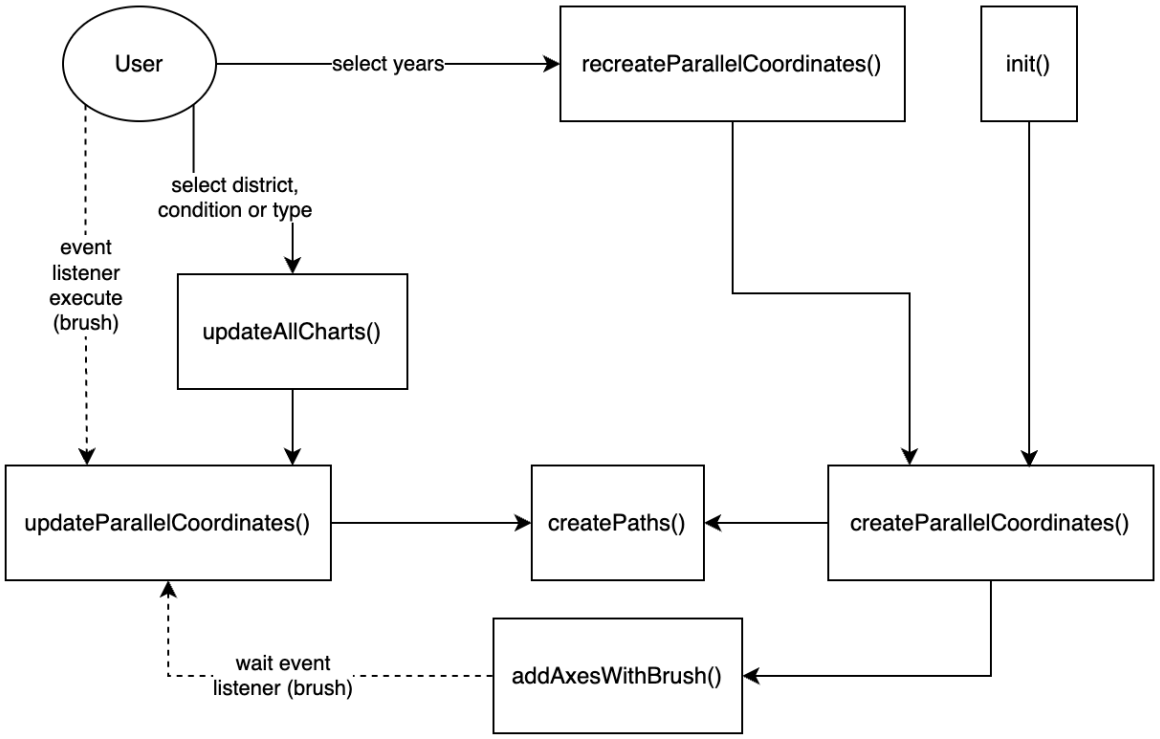
Checkpoint III: First Prototype

Group: G23
Date: 2024/10/08

Prototype Architecture



The architecture of this dashboard is designed to efficiently manage data loading, processing, and visualization using D3.js. The process begins in the **init()** method, where data is loaded using the **d3.json()** function and stored globally in the **global_data** variable, making it accessible for various visualizations.

When the user interacts with the menu controller, they can apply different filters, resulting in a new auxiliary dataset stored in **filtered_data**. If the filter involves selecting specific years, the original dataset in **global_data** is recalculated to reflect the new selections.

