Binary Tree World

CSIS-3475-001

Group Members:

Navdeep Singh Chattha

Weihua Wang

Sandy Ngan

Viresh Soedhwa

Program description

The Binary Tree World program is a database of numbers structured in a binary tree.

The system consist of a Client and a Server program. The Screenshot below is the Client program.

The Server program does not have a user interface.

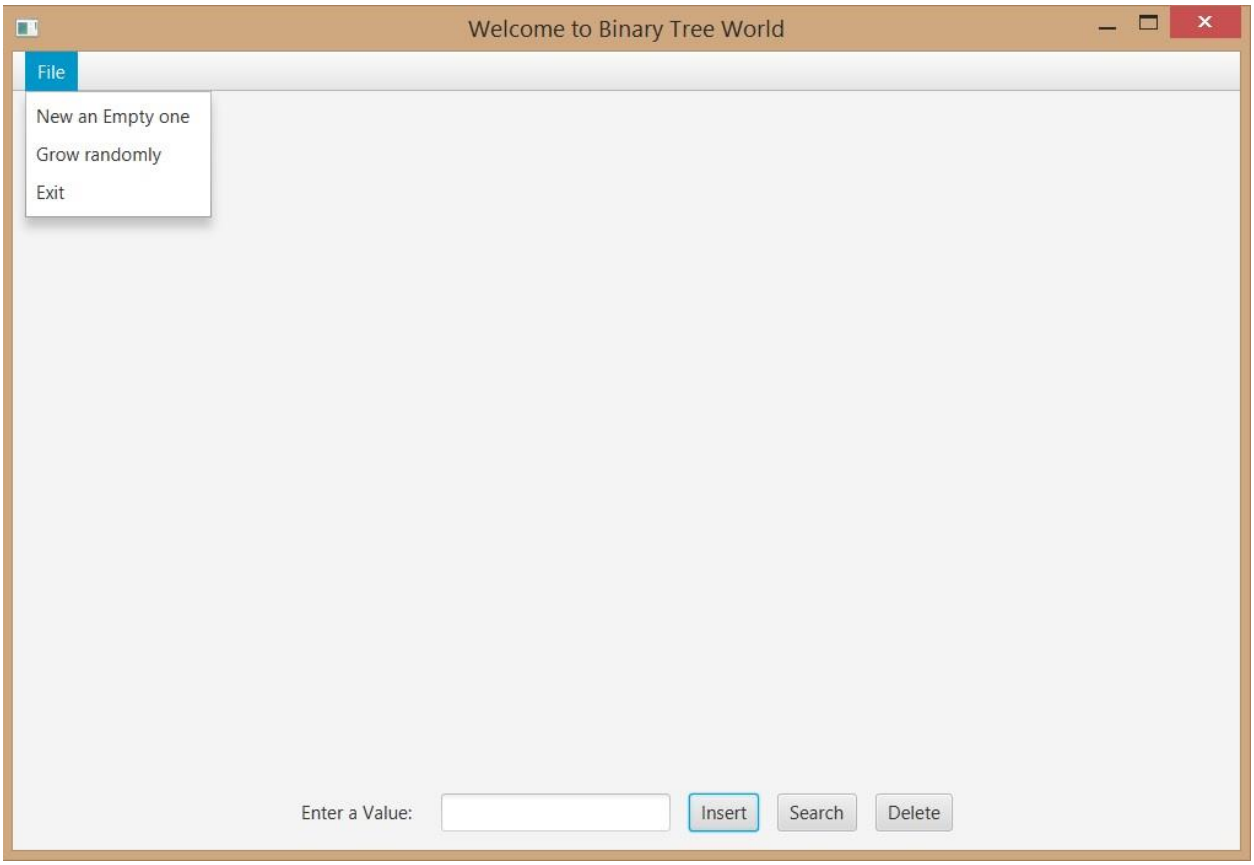
In the upper left corner there is a “File” menu item. Clicking on this will give the user three options:

“New an Empty one”, “Grow randomly” and “Exit”.

The “New an Empty one” clears the tree of all nodes. The user can then insert new elements in the tree.

The “Grow randomly” will generate several randomized numbers and insert them in the tree. The tree is not cleared first so these numbers will be appended to the numbers that may be already present in the tree.

