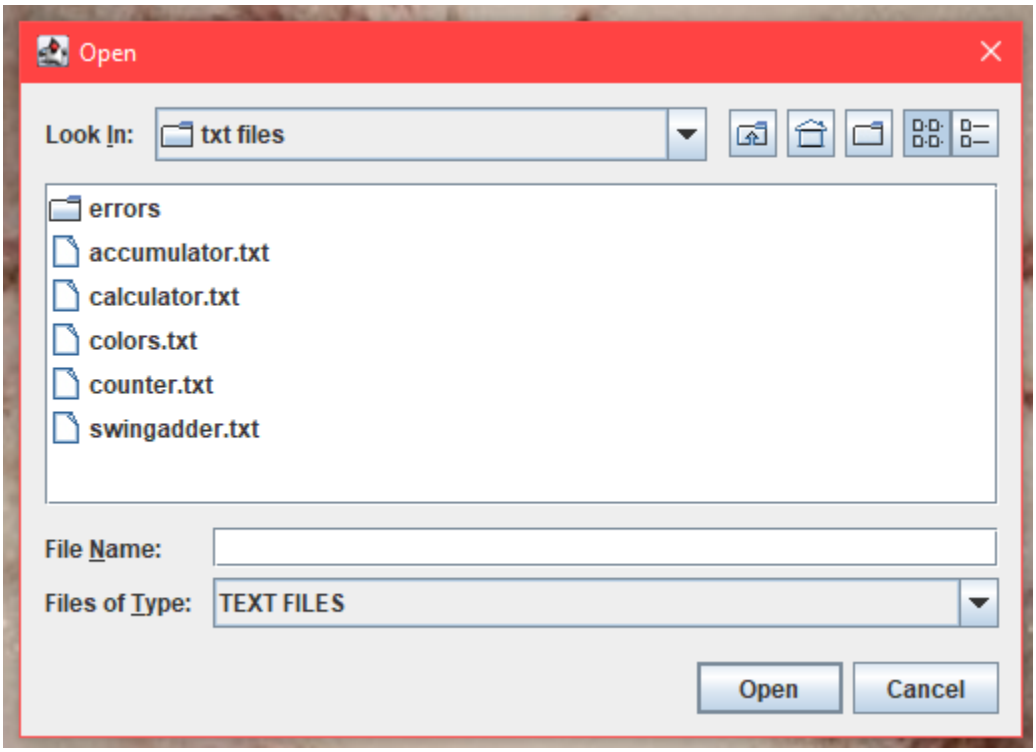
Kyra Samuel
CMSC330
Project 1

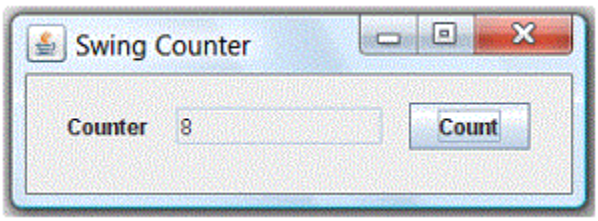
Tests 1-3 are practice GUIs found on [this website], using swing: Counter, Accumulator, Swing Adder. Please view the log for info and errors. All the txt files can be found in the “txt files” folder.



Inside /txt files

Tests

Test 1. counter.txt
Example:



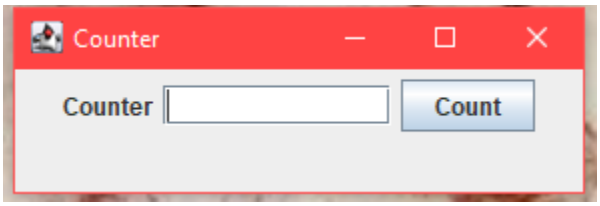
Chapter 4: Swing GUI Applications Example 4.1

Input:

```
counter.txt
1   Window "Counter" (300, 100) Layout Flow:
2   Label "Counter";
3   Textfield 10;
4   Button "Count";
5   End.
```

