**Triangle Solutions Design Document**

**Tuesday 11/14/17**

First talked to myself about coding the triangle solution thing from scratch in Java.

**Tuesday 11/28/17**

Did some work on the java Triangle bit. I have the data types captured.

**Friday 12/01/17**

I’ve managed to screw up the triangle solution to the point where the sides are now not getting assigned correctly.

I think I’m getting the sides in the correct position internally, but messing up the assignments.

The internal angle assignments aren’t right, but then I’m messing up the outputs, which is making them correct.

**Sunday 12/03/17**

Triangle is now copying to the correct values, but the angles are still in the wrong places. I think the point is that the parameters on the lawOfCosines() thing is messed up.

When I went through it again carefully, and carefully looking at the example solution, it all comes out right.

<http://www.calculator.net/triangle-calculator.html?vc=90&vx=4&vy=3&va=&vz=&vb=&angleunits=d&x=49&y=24>

**Tuesday 01/23/18**

Made some progress on Triangle. I still need to add validCount and an accessor to Sides and pull the for loops out of findSolution(). Done.

**Wednesday 01/24/18**

* See if you can get the Triangle thing to a state where you can send it out.

**Wednesday 03/14/18**

Started more work on Triangle…

* Presently, I have getNext/Prev ID stuff in Triangle rather than as parts of Sides and Angles. It would conceptually make more sense for these to go there. Even better…they should be in the base class of Sides and Angles.
* findSolution() needs for loops to find included sides or included angles.
* What would the base class name be?  
  They are triangle data sets.  
  TriangleData
* What would the common functionality be?
* It would then make more sense to identify angles/sides as a, b, c (or even 0, 1, 2) through the base class
* What maintains the idea of what’s next to what?  
  Uppercase are angles  
  Lowercase are sides  
  (...c) A b C a B c (A...)

**Saturday 04/28/18**

Finally got some work done on Triangle. Wrote the TriangleData base class.

This is where source control would come in handy.

At the level of Triangle.java, there’s nothing to say whether an item ID is for a side or an angle.

**Sunday 04/29/18**

In Triangle.java, I have getNext/PrevSide/AngleID. Since the ID’s are no longer side or angle dependent, they can go in TriangleData.

Maybe it’s not that simple. These functions take a side and return an angle and vs versa.

I don’t know where the output is going during debug.

**Monday 04/30/18**

There’s separate sub-tabs under Output in the NetBeans IDE where the printf stuff is going.

The triangle solution thing appears to have survived the class/sub-class conversion.

Triangle to do…

* findSolution() needs for loops to find included sides or included angles.
* There’s an issue in ASA() and AAS() where the B side is wrong.

**Thursday 05/03/18**

Triangle to do…

* findSolution() needs for loops to find included sides or included angles.
* There’s an issue in ASA() and AAS() where the B side is wrong.  
  It appears that it may have *always* been wrong.  
  done

**Tuesday 05/08/18**

Started complaining about the need for a separate design doc versus burying the details in my job search.

**Wednesday 05/09/18**

Created the Triangle Solutions Design doc.

**Tuesday 05/22/18**

Finally got around to adding data search functions.

It would appear that I still have confusion on traversal direction in getNext/PrevDataID.

The stuff still seems to work.

To do…

* Implement findDataASA()  
  done
* Implement findDataAAS()  
  done
* Implement findDataSAS()  
  done
* Finish SAS() call sequence in findSolution()  
  done
* Call findSolution() instead of hard coded stuff in setFromCmdLine()  
  done with useFindSolution flag.

I’m still confused by interface consistency on ASA and SAS…which should specify the included Side or Angle.

**Wednesday 05/23/18**

Finished the list I made yesterday. Now I need to test it all.

It works on SSS. I need to change the command line data to do the other stuff.

**Thursday 05/24/18**

To do…

* Test findSolution()
* Come up with a way to run a bunch of different data sets instead of just loading stuff from the command line.
* There was a thought to read the data from an XML formatted file, since Java has a built in parser.
* Get started on a GUI

Rearranged the code to move crap out of the command line capture function that is actually unrelated test code.

**Thursday 05/31/18**

Added setAll() and made hacky calls to it inside main().

AAS() is having issues. It doesn’t think there are two angles.

Only one angle and one side provided. No solution possible.