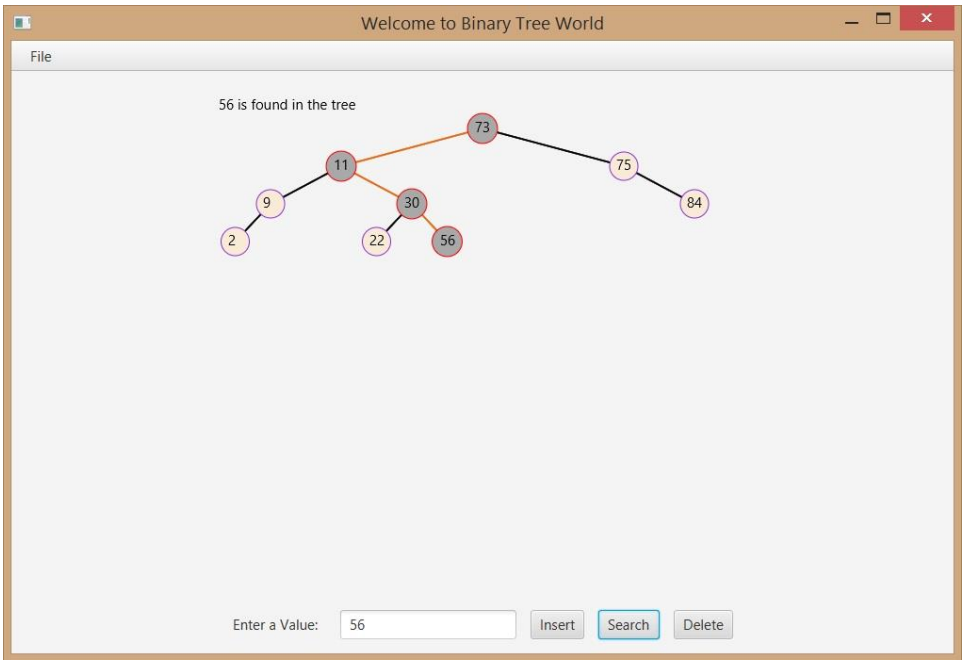
“Exit” exits the program.



On the bottom of the window there is an Input space with the label: “Enter a Value”. The user can enter a numeric value and issue the command Insert, Search or Delete.

The Insert command will insert the value in the tree. The tree will be balanced after the element is inserted.

The Search command will highlight the path to the value if it is found in the tree and the Nodes leading to this value will blink.



➢ *The path leading to the node that is found.*

The delete Command removes the value from the tree.

Running the program

Before running the program the user needs to start the server program. The Main function is found in the WServer class.

Once the server is started there will be a message in the console that says: “Server ready”

Please note that the client should be running on the same computer as the Server.

If the User wishes to run the Client on the different computer then the IP-addres of hostname should be changed accordingly. This can be done in the Wclient class of the client program. Simply change “localhost” to the desired IP or hostname. See the Screenshot below.

```
1  package Clientmain;
2  import java.io.IOException;
8
9  public class Wclient {
10
11     public static final String Host = "localhost";
12     public static final int Port = 6789;
13
14     private ArrayList<Integer> list;
15
16     public Client client;
17
18     public static Node temproot;
19     public static boolean insertconfirmer;
20     public static boolean deleteconfirmer;
21     public static boolean searchconfirmer;
22     public static boolean ipconfirmer;
23
24     public Wclient()
25     {
26
27         if (list == null) {
28             list = new ArrayList<Integer>();
29         }
30     }
```

```
Console
Wserver (2) [Java Application] C:\Program Files (x86)\Java\jre1.8.0_25\bin\javaw.exe (Aug 2, 2015, 1:30:47 AM)
Server ready
[SERVER] client trying to connect/127.0.0.1:62170
[SERVER] client trying to connect/127.0.0.1:62174
[SERVER] client trying to connect/127.0.0.1:62178
```

Please note the Screenshot above.

Once the client is started there will be a message in the Server console that states that the client is trying to connect.

The client window should pop open at the point and the user can begin working with the program.

To test the client functionality the user can open multiple client programs and notice that all changes made to one client will reflect to all clients.

The Search function is isolated to the client where the search command is issued. The path will be shown on that client only.

For a graceful exit close the client using File>Exit. Using the other close can cause the application to hang.

For you convenience there is a compiled executable “clientexecutable.exe”. This is configured to run on localhost.