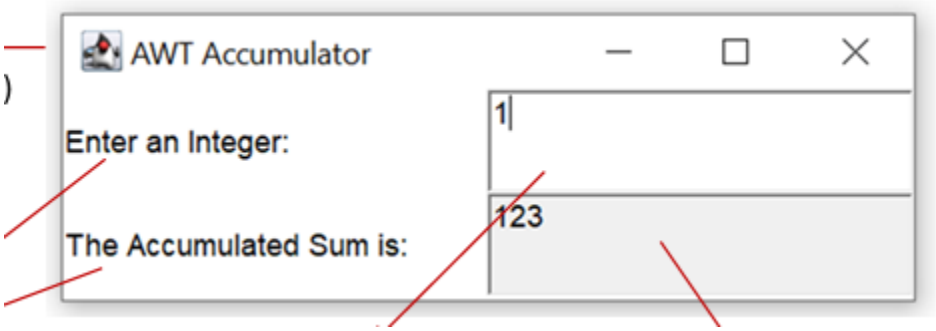
counter.txt

Output:



The only downfall is that my text fields are always editable and there are no action implementers.

Test 2. accumulator.txt



Chapter 1: AWT GUI Applications Example 1.2

(I am still using Swing)

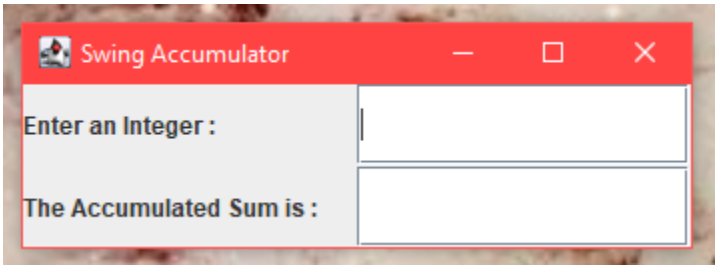
Input

accumulator.txt

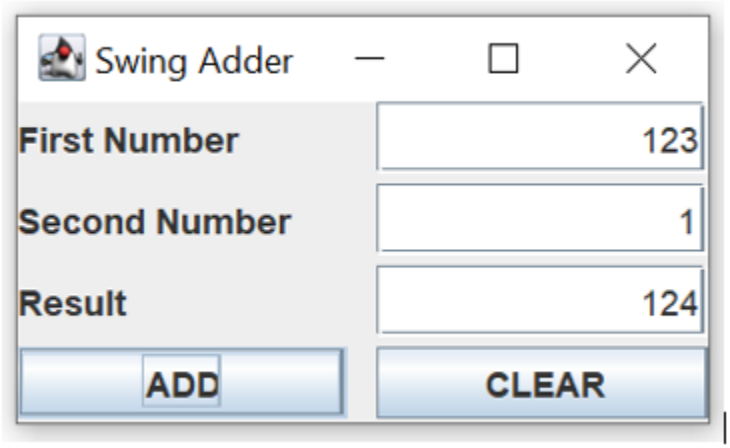
```
Window "Swing Accumulator" (350, 120) Layout Grid (2, 2, 1, 1):
Label "Enter an Integer: ";
Textfield 10;
Label "The Accumulated Sum is: ";
Textfield 10;
End.
```

accumulator.txt

Output



Test 3. swingadder.txt



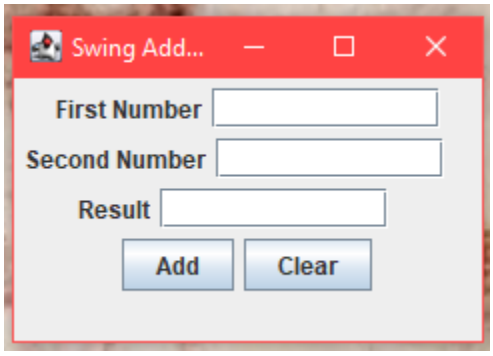
Input:

swingadder.txt

```
1   Window "Swing Adder" (250, 170) Layout Grid(1, 1, 3, 3):
2   Panel Layout Grid(4, 1, 10, 3):
3   Panel Layout Flow:
4   Label "First Number";
5   Textfield 10;
6   End;
7   Panel Layout Flow:
8   Label "Second Number";
9   Textfield 10;
10  End;
11  Panel Layout Flow:
12  Label "Result";
13  Textfield 10;
14  End;
15  Panel Layout Flow:
16  Button "Add";
17  Button "Clear";
18  End;
19  End;
20  End.
```

swingadder.txt

Output:



This is a bit different from the example above because I am using nested panels with Flow Layout.

Test 4. colors.txt

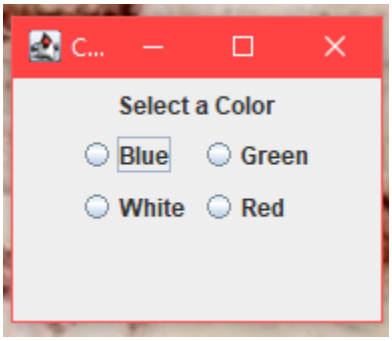
Input:

colors.txt

```
1   Window "Colors" (200, 160) Layout Flow:
2   Label "Select a Color";
3   Panel Layout Grid(2, 2, 2, 2):
4   Group
5   Radio "Blue";
6   Radio "Green";
7   Radio "White";
8   Radio "Red";
9   End;
10  End;
11  End.
12
```

color.txt

Output:



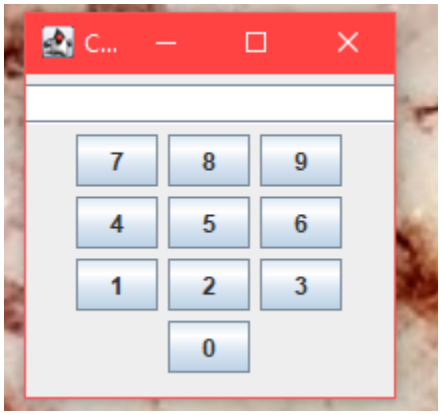
Test 5. calculator.txt

Input:

calculator.txt

```
Window "Calculator" (200, 200) Layout Flow:
 Textfield 20;
 Panel Layout Grid(4, 3, 5, 5):
 Button "7";
 Button "8";
 Button "9";
 Button "4";
 Button "5";
 Button "6";
 Button "1";
 Button "2";
 Button "3";
 Label "";
 Button "0";
 End;
End.
```

Output:

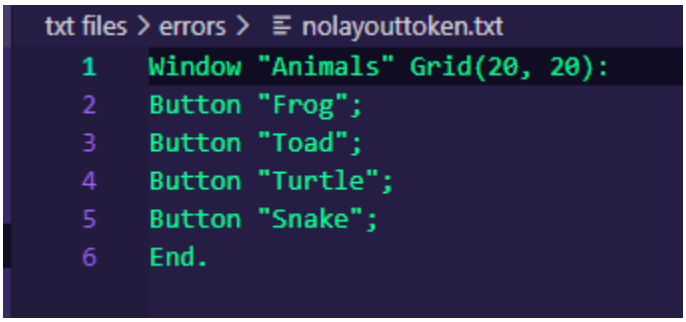


Error Checking

All txt files that include errors can be found in the /txt files/errors folder.

