Project 3 Test Report

Chloe Ingram **23 April 2022** **CMSC 250 – 7380**

UML Diagram:

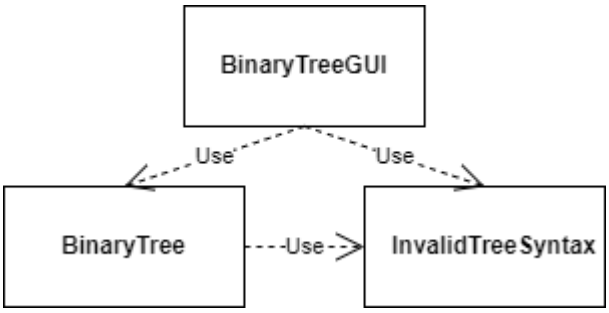

Figure 1. UML Diagram for BinaryTree project

Test Plan:

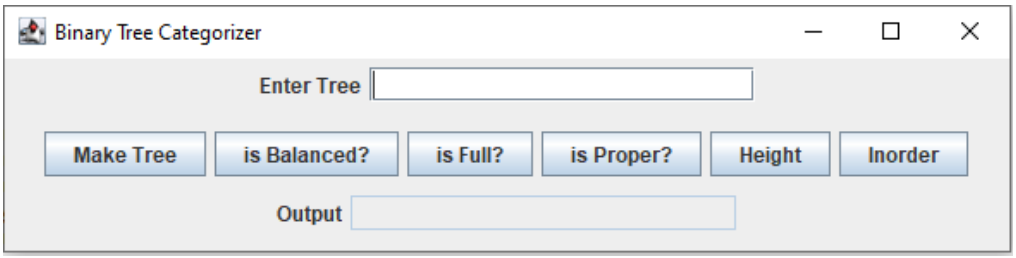

Figure 2. Main User Interface

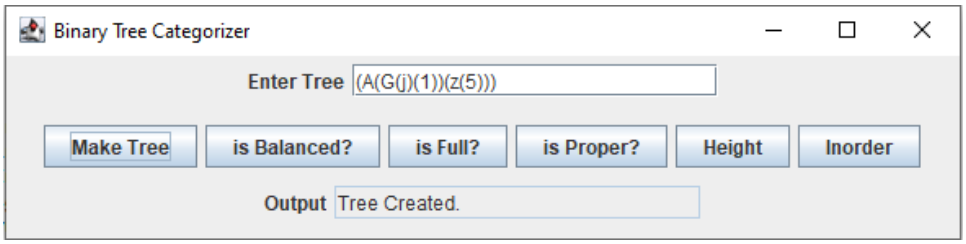

Figure 3. Binary Tree Successfully Created

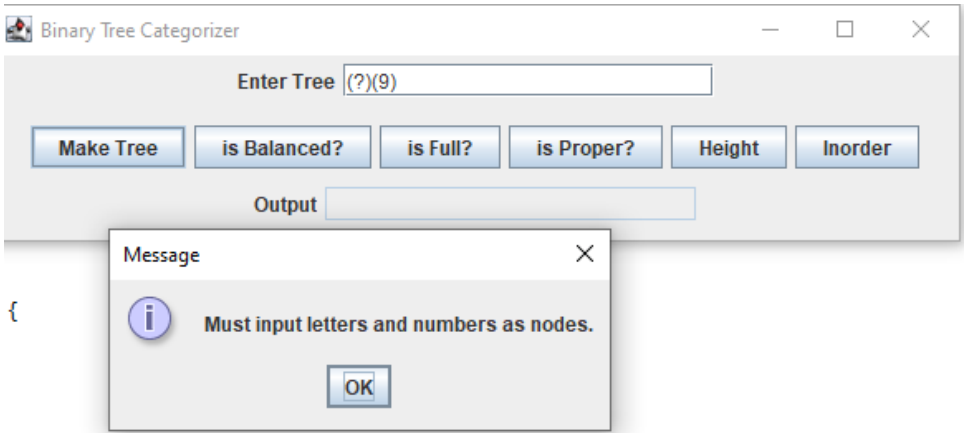

Figure 3. makeNode Exception 1

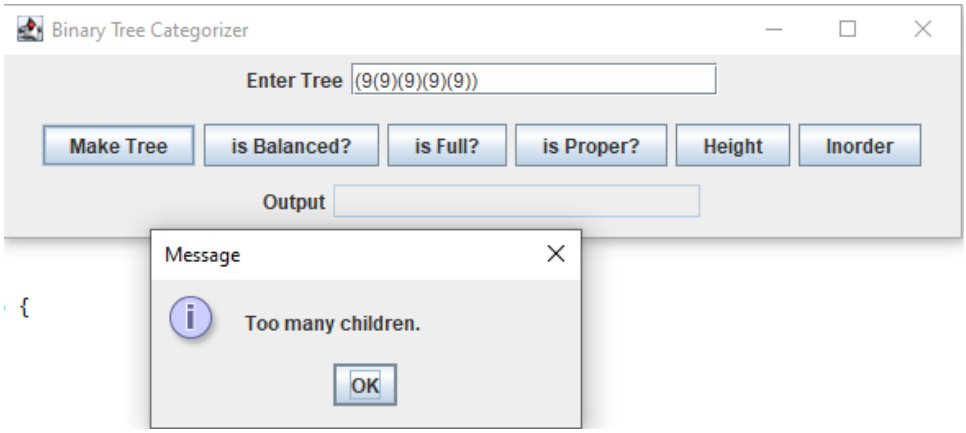

Figure 4. makeNode Exception 2

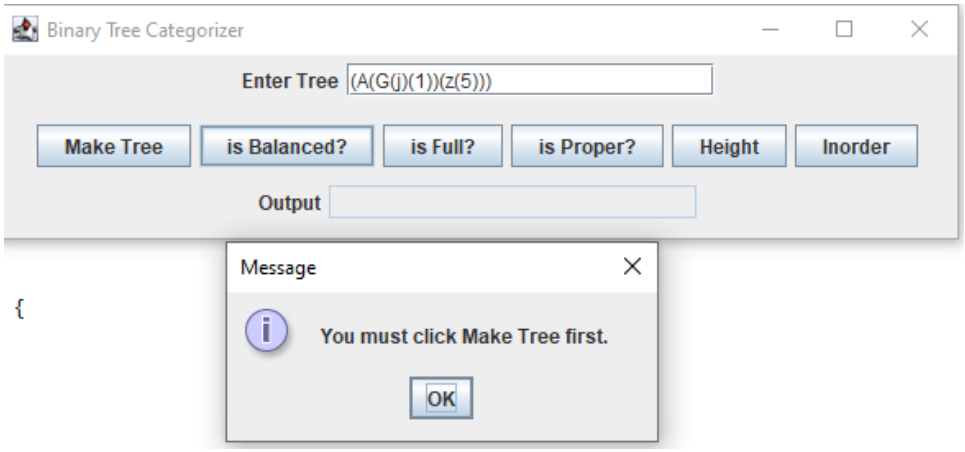

Figure 5. Categorize before Instantiation

Exception Handling				
Test Case No.	**Inputs**	**What is it testing?**	**Output**	**Pass/Fail**
1a	**(A(G(j)(1))(z(5)))**	**Correct tree syntax handling**	**Output box indicates tree successfully created**	**Pass**
1b	**(?)(9)**	**Alphanumeric only**	**Error indicating numbers and letters only**	**Pass**
1c	**(9(9)(9)(9)(9))**	**Too many children**	**Error indicating too many children**	**Pass**
1d	***presses the exit "X" button***	**Default Close Operation**	**Code terminates**	**Pass**
1e**	**(A(G(j)(1))(z(5))) *pressed "is Balanced? Button***	**Order of button presses/must make the tree first**	**Error indicating make the tree first**	**Pass**
****This was tested with all buttons with the same output. For brevity, this is included as a note.**				