That implies that it’s in the SAS() section…which means that something in AAS() failed.

**Friday 06/01/18**

The site I’ve been using as a reference seems to have some confusion as to the direction the traverse is made. The result graphic goes the opposite direction from the data entry graphic.

http://www.calculator.net/triangle-calculator.html?vc=&vx=&vy=31&va=37&vz=&vb=54&angleunits=d&x=72&y=18

In findDataAAS() it seems the problem is in getting the ID for the correct side in the if statement. If I have angles A & B, I should look for side B (or A).

I guess what I need is the side opposite the first angle rather than the next from the second angle.

Now it appears that we *don’t* have all three angles. That’s because I assigned the angle result in AAS() incorrectly.

Now it kinda appears to work.

To do…

* There was a thought to read the data from an XML formatted file, since Java has a built in parser.
* Do we want to get the area of the triangle while we’re at it?
* Get started on a GUI

**Sunday 06/10/18**

Got started on some GUI stuff using a tutorial site…

<https://www.codeproject.com/Articles/33536/An-Introduction-to-Java-GUI-Programming>

The code is now in an incomplete and confused state until I figure out what all the panes and panels are supposed to do. I think panes and panels are interchangeable.

**Saturday 06/23/18**

Yeah…where was I?

Now I’m showing a window that doesn’t contain anything.

The example that worked had the JButton added to the (base)JPanel that was added to the Container.

Now I have a calcPanel. Adding it to the basePanel doesn’t help.

Set the layout, location and size of the calcPanel that the button goes on. Now something shows up…in a weird place.

The window is 486 x 493. I had specified 500 x 500. The button is 100 x 30 as specified.

The location is 236, 231 relative to the window. 236 x 200 relative to the client area. So, it seems to be located relative to the client area.

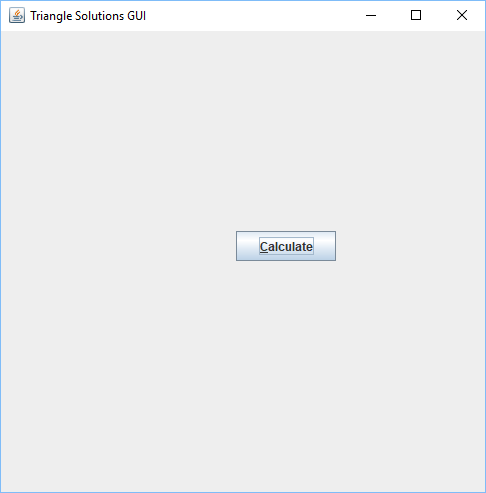
With a 300 x 300 window, the client area is 286 x 263. Total 286 x 292. Suggesting that the title bar is 29 or 30 pixels.

X is under by 14 pixels

Y is under by 7 plus titlebar pixels.

With the fixups the total window size is now 300 x 330, with a client area size of 300 x 300. The button is still in the wrong place. 136 x 101.

It may be that my math is just wrong.



**Sunday 06/24/18**

The math was “right”, but the parameter order was wrong. Now it looks like it should.

To do…

* Find out how to get the screen dimensions
* Find out what gokPixels and tbHeight really represent.
* Find out how to do output fields

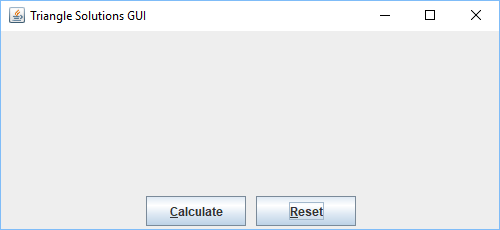
Ctrl-Shift-R turns on rectangular select.

There’s a pile of different things that are piling up in TriangleGUI that probably need to be in separate classes, like…

* calcButton
* dataPanel
* valueField
* valueFieldAngle
* valueFieldSide
* graphicPanel

Did a bunch of fancy stuff to make a button panel and add a reset button. Now I don’t show the right sized anything.

Got it to work…



The example code I’ve been working from doesn’t include input fields.

Got the input field stuff started, but I’m not seeing anything work yet.

**Thursday 06/28/18**

There was a bit where I wanted to return two different objects from a function (JFormattedTextField and JLabel), but Java doesn’t let you pass by reference and there’s no such thing as a pointer. What I need to do instead is create a structure(ish) thing.

