Shape Viewer Application

Overview

The Shape Viewer Application allows users to load and create various shapes (Rectangle, Circle, Polygon, Triangle) on a dynamic canvas. It features a user-friendly interface for creating, modifying, and rendering shapes, with support for saving and loading custom shape files. The application supports several bonus features such as shape deletion, translation, and clearing the viewport.

Features Implemented:

Iteration 1: User Interface Layout

The user interface is divided into three main components:
- Top Toolbar: Displays the name of the loaded shape file or provides an 'Open Shape File' button when no file is loaded.
- Left Menu: Contains a form for creating new shapes. Users can specify the shape type (Rectangle, Triangle, Circle), size, position, and color.
- Shape Viewport: Occupies the remaining screen space and dynamically resizes with the browser window. This area is where shapes are rendered after being loaded or created.

Iteration 2: Shape File Format

A custom shape file format is used to store shape data. The file format includes:
- Shape Type: Rectangle, Circle, Polygon
- Position: X and Y coordinates
- Size: Width/height for rectangles or vertex coordinates for polygons
- Color: Specified in hexadecimal format

Tested Demo Data:
Polygon,100,100,5,110,140 140,130 150,150 120,170 110,180,ff5733;
Polygon,200,150,4,250,200 270,180 290,170 270,220,ffff00;
Rectangle,100,100,0,60,80,ff9900;
Rectangle,570,200,1,100,100,00ccff;
Circle,400,400,undefined,50,undefined,0;
Polygon,450,350,5,500,400 530,380 550,400 530,420 500,410,ff66ff;

Testing files are included in the Public folder of code.

Iteration 3: Shape Rendering

Shape Rendering: Shapes are rendered in the Shape Viewport using HTML Canvas.
Supported shapes include Rectangles, Circles, Polygons, and Triangles.
Shapes are dynamically positioned based on the viewport and file specifications.

Iteration 4: Polygon Support

Polygon Rendering: The shape file format was extended to support polygons with a list of vertices.
Polygons are rendered by connecting vertices with lines.

Iteration 5: UI Enhancement

UI Design: Enhanced the UI for a modern and visually appealing experience.
Applied consistent color schemes, typography, and spacing.
Ensured responsiveness across devices.
Added hover effects and transitions for better interactivity.

Bonus Features:

Bonus Feature 1: Shape Translation

Manual Shape Translation: Users can drag shapes to a new location on the canvas.
Save as Option: Users can save the updated shape file using the 'Save as' button in the toolbar.

Bonus Feature 2: Shape Creation

Create New Shapes: Added functionality to create new shapes via a form in the Left Menu. Users can specify type, size, position, and color for the shape.
Shape Integration: New shapes are integrated into the viewport and memory.

Bonus Feature 3: Extra Features

Delete Button: A toggle delete button has been added to the toolbar. When clicked, the button's background color changes to red. Users can then click on any shape to delete it. The 'Save as' button allows users to save the changes after deletion.
Clear Shape Button: A button to clear all shapes from the viewport. This feature can be extended for additional functionality in the future.
Circle Rendering and Creation: Logic for rendering and creating Circles has been added.

How to Use:

1. Loading a Shape File: Click the 'Open Shape File' button in the top toolbar to load a shape file.
2. Creating a New Shape: Use the form in the Left Menu to create new shapes by specifying the shape type, size, position, and color.
3. Shape Editing:

- Drag shapes to new locations to translate them.
- Delete shapes by toggling the delete button, which turns red. Then, click on the shape you want to delete.
- Clear all shapes from the viewport using the 'Clear Shape' button.

4. Saving Changes: After editing, use the 'Save as' button in the toolbar to save the updated shape file.

Known Problems:

1. Triangle Rendering Issue: Issues with triangle rendering when opening the files. This will be addressed in a future update.

2. Shape Creation Fixes:

- Triangles: Vertices were initially miscalculated. This was fixed by dynamically computing the three vertices based on the position, base, and height inputs.
- Circles: The radius was incorrectly set to the full width. This was corrected by dividing the width by 2 to get the proper radius.

Improvements & Future Plans:

1. Multiple File Support: Implement logic for opening multiple files simultaneously by dividing the viewport into separate sections.

2. Save As Location: The current 'Save as' feature only downloads the file to the default download location. In the future, this can be enhanced to allow users to choose their preferred save location.

Implementation of Save As Feature:

To implement a 'Save As' feature where users can choose the save location, we can use the **FileSystemAccessAPI**, available in modern browsers like Chrome and Edge. This API allows the user to select a location and save files directly to their desired folder. For broader compatibility, we can use the **Blob** and **download** attribute method, which triggers a download but defaults to the user's download folder. The **FileSystemAccessAPI** is recommended for more control over the file location in supported browsers.

Files Attached:

1. Sample data files (.shapefile) included in the public folder of a code for testing.
2. Demo video also attached in zip file under shape-viewer folder.