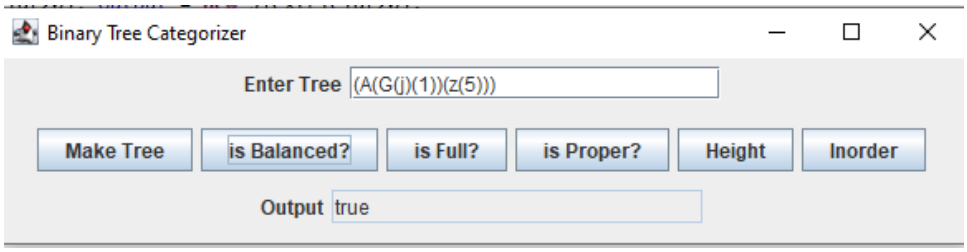

Figure 6. isBalanced Test 1

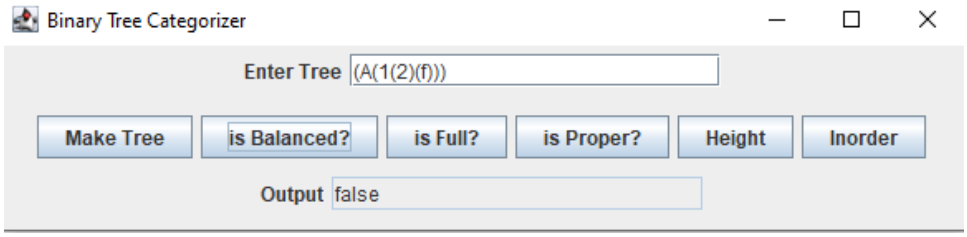

Figure 7. isBalanced Test 2

Balanced Tests				
Test Case No.	**Inputs**	**What is it testing?**	**Output**	**Pass/Fail**
2a	**(A(G(j)(1))(z(5)))**	**IsBalanced = true**	**true**	**Pass**
2b	**(A(1(2)(f)))**	**IsBalanced = false**	**false**	**Pass**

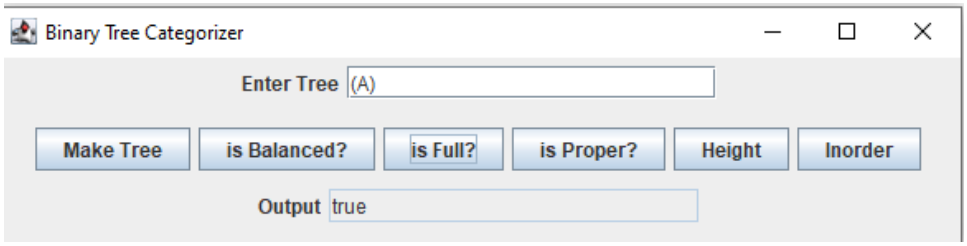

Figure 8. isFull Test 1

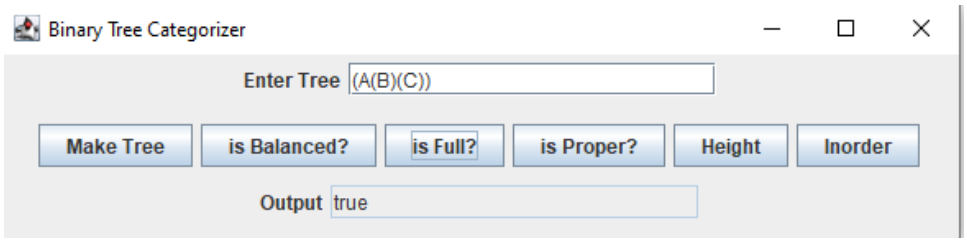

Figure 9. isFull Test 2

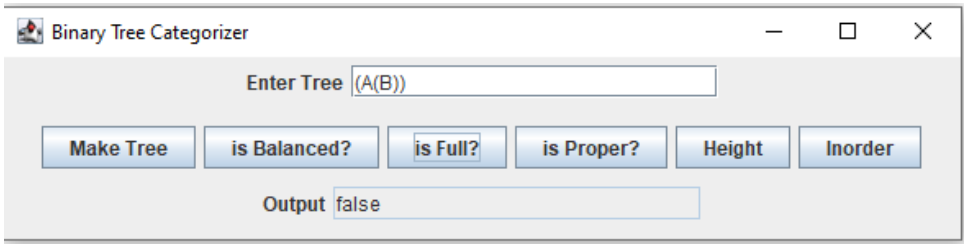

Figure 10. isFull Test 3

Full Tests				
Test Case No.	**Inputs**	**What is it testing?**	**Output**	**Pass/Fail**
3a	**(A)**	**Full is true with one node**	**True**	**Pass**
3b	**(A(B)(C))**	**Full tree is true**	**True**	**Pass**
3c	**(A(B))**	**Tree with height of two but isn't full should be false**	**False**	**Pass**