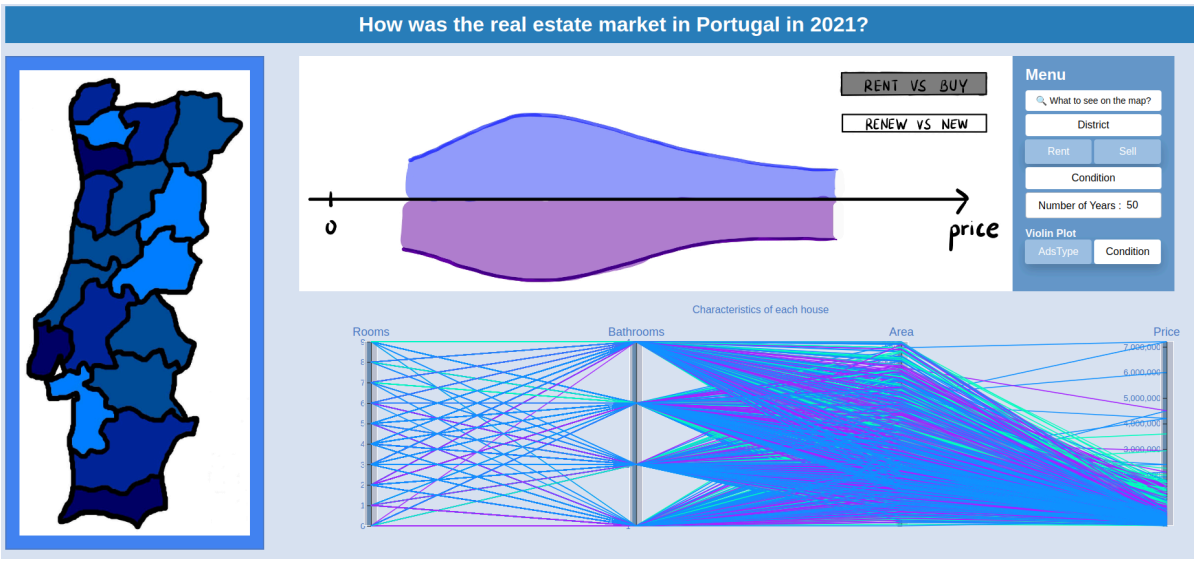
The parallel coordinates chart is created by the **createParallelCoordinates()** function, which uses **global_data** and applies default filter values. Within this function, **createPaths()** is called to draw the connections between the axes, and **addAxesWithBrush()** sets up the axes and interactive brush events, allowing users to filter directly on the chart.

When the user uses the brush to filter visible data, the **updateParallelCoordinates()** function is triggered, redrawing the chart based on the new selection. Additionally, if the user changes the number of years they plan to live in the house, the **recreateParallelCoordinates()** function is called to recalculate **global_data** and dynamically adjust the axes.

Whenever other filters in the menu controller are changed, the **updateAllCharts()** function is executed. Currently, it only updates the parallel coordinates chart but is designed to handle the future updates of other visualizations on the dashboard.

The code is organized in a modular way: functions related to the parallel coordinates are in **parallel_coordinates.js**, filtering functions are in **filter.js**, interactions from the menu controller are managed in **controllers.js**, and global variables, such as global_data, are defined in **script.js**. This modular structure makes the system both easy to maintain and expand.

Dashboard Layout



(dashboard layout updated)

Initial Structure:
In this section, we present the initial structure of the Dashboard. The system has already implemented the Parallel Coordinates graphic.

Visual Placeholders:
For the remaining graphic components, which are still under development, we use visual placeholders in the form of rectangles. These placeholders indicate the space that the final graphics will occupy in the layout, providing a clear idea of the organization and distribution of each component. The arrangement of the rectangles is designed to optimize the use of space and maintain visual coherence, guaranteeing a smooth transition when the final graphics are implemented.