**Friday 07/06/18**

I need to find an input field example that works.

To do…

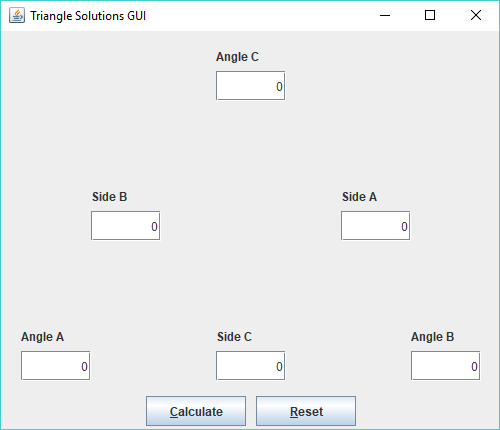
* Find out how to get the screen dimensions
* Find out what gokPixels and tbHeight really represent.
* Find out how to do input and output fields
* Find out how to format fields to only allow doubles
* Find out how to capture field values
* Find out how to populate field values
* Is there a way to handle the fields as either one array of 6 elements, or 2 arrays for sides and angles?

**Saturday 07/07/18**

Discovered that I hadn’t added the field and label to the dataPanel.

Now it’s showing up.

And now I have all 6 fields.



I think I need PropertyChangeListener

**Sunday 07/08/18**

I’m seeing *some* example code, but nothing yet that I think is complete and explicit enough to get all the objects and interactions down.

There are form examples that I probably put in C:\Development\Java, but it doesn’t seem that the forms try to do anything with the data that goes in the fields.

See what’s around 11/30/17. My notes say I installed the NetBeans IDE, but not anything about the examples.

C:\Program Files\NetBeans 8.2

It would appear that it came from the NetBeans IDE.

Got an OpenGL thing that I don’t need today…

C:\Downloads\Development\Java\1246294100841\_netbeans-opengl-pack\_0.5.5.zip

I’m looking at…

<https://netbeans.org/kb/docs/java/gui-functionality.html>

It appears that I might not *need* a listener after all. I might be able to harvest values from the fields at the point that the Calculate button is handled.

I’m still thinking that I need a class for the fields with…

* xPos
* yPos
* Name string

I was thinking that I also needed to hold the listener stuff, but that might not be necessary.

I’ve got something that makes the clear button work.

I’m thinking I need to host the Triangle object in the UI code…maybe. Or maybe I need to pass it to the UI’s constructor.

I think I’ve got something that pulls from the form. Now I need to repopulate.

It seems to properly pull stuff from the fields, get the solution and repopulate the fields. I think the problem now is that I need to format the angle data to ignore tiny floating point drift.

**Monday 07/09/18**

Started poking at Git

C:\Downloads\Development\Git\Git-2.18.0-64-bit.exe

<https://git-scm.com/book/en/v2>

Also git extensions. Should install first.

<https://www.youtube.com/watch?v=zMz9IZjUBFM>

It would appear that you don’t need to install Git itself.

GitExtensions-2.51.04.msi (Norton doesn’t like this.)

GitExtensions-2.51.03.msi (Norton doesn’t mind this one.)

Tried setting up an account on GitHub. You have to pay $7/month for private repos.

TriangleJava

Tried installing and running Git Extensions. It appears that you really *do* need to have Git installed.

Installed Git-2.18.0-64-bit.exe. It would appear that the git extensions version referenced in the install tutorial I found, also included git. The more current version apparently doesn’t.

Apparently KDiff3 wasn’t included either.

Back to triangle…

Added a rounding function roundDouble() which makes the results readable.

It fails to fail if the two shortest sides are shorter than the longest side. In that case lawOfCosines() returns nan. Can you even *test* for that? Double.isNaN()

To do…

* Is there a way to automatically highlight/select text when a field is clicked on or tabbed to?
* Create field class to simplify code.
* Find a way to use indexes or iterators on arrays to get through the fields and stuff to simplify the code.
* Find out how to get the screen dimensions
* Find out what gokPixels and tbHeight really represent.
* Find out how to get the window dimensions.
* Implement redraw/resize functions
* Implement drawing of original equilateral triangle and result triangle.