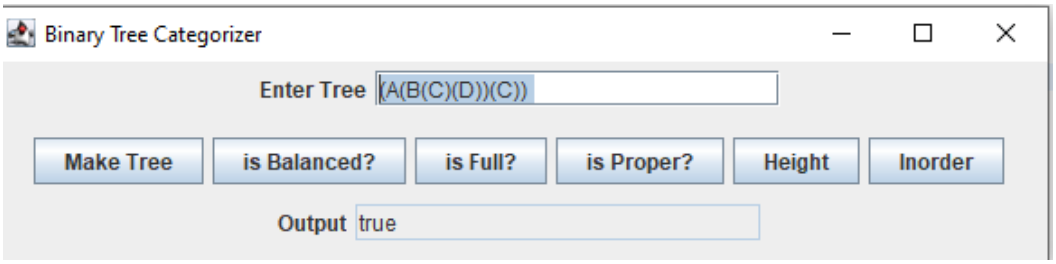

Figure 10. IsProper Test 1

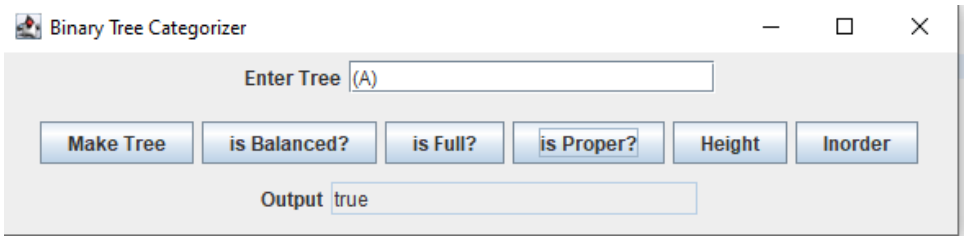

Figure 11. IsProper Test 2

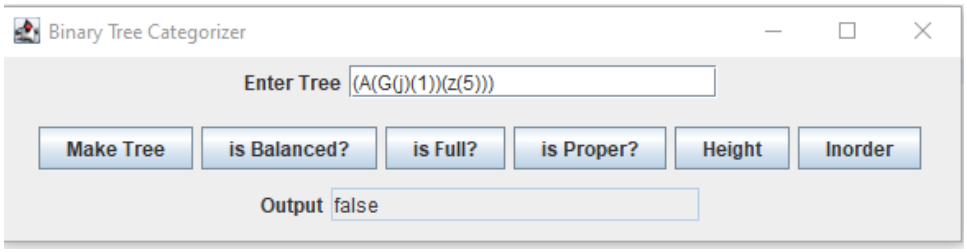

Figure 12. IsProper Test 3

Proper Tests				
Test Case No.	**Inputs**	**What is it testing?**	**Output**	**Pass/Fail**
4a	**(A)**	**Proper is true with one node**	**True**	**Pass**
4b	**(A(B(C)(D))(C))**	**Proper tree is true**	**True**	**Pass**
4c	**(A(G(j)(1))(z(5)))**	**Tree from project guide indicates this should be false**	**False**	**Pass**

