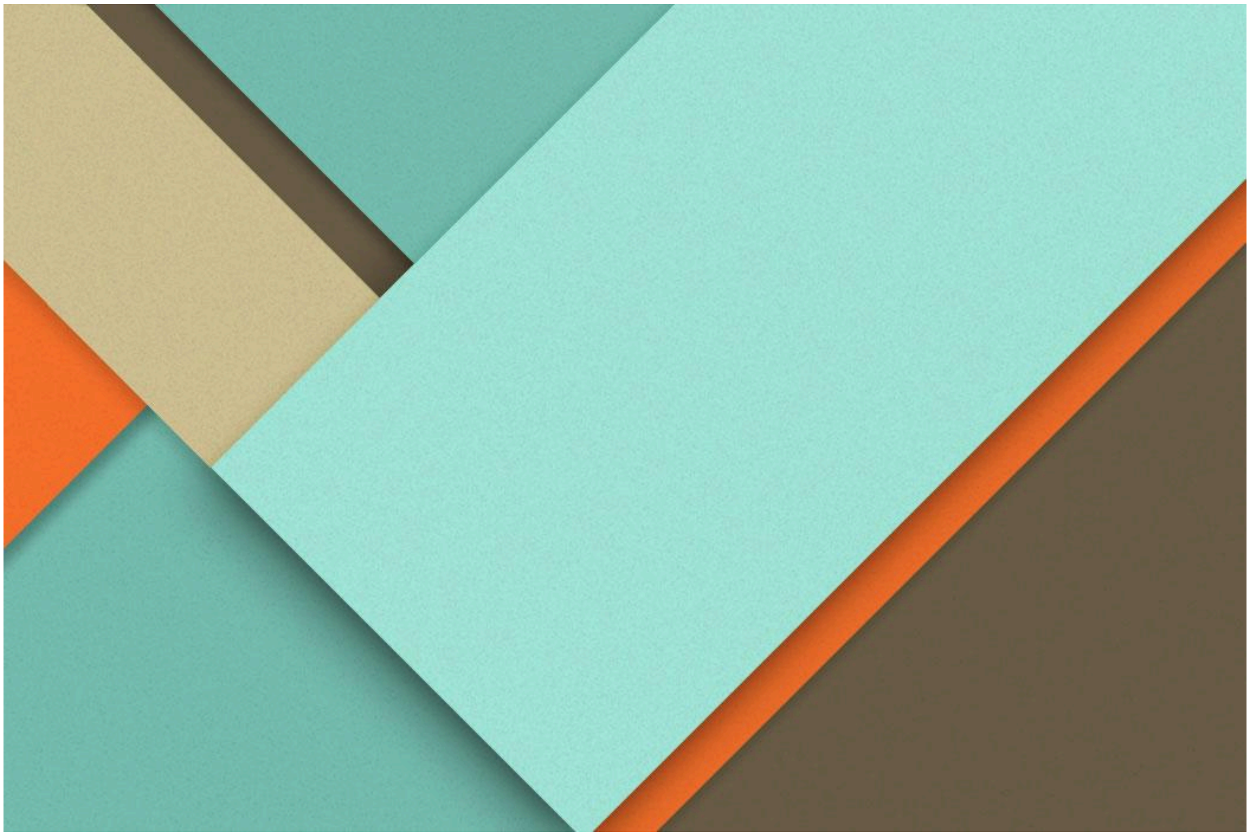

Technical Documentation for Selection Sort Visualization Application

CS346: Software Engineering Lab

Assignment 1 :: Milestone 3 Full Implementation of working software

—

M Devi Naga Sai Srinivas (210101070)
B.Tech 3rd Year,
Dept of CSE, IITG

Overview

The Selection Sort Visualization Application is a Windows Form application developed using Visual Studio and Visual Basic. The primary purpose of this application is to visually demonstrate the selection sort algorithm by allowing users to input an array of integers and observe the sorting process step by step.

Features

1. Selection Sort Algorithm

The core functionality of the application is the implementation of the selection sort algorithm. The algorithm iteratively selects the minimum element from the unsorted portion of the array and swaps it with the element at the current iteration index.

2. Step-by-Step Visualization

The application provides a step-by-step visualization of the selection sort process. Users can advance through the iterations of the algorithm, observing the changes in the array and the highlighting of selected elements.

3. User Input

Users can input an unsorted array of integers with a specified size. The application validates user input to ensure it meets the required criteria. Once the array is entered, users can initiate the sorting process.

4. Dynamic UI Updates

The user interface dynamically updates during the sorting process to reflect the current state of the array. RichTextBox controls are used to display the elements of the array, with different colors indicating the current iteration, selected elements, and elements being swapped.

5. Restart Functionality

The application includes a restart button, allowing users to restart the application and input a new array. A warning message is displayed to confirm the user's intention before restarting.

Code Structure

1. Form1 Class

The primary class encapsulates the application's logic, UI elements, and event handlers.

2. Private Variables

- **array() As Integer**: Represents the array to be sorted.
- **currentIndex As Integer**: Tracks the current index during the sorting process.
- **currentIteration As Integer**: Tracks the current iteration of the selection sort algorithm.

3. SelectionSort Method

Performs one iteration of the selection sort algorithm. It updates the UI and array state after each iteration.

4. Event Handlers

- **btnNextIteration_Click**: Handles the button click event to advance to the next iteration.
- **Strt_Click**: Initializes the application, captures user input for the array size, and configures UI elements.
- **btnAddNext_Click**: Handles the button click event to add the next element to the array.

- **btnUnsortedDisp_Click**: Displays the unsorted array for user review before starting the sorting process.
- **btnRestart_Click**: Handles the button click event to restart the application.

5. Display Methods

- **DisplayIteration**: Updates the RichTextBox controls to display the current array state.
- **DisplayMinIndexSearch**: Highlights the elements being compared while searching for the minimum index.
- **DisplaySwapping**: Highlights the elements being swapped and displays the indices.

Usage

1. Launch the application and input the array size (between 1 and 10).
2. Enter the elements of the array one by one.
3. Click "Next Iteration" to step through the selection sort process.
4. Review the sorted array after all iterations are complete.
5. Optionally, click "Restart" to input a new array.

Conclusion

The Selection Sort Visualization Application serves as an educational tool to help users understand the selection sort algorithm through interactive visualizations. The user-friendly interface and dynamic updates make it an effective learning resource for individuals interested in sorting algorithms.