Easy Gantt, v. 1.5.2 – Features description

Easy Gantt provides a rich set of features for project planning and visualization directly within a web browser, without requiring any server-side processing. It focuses on user-friendliness and extensive customization options to tailor the Gantt chart to specific project needs and visual preferences.

Users can input task data in CSV format and the script will generate a visual Gantt chart with customizable settings. The tool allows for defining project settings, chart appearance, bar styles, task labels, and date formats, and includes features for saving and loading projects as JSON files and exporting the chart as an SVG or tasks as CSV. It also incorporates handling of task dependencies, critical paths, and non-working days.

Here's a detailed summary of its functionalities:

1. Data Input and Parsing:

- The tool allows users to input project task data in a **comma-separated or tab-separated values (CSV)** format via a <textarea>.
- The expected format for each task includes **Task Id, Task name, Duration, Percent complete, and Predecessors**.
- It includes a JavaScript function parseCSV that **parses this input data** into an array of task objects.
- The parser handles **different dependency types** between tasks specified in the Predecessors field, including **Finish-to-Start (FS), Start-to-Start (SS), Finish-to-Finish (FF), and Start-to-Finish (SF)**. The default dependency type is FS.
- It also supports **lead and lag times** for dependencies, specified in days (e.g., 1FS-1 day, 2FF+1 day).
- The parseCSV function calculates **earliest start and finish dates** for each task based on dependencies and the project start date, considering non-working days.
- It also calculates **latest start and finish dates** using a backward pass algorithm.
- The function identifies **milestone tasks** (tasks with a duration of 0 days).

2. Gantt Chart Visualization:

- The tool generates a **Gantt chart as an SVG (Scalable Vector Graphics) element** within the svgContainer div.
- Each task is represented by a **horizontal bar** on a timeline, with the bar's length corresponding to the task's duration.
- The bars are visually divided to show the **percentage of task completion** using two different configurable colors: one for the completed portion and another for the not completed portion.
- **Critical path tasks** are highlighted with a distinct configurable color within their bars.

- **Dependencies** between tasks are visualized as lines connecting the end of a predecessor task's bar (or milestone) to the start of a successor task's bar (or milestone), with optional arrow markers. The style of these lines (solid or dashed) is also configurable.
- **Milestones** are represented by **rhombus shapes**.
- Task **start and end dates** are displayed as text labels near the corresponding task bars or milestones.
- The chart includes a **horizontal timeline** at the top, displaying dates based on the project's duration and the configured date format.
- Optional **vertical timeline lines** can be displayed at each date marker.
- Optional **horizontal separator lines** can be drawn between the task bars.

3. Interactive User Interface and Customization:

- The tool provides a **tabbed interface** (tab-container) to organize various project and chart settings.
- **Project Settings (Tab 1):**
    - Allows setting the **Project Start Date**.
    - Allows setting a **"Today Date"** which is used to calculate the ideal percent complete and identify delayed tasks.
    - Provides a button to open a **"Calendar" popup** for configuring non-working dates (weekends and specific dates/periods).
    - An option to **multiply the "Percent complete" values in the input by 100**.
- **Chart Settings (Tab 2):**
    - Allows customization of the **chart width**, **left margin**, **vertical and horizontal axis zero positions**, **right margin**, **header vertical space**, and **header dates space**.
    - Users can select the **colors for vertical timeline lines** and **horizontal separator lines**, with an option to disable them.
- **Bars Settings (Tab 3):**
    - Allows setting the **height** and **spacing** between the task bars.
    - Users can choose the **colors for the completed and not completed portions of the bars**, with an option to hide either.
    - The **border color** and **thickness** of the bars can be customized, with an option to have no border.
    - The **corner angles** of the bars can be adjusted.

    - The **color** and **width** of the critical path highlighting within the bars are configurable, with an option to disable it.
    - The **color** and **line style (dashed or solid)** of the dependency lines can be set, with an option to hide them.
- **Tasks Settings (Tab 4):**
    - Allows customization of the **font size**, **font family**, **font style (normal, italic, oblique)**, and **font weight (normal, bold, lighter, bolder)** for the task labels.
    - The **color of the task labels** can be selected, with an option to hide them. A separate **delayed color** can be chosen for task labels that are behind schedule.
    - The **vertical alignment** of task and date labels relative to the bars can be adjusted.
    - The **number of characters to truncate** task names can be specified.
    - Checkboxes allow users to optionally display **detailed additional task information** next to the task names, including Actual Percent Complete & Ideal Percent Complete, Duration & Elapsed Duration & Total Float, and Predecessors.
- **Dates Settings (Tab 5):**
    - Users can select the **format** for the date labels displayed on the chart from a dropdown list (e.g., MM/DD/YYYY, DD/MM/YYYY).
    - The **font size**, **font family**, **font style**, and **font weight** for the date labels can be customized.
    - The **color of the date labels** can be chosen, with an option to hide them.
    - The **horizontal spacing** of the date labels from the left and right of the task bars (or milestone) can be adjusted.

4. Non-Working Dates Management:

- The tool provides a **popup window** to configure non-working days.
- Users can specify whether **Saturdays** and/or **Sundays** should be considered non-working days via checkboxes.
- They can also define **specific date ranges** as non-working periods by adding multiple "From" and "To" date inputs.
- A button is provided to **add more date range rows**, and a **"Clear" button** allows resetting the non-working date configurations.
- The configured non-working days are stored and taken into account when calculating task durations and start/end dates.

5. Data Persistence and Export:

- **Open JSON:** A button labeled "Open JSON" allows users to **load previously saved project settings** (including non-working dates and all customization options) from a JSON file.
- **Save as JSON:** A button labeled "Save as JSON" enables users to **save the current project settings** to a JSON file, preserving all the configured options. This allows users to easily restore their project's appearance and settings.
- **Save as SVG:** The "Save as SVG" button allows users to **export the generated Gantt chart as a vector-based SVG file**. This format is scalable without loss of quality and suitable for printing or further editing.
- **Save as CSV:** The "Save as CSV" button allows users to **export the task data, along with calculated information** like elapsed duration, total float, earliest/latest start/finish dates, critical status, delay status, and milestone status, back into a CSV file.

6. Core Functionality and Utilities:

- The code includes JavaScript functions to **calculate the timeline** (min and max dates, unique start dates) based on the task data.
- It provides utility functions for **date formatting** (formatDate) according to the selected date format.
- Helper functions like todayDate and todayDateMinusOneMonth are used to set default values for the "Today Date" and "Project Start Date" inputs.
- Basic **error handling** is implemented to catch issues during CSV parsing and provide user feedback via alerts.
- The code uses the **File System Access API** (where supported by the browser) to provide a more integrated file saving and opening experience, prompting users to select file names and locations. It includes a fallback mechanism for browsers that do not support this API.