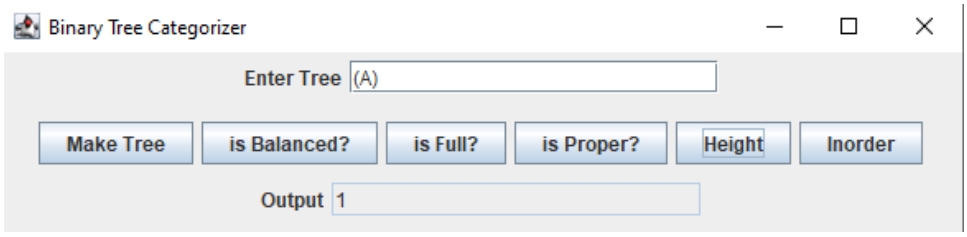

Figure 13. Height Test 1

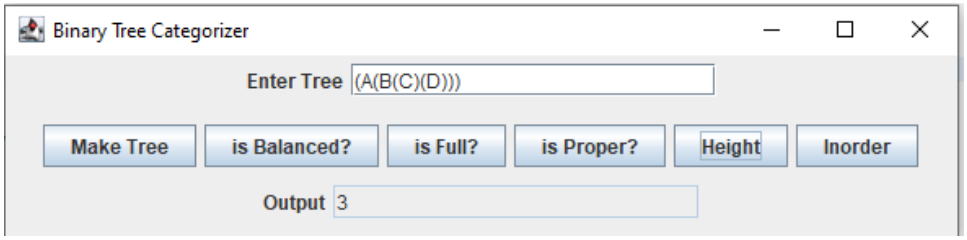

Figure 14. Height Test 2

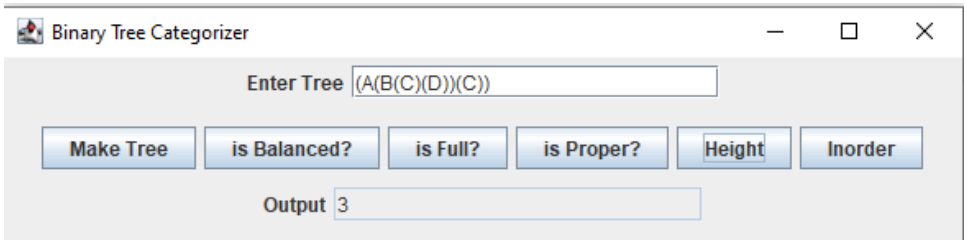

Figure 14. Height Test 3

Height Tests				
Test Case No.	**Inputs**	**What is it testing?**	**Output**	**Pass/Fail**
4a	**(A)**	**Height of one**	**1**	**Pass**
4b	**(A(B(C)(D))(C))**	**Longer subtree determines the height**	**3**	**Pass**
4c	**(A(B(C)(D)))**	**Different number of nodes but same height as above**	**3**	**Pass**

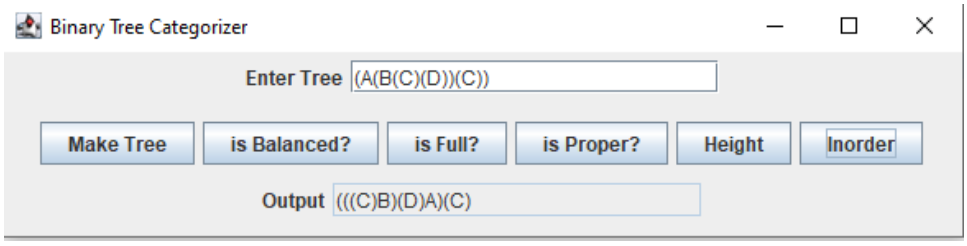

Figure 15. Inorder Test 1

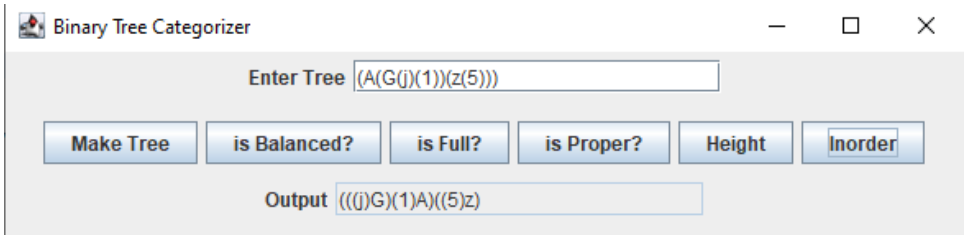

Figure 16. Inorder Test 2

Inorder Tests

Test Case No.	Inputs	What is it testing?	Output	Pass/Fail
5a	(A(B(C)(D))(C))	Inorder traversal	C B D A C	Pass
5b	(A(G(j)(1))(z(5)))	Validating my code against the answer found in the project requirements document	J G 1 A 5 z	Pass

Lessons Learned:

This was a huge help for me to understand recursion. I think it's pushed my coding much farther forward. Additionally, this was a good exercise in keeping code minimal, and trying not to re-do the same things over and over. I felt more creative. I feel more confident with GUIs as well.