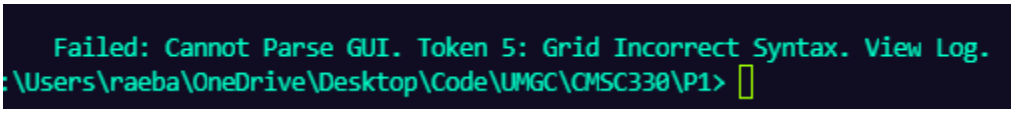
Error 1. nolayouttoken.txt

Input



nolayouttoken.txt

Output

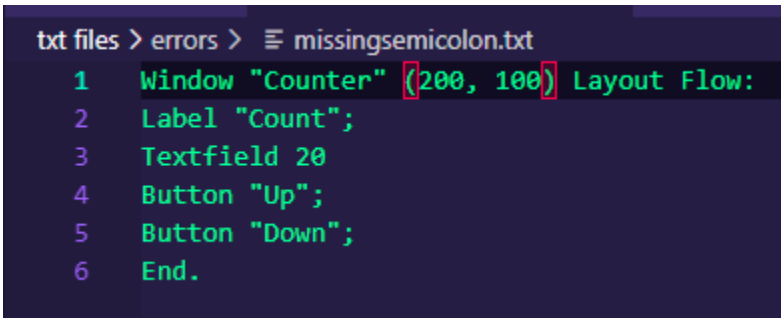


This lets users know that something is missing before token 5, where the word “Grid” is placed. In this case, the file has numerous errors, but the first error would be that there is not a size, the terminal “Layout”. Grid is also missing two parameters as well, so we must add those:

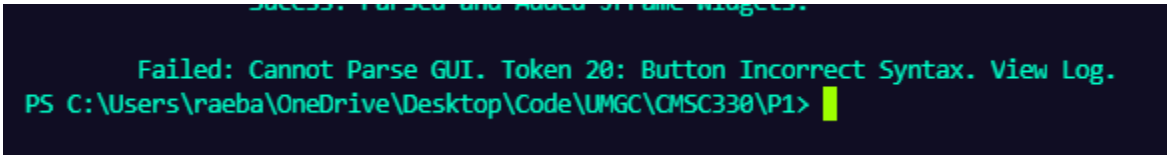
Window “Animals” (300, 200) Layout Grid(2, 2, 20, 20):

Error 2. missingsemicolon.txt

Input



Output

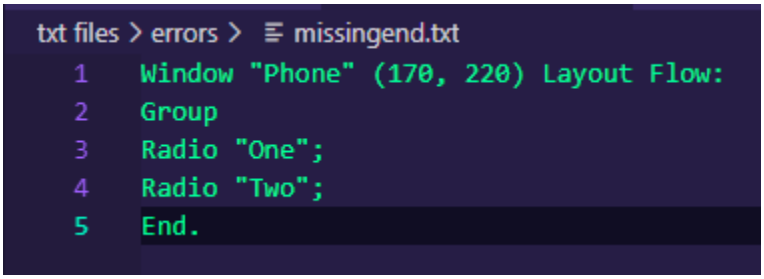


This lets users know that something is missing before token 20, where the word “Button” is placed. In this case, it would be a semicolon for the textfield widget:

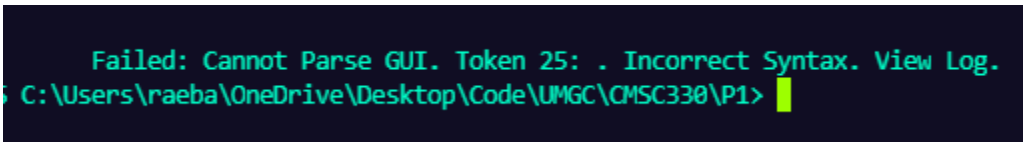
Textfield 20;

Error 3. missingend.txt

Input



Output



This lets users know that something is missing before token 25, where the character "." is placed. Panel and Group both require "End;" statements, we can fix the issue by adding "End;" on line 5 before the "End.":

Recap

I've learned how to read/write grammars and apply that to programs. I never realized that programming languages were parsed using numerous if statements. I am trying to apply this method to creating different applications using my own grammar. I want to create a drag and drop panel that outputs a txt file which would then be parsed into something! I've learned about different parsing methods as well and the difference between them. I feel as if I can understand syntax errors more and why these things occur.