4. Monitor Changes: Track how actions impact priorities.\n\n"
        "Use UP and DOWN arrow keys to adjust text size.\n\n"
        "\n\n\n\n\n\n\n\nPress SPACE to begin...";

    sf::Text introText = createText(content, font, 30, 30, textSize); // Create the intro text

    while (introWindow.isOpen()) {
        sf::Event event;
        while (introWindow.pollEvent(event)) {
            if (event.type == sf::Event::Closed)
                introWindow.close();
            if (event.type == sf::Event::KeyPressed) {
                if (event.key.code == sf::Keyboard::Space) {
                    introWindow.close(); // Close intro screen when SPACE is pressed
                }
                if (event.key.code == sf::Keyboard::Up) {
```

```
introWindow.clear();
        introWindow.draw(background); // Draw the background image
        introWindow.draw(introText);  // Draw the intro text
        introWindow.display();
    }
}
```

Key Points:

- `sf::RenderWindow`: Creates a window for displaying the intro screen.
- `background`: The sprite passed to this function, which will be drawn on the window.
- `introWindow.draw(...)`: Displays the background and text.

Function: `createText`

This function simplifies how you create text to display in your app.

```
// Function to create text for displaying in the window
sf::Text createText(const string& content, const sf::Font& font, float x, float y, unsigned int size) {
    sf::Text text;
    text.setFont(font); // Set the font for the text
    text.setString(content); // Set the text string
    text.setCharacterSize(size); // Set the font size
    text.setFillColor(sf::Color::Green); // Set the text color to green
    text.setPosition(x, y); // Set the position of the text in the window
    return text;
}
```

Key Points:

- `sf::Font`: Holds the font style for the text.
- `setPosition(x, y)`: Positions the text on the window.
- `setCharacterSize(size)`: Adjusts the text size.

How Backgrounds and Sprites are Used in Main Menu

In the main menu, the **background sprite** and text are displayed as follows:

sf::Sprite menuBackground = loadBackground("startup.jpg", menuTexture, 0.5f, 0.6f);

```
sf::Text menuText = createText(
    "1. Insert Shipment\n\n"
    "2. Dispatch Shipment\n\n"
    "3. View Heap ",
    font, 350, 150);
sf::String userInput;
```

1. **`loadBackground`**: Loads the `startup.jpg` image, scales it, and stores it in `menuBackground`.
2. **`createText`**: Creates text for the menu options, positioning it at `(350, 150)`.

Inside the event loop, the window is cleared and the sprite and text are drawn:

```
window.clear();
window.draw(menuBackground); // Use the menu background here
window.draw(menuText); // Display menu options
window.draw(inputText); // Display user input
window.display();
```

Summary of Key SFML Concepts

1. **Textures**: Load image files using `sf::Texture`.
2. **Sprites**: Display those images on the screen using `sf::Sprite`.
3. **Windows**: Create a window using `sf::RenderWindow` to draw sprites and text.
4. **Drawing Order**: Use `window.draw(...)` to render graphics, starting with backgrounds first, then other elements like text.