**B-Tree and B+-Tree Visualization Project Outline**

Samuel Morrison

Brayden Peck

Conley O’Neill

Thomas Wood

Table of Contents

[Project Description 3](#_Toc179405185)

[Who are the Users? 3](#_Toc179405186)

[Goals 3](#_Toc179405187)

[Requirements 3](#_Toc179405188)

[External Design 4](#_Toc179405189)

[Banner 4](#_Toc179405190)

[Tree Selection 4](#_Toc179405191)

[Degree Selection 4](#_Toc179405192)

[Insertion 5](#_Toc179405193)

[Deletion 5](#_Toc179405194)

[Find 6](#_Toc179405195)

[Clear 6](#_Toc179405196)

[Visualization Window 6](#_Toc179405197)

[Step/Skip Forward 6](#_Toc179405198)

[Step/Skip Backward 6](#_Toc179405199)

[Pause 6](#_Toc179405200)

[Animation Speed Slider 6](#_Toc179405201)

[Colors 6](#_Toc179405202)

[Potential GUI Changes and Improvements 7](#_Toc179405203)

[Internal Design 7](#_Toc179405204)

[Classes 7](#_Toc179405205)

[BTreeNode and BPlusTreeNode 7](#_Toc179405206)

[BTree, BPlusTree, and BStarTree 7](#_Toc179405207)

[MainWindow 7](#_Toc179405208)

[Potential Changes and Improvements 8](#_Toc179405209)

# Project Description

The software project that is the focus of this document is that of a B-Tree, B+-Tree, and B\*-Tree visualizer. The goal of the project is to ultimately provide an effective tool for understanding these complex data structures which may be used by those in academia, especially students who are trying to implement these data structures for their own projects.

## Who are the Users?

The users in this case will primarily be students, including ourselves. However, this software project may be beneficial to anyone who is trying to understand the B-Tree and adjacent data structures, which could range from professors and students to data scientists and more.

## Goals

The goals of this project are straightforward. Below the major goals and objectives of this project are listed out:

1. Create an effective tool that can be used for learning and understanding the different functions of a B-Tree, B+-Tree, and B\*-Tree (insertion, deletion, and searching).
2. Make an easy to use and understand Graphical User Interface that allows the user to step forward and/or backward through the steps of any of the functions for the data structures.

## Requirements

To accomplish the stated goals, there are some requirements that must be met. Those requirements are listed below in order by priority.

1. Have a fully implemented B-Tree in C++.

* Insertion
* Deletion
* Search

1. Have a fully implemented B+-Tree in C++.

* Insertion
* Deletion
* Search

1. Have a fully implemented B\*-Tree in C++.

* Insertion
* Deletion
* Search

1. A working GUI to interact with data structures.

* Ability to change the max number of keys for each node in the tree within the GUI
* Switch between any one of the data structures to another.
* Insert values into the trees.
  + Visualize the process of inserting a value into the tree.
* Delete values from the trees.
  + Visualize the process of deleting the value from the tree.
* Search for a given value in the tree.
  + Visualize the process of searching for the given value in the tree.
* Step forward through any of the visualizations.
* Step backward through any of the visualizations.

# Dependencies

For this project to come to fruition, discussions were had to determine how to program the code required for the project. The decision that was made was to use C++ to program the data structures and QT for C++ for the GUI and visualization. As such there are some required downloads for the project to be compiled or developed on. The list of these dependencies are as follows:

* QT 6.7.2
* C++ v.13 or higher

# External Design

For the external design of the project, we have designed a rough interface for the user to interact with. As the project continues further into development the GUI may change in slight ways but will never stray too far from this original concept. Figure 1 below shows the rough implementation of the GUI, which the user will use to interact with the software. Each interface on the GUI is described in more detail below.

## Banner

The banner is the orange element at the top of the GUI. This will display the title of the application.

## Tree Selection

The tree that is being visualized can be selected from the buttons to the right of the banner. The user can select to visualize a B-Tree, B+-Tree, or a B\*-Tree. Only one option can be selected at one time, and selecting a different option will unselect the previous option. This will also empty the visualization window of any elements that may have been added before changing.

