1. Visualization of operations on tree data structures

Overview: Tree is a useful data structure with lots of applications in computer science. In this project, you will design a program to display and explain some basic operations of four types of tree structure.

Basic knowledge:

- Generic tree (no special properties): A tree is a nonlinear hierarchical data structure that consists of nodes connected by edges and contains no cycles.
- A Binary Search Tree (BST) is a binary tree in which each vertex has only up to 2 children that satisfy the BST property: All vertices in the left subtree of a vertex must hold a value smaller than its own and all vertices in the right subtree of a vertex must hold a value larger than its own (we have an assumption that all values are distinct integers in this visualization and small tweak is needed to cater for duplicates/non-integer).
- An Adelson-Velskii Landis (AVL) tree is a self-balancing BST that maintains its height to be O(log N) when having N vertices in the AVL tree. The heights of two subtrees from its root nodes are limited to a "certain distance". The child nodes are rebalanced when the height difference exceeds a certain distance.
- Balanced binary tree: A balanced binary tree has the properties of both a BST and an AVL tree

Operations on the above tree structures: create, insert, delete, update, traverse.

Specifications:

- GUI: You can refer to this source for some ideas: https://visualgo.net/en/bst
- Design:

+ We only consider undirected-weight trees, with integer node values and no duplicated node values allowed.

+ For the balanced tree and balanced binary tree, the maximum difference in distance from the root of the leaf nodes must be chosen by the user

+ On the main menu: title of the application, navigation bar for user to choose between the four types of tree, help menu and quit

- User must select a type of data structure before getting into the visualization
- The help menu shows the basic usage and aim of the project
- Quit button exits the application. Remember to ask for confirmation

+ In the visualization

- Users can choose to visualize one of six operations, by selecting an option on the operations menu, and then providing the necessary parameter.
  The description of the operations is as follows:

Operations	Parameters	Description
Create	None	Create a new empty tree
Insert	Value of parent node, value of new node	Add the new node with a specified value as a child of the specified parent node
Delete	Value of the node	Delete the node from the tree
Update	Current value of the node, new value of the node	Change the node with current value to new value

Traverse	Algorithm (DFS or BFS)	Traverse all nodes in the tree (highlight the current node in each step of traversal)
Search	Search value	Search for the node value in the tree

- When an operation starts to execute, on the code panel, the pseudo-code (or actual code) should be displayed, and the currently executing line is highlighted to help the user keep track of the process. On the bottom bar, the user can see the progress bar of the executing operation and choose to pause, continue, or go backward or forward a step in the execution.
- The user can also undo or redo operations from the bottom bar.
- Always have a Back button for the user to return to the main menu at any time