Instructions to run my code:

```
// compile
javac -cp :../lib/*: myGUI.java

// execute
java -cp :../lib/*: myGUI
```

Instructions to run my tests:

```
// compile all tests
javac -cp :../lib/*: *_Test.java

// run specific test
java -cp :../lib/*: org.junit.runner.JUnitCore SimplifiedTrackers_Test
```

How and why, I tested these components:

SimplifiedTrackersComponent

This component of my GUI displays, for each tracker, simplified GPS events by stripping their altitude information. To strip the altitude information from each GPS event, I created a function that converts a stream of GpsEvents to a stream of SimpleGpsEvents by mapping a GpsEvent object to a new SimpleGpsEvent object, created with that initial event.

Since we can check the SimpleGpsEvent class file itself to see that there is no altitude field, I decided that I only needed to test that the function correctly carries over the fields. I suppose it also checks that none of the fields were set to the altitude as well. Testing this function was done by creating a StreamSink that we could send an event to later, and the converted SimpleGpsEvent stream. I then set up cells for each of the desired fields, sent the event, and finally compared the event's information to the cell's sampled information.

Now that we can see that the information is correct, I wanted to test whether the information was being displayed in the correct output. To do this, I created an array of GpsEvent streams and a SimplifiedTrackersComponent object. I then sent down two events in separate streams. I have written my program such that the trackers are in order and that the cells used in the display are stored, so when sending events down an array of two streams, there should be two groups of cells. By checking the first two group of cells and comparing their values to the event downstream, I was able to verify that the GpsEvent data were being outputted to the correct sub-displays. Viewing the display also shows that the trackers are in ascending order.