## Degree Selection

To change the degree of the tree (which determines how many values each node can hold) the user can use the drop down just below the tree selection area of the GUI. This will be a dropdown with many different options, starting from the degree value of three. This is because any degree less than three is impossible with how the data structures work. Selecting a different degree will empty the visualization window of any elements that may have been added before changing.

## Insertion

Inserting elements into the tree can be done from the first text box and the button next to it which is titled “Insert”. Clicking this button will then cause the visualization window to add a new node, displaying animations on how the data structure is working.

## A screenshot of the roughed in GUI implementation.Deletion

Figure A screenshot of the rough GUI implementation.

Deleting elements will work just the same as inserting elements. The second text box is where the user can enter a value to remove from the tree. The button next to this text box titled “Delete” will then begin the visualization of how the selected tree would remove that value.

## Find

The third and final text box will be used to find an element within the selected tree. Clicking the third button titled “Find” will visualize the process of searching the tree for the given value in the visualization window.

## Clear

The fourth button beneath the banner is the “Clear” button. This button can be used to empty the entire tree of any elements that may have been inserted before.

## Visualization Window

The visualization window is where the entire visualization process will be shown. From inserts to deletes, everything will appear here. The main visualization window is all the empty space between the upper half of the GUI which contains all of the elements previously described down to the elements yet to be described.

## Step/Skip Forward

These buttons will allow the user to step forward through the steps they’ve done so far. This button will have no function is the user has not stepped backwards yet or if the process has not been paused.

## Step/Skip Backward

These buttons will let the user step backwards through the steps they’ve taken so far (insertion or deletion). This is to allow the user to slow down and revisit steps that may not have made sense during the original animation.

## Pause

This button will allow the user to stop any current animations to allow a further dissection of the process being performed.

## Animation Speed Slider

The final element of the GUI is the animation slider, which can be found in the bottom right corner. This will allow the user to determine the speed at which the animations should be played to help with the visualization process.

## Colors

The colors for the GUI were chosen to represent the school in which this project was developed, Idaho State University.

## Potential GUI Changes and Improvements

To simplify the GUI we may change things as we determine that they are unneeded, or add things as we find things that overall improve the experience for the user. Some of the potential changes are listed below:

* Have one textbox for input rather than three separates for each of the buttons.
* Add a textbox that will be used to display a better description of the current step that is being visualized.
* Remove skip forward and backward buttons, as they may be redundant with the step forward and backward buttons.

The given list is by no means comprehensive but gives a good idea of some changes that are being thought of already. The level of change will never be larger than any of these examples, ideally only changing minor things overall to improve user experience.

# Internal Design

The internal design of the application has been simplified from the previous tree visualization. We have reduced the classes from over ten down to just six. This design may change through development. These six classes are outlined below, with a class diagram attached.

Class Diagram for B Trees Visualization


Figure Class Diagram for B Trees Visualization Project

## Classes

### BTreeNode and BPlusTreeNode

These classes are used to represent the individual elements of their respective trees. They contain all of the information for any node in the trees including the keys, values, children, and parent.

### BTree, BPlusTree, and BStarTree

These classes are where the data structure logic is implemented. These classes will use the BTreeNode and BPlusTreeNode classes to build their respective structured trees, which the tree only keeps track of the root node. From the root node all the functionality of the trees can be performed including insertion, deletion, and searches.

### MainWindow

This is the class which creates the GUI. This will call and use the BTree, BPlusTree, and BStarTree classes to perform the logic of inserting, deleting, and searching these data structures. Hooks or events will be used to send the required information for displaying and visualizing the functionality of the data structures.

### Potential Changes and Improvements

Even with the current design, there are still some things that we have recognized may be better to change with the current structure. One of those changes is to combine the BTreeNode and BPlusTreeNode classes into one class or to create a Node class which they will inherit from. As development is furthered, this change may be made if it is determined that the classes are similar enough to warrant this change.

All pictures used in this document can be found within the same folder as this document.