Data Processing

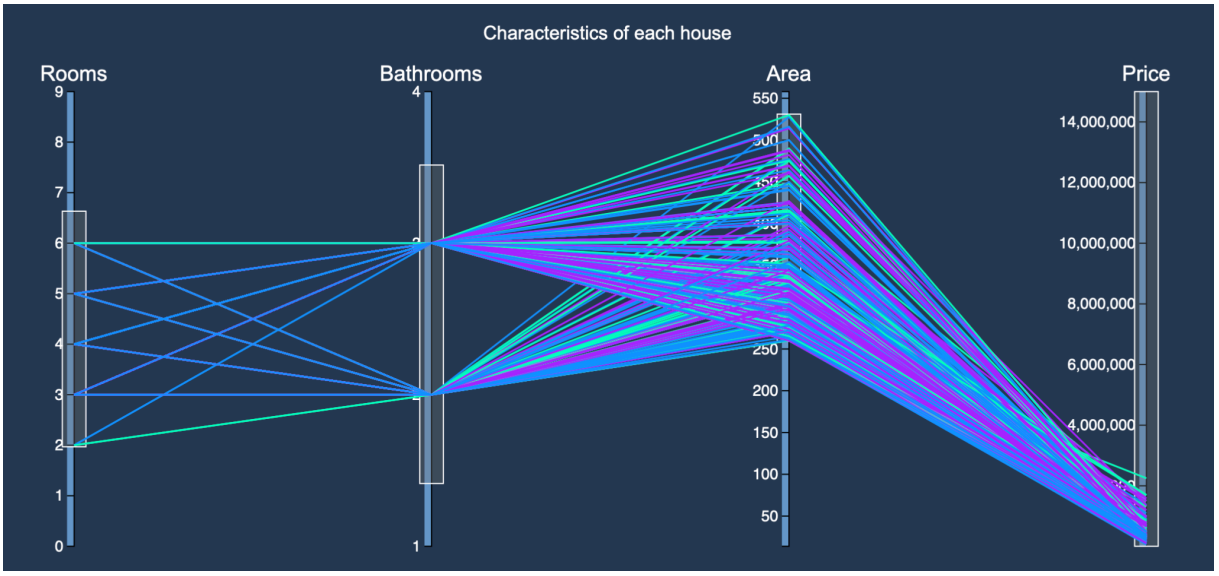
- Transforme the final dataset into .json because it supports complex data structures and integrates seamlessly with JavaScript, making data handling more efficient.
- Perform Data Processing Once: Data processing is done once, saving the dataset for consistent analysis. While most data remains static, price data can change over time based on the year, allowing dynamic updates in price trends.
- Remove:
    a. **outliers**: Outliers can skew analysis results, leading to misleading interpretations.

b. **Açores and Madeira islands**: These islands have distinct characteristics that may not represent the mainland. Excluding them maintains the relevance of the data for the principal cities.
c. **other conditions**: Removing non-relevant conditions simplifies the analysis, allowing for clearer and more applicable results to the main context.
d. **items that are not in the principal cities**: Excluding items outside the principal cities ensures that the analysis focuses on the most relevant and representative data for the target urban environment.

- Added 'id' column: For future interaction between the Parallel Coordinates and Choropleth Map.
- Added 'Zone': Portugal its divided in 3 zones: North, Center and South

Compared to Checkpoint 1, we decided not to include Municipality because we did not have enough data to cover all municipalities in Portugal. As a result, we chose to limit filtering the information by District instead.

Chart Interaction



We have the following interactions in the axes, depending on the min. and max. value of the attribute: Rooms (quantity), Bathrooms (quantity), Area (m^2) and Prices (euro, €).

The user can select the range they are interested in by pulling the maximum and minimum limits of each axis, displayed as a gray bar in Parallel Coordinates.

Chart Integration

We manage global data and filtered data within script.js. Each time the data is filtered, it is updated accordingly. Storing data as a global variable in a single file simplifies dataset access, making it easier for all languages to retrieve the necessary information.

Modular Functions: Each function operates through dedicated functions:

- init(): Loads the dataset and processes it.
- processData(data): Converts specific values in the dataset to numbers.
- updateChart(data): Updates the chart based on the filtered data.
- createParallelCoordinates(selector): Initializes the parallel coordinates chart and sets up the SVG container and scales.

Centralized Update Mechanism: The updateAllCharts(data) function serves as a central hub that synchronizes all chart updates. By integrating individual update functions into this centralized function, we enable efficient interactions across different charts.

Shared Filtered Data: All charts share the same filtered data structure. Any updates made to the data will automatically reflect across all charts. This design simplifies programming, allowing developers to focus on the specific logic for new functionalities without worrying about duplicating data processing.

Controllers Menu

- When selecting a specific characteristic from the Controllers Menu, the corresponding dataset in Parallel Coordinates is updated. For example, if "Rent" is selected, only houses available for rent will be displayed in Parallel Coordinates.
- Characteristics can be:
    - What to see on the map? (will be implemented for Choropleth Map)
    - Districts (18 districts of Portugal)
    - Rent or Buy (if user wants to rent or buy or both)
    - Condition (New, Renovated or Used)
    - Number of Years (min. of 1 year and max. of 99 years)

Parallel Coordinates

- When hovering a line in Parallel Coordinates, the corresponding line of the house will be marked by a white color, and if the line is not being hovered over, it will be marked a contour line of the same color that stands out on the zone in Choropleth Map.
- When the Parallel Coordinates are changed, the data displayed will be filtered as the selection of characteristics.