

---

# **EDC ASSIGNMENT 3**

## **TESTING**

---

# HOW TO COMPILE AND RUN THE CODE

COMMAND TO COMPILE CODE :

```
javac -classpath ".:sodium.jar:swidgets.jar" GpsGui.java
```

COMMAND TO RUN CODE:

```
java -classpath ".:sodium.jar:swidgets.jar" GpsGui
```



# OVERALL GUI LAYOUT:

GUI

Trackers with their respected latitudes and longitudes below:

Tracker0

latitude:39.980743

longitude:116.327624

Tracker1

latitude:39.980974

longitude:116.329169

Tracker2

latitude:39.977818

longitude:116.310606

Tracker3

latitude:39.966156

longitude:116.322635

Tracker4

latitude:7.883777

longitude:98.392853

Tracker5

latitude:40.0084016

longitude:116.3685883

Tracker6

latitude:39.86355

longitude:116.2598883

Tracker7

latitude:30.4783133

longitude:114.3446549

Tracker8

latitude:39.241411

longitude:113.563599

Tracker9

latitude:39.986488675

longitude:116.381003804

Last Tracker Event time:

Tracker0, latitude:39.977643, longitude:116.327703 07/11/2022, 07:36:29

Control Panel for Latitude and Longitude:

Latitude Lower Bound:

-90.000

-90.000

Latitude Upper Bound:

90.000

90.000

Longitude Lower Bound:

-180.000

-180.000

Longitudes Upper Bound:

180.000

180.000

SET RANGE

Filtered Event:

Tracker0

latitude:39.980743

longitude:116.327624

Tracker1

latitude:39.980974

longitude:116.329169

Tracker2

latitude:39.977818

longitude:116.310606

Tracker3

latitude:39.966156

longitude:116.322635

Tracker4

latitude:7.883777

longitude:98.392853

Tracker5

latitude:40.0084016

longitude:116.3685883

Tracker6

latitude:39.86355

longitude:116.2598883

Tracker7

latitude:30.4783133

longitude:114.3446549

Tracker8

latitude:39.241411

longitude:113.563599

Tracker9

latitude:39.986488675

longitude:116.381003804

Total distance travelled over the last 5 minutes by each tracker:

Total Distance travelled:

Tracker0

43m

Total Distance travelled:

Tracker1

43m

Total Distance travelled:

Tracker2

43m

Total Distance travelled:

Tracker3

43m

Total Distance travelled:

Tracker4

43m

Total Distance travelled:

Tracker5

43m

Total Distance travelled:

Tracker6

43m

Total Distance travelled:

Tracker7

43m

Total Distance travelled:

Tracker8

43m

Total Distance travelled:

Tracker9

43m

# TESTING WHILE COMPILING:

- 1.This section is dedicated to the testing that was done in order to be able to compile and run the application using sodium and swidgets. In order to be able to compile and run the code successfully first thing I did was to download swidget and sodium jars from my uni and then trying to compile the provided example.java file as well as the files provided in the week9 example folder. It took me a long time and research to be able to do this.
- 2.After getting the jars I tried compiling using the path of the sodium and swidget jar on my laptop. At first I added it to my main Library/Java/Extensions folder path and tried compiling it from my edc\_assignment folder but it did not work for me.
- 3.Therefore, we can call it a second test- after sometime of trying I decided to add the jars directly to my edc\_assignment folder and then compile but it still did not work.
- 4.Later I realised the the text for the path is very sensitive and looked online to find the correct path and was able to compile but with errors. The error received was regarding the version of the swidgets file so I looked up on how to update it.
5. Then after finding I it I was finally able to correctly compile my code and I tested it using different paths and got it to work successfully. The other paths I tried includes:

```
java -classpath ".:Library/Java/Extensions/sodium.jar:swidgets.jar" Example.java
```

```
javac -classpath ".:Users/archieverma/Desktop/edc_assignment3/sodium.jar:swidgets.jar" Example.java
```

```
javac -classpath ".:sodium.jar:swidgets.jar" Example.java
```

## RESULT AFTER COMPILING AND RUNNING EXAMPLE.JAVA :

```
archieverma@Archies-MacBook-Air edc_assignment3 % java -classpath " ./Users/archieverma/Desktop/edc_assignment3/sodium.jar:swidgets.jar" Example
Tracker9 | lat:39.984960026 lon:116.31073856 alt:489.42906496063
Tracker1 | lat:39.980998 lon:116.329279 alt:157.0
Tracker4 | lat:7.879058 lon:98.39548 alt:0.0
Tracker3 | lat:39.965723 lon:116.319246 alt:627.0
Tracker2 | lat:39.968415 lon:116.314171 alt:237.0
Tracker5 | lat:40.0138699 lon:116.3434599 alt:154.2
Tracker6 | lat:39.8593933 lon:116.2573716 alt:164.0
Tracker7 | lat:30.4775883 lon:114.3582583 alt:150.9
Tracker8 | lat:39.266822 lon:113.578671 alt:3427.0
Tracker0 | lat:39.977643 lon:116.327703 alt:492.0
Tracker5 | lat:40.0138716 lon:116.3435749 alt:154.2
Tracker8 | lat:39.266557 lon:113.579018 alt:3456.0
Tracker6 | lat:39.8593933 lon:116.2574083 alt:141.1
Tracker1 | lat:39.981074 lon:116.329211 alt:156.0
Tracker8 | lat:39.266614 lon:113.57891 alt:3441.0
Tracker7 | lat:30.4777516 lon:114.3582616 alt:147.6
Tracker5 | lat:40.0138799 lon:116.3436883 alt:154.2
Tracker2 | lat:39.968477 lon:116.31416 alt:237.0
Tracker5 | lat:40.0138816 lon:116.3438099 alt:154.2
Tracker1 | lat:39.981125 lon:116.32917 alt:155.0
Tracker7 | lat:30.4779166 lon:114.3582699 alt:147.6
Tracker5 | lat:40.0138849 lon:116.34393 alt:154.2
Tracker9 | lat:39.98489813 lon:116.311883625 alt:477.595242782152
Tracker4 | lat:7.879091 lon:98.395467 alt:3.0
Tracker2 | lat:39.968537 lon:116.314144 alt:237.0
Tracker5 | lat:40.0138833 lon:116.3440516 alt:154.2
Tracker1 | lat:39.981163 lon:116.329138 alt:153.0
Tracker7 | lat:30.4780833 lon:114.3582749 alt:144.4
```

# TESTING FOR PART 1:

## PART 1 DESCRIPTION :

[FROM THE ASSIGNMENT PAGE]

1. Ten tracker simplified displays. Simplified tracking data has its altitude data removed and this should be carried out with Sodium FRP operations. From this stream, display, in separate cells:
  - Tracker number
  - latitude
  - longitude

# TESTING FOR PART 1:

## PART 1 RESULTS :

GUI		
Tracker0	latitude:39.977643	longitude:116.327703
Tracker1	latitude:39.981175	longitude:116.329128
Tracker2	latitude:39.968605	longitude:116.314109
Tracker3	latitude:39.965723	longitude:116.319246
Tracker4	latitude:7.879091	longitude:98.395467
Tracker5	latitude:40.0138783	longitude:116.3444083
Tracker6	latitude:39.8593933	longitude:116.2574083
Tracker7	latitude:30.4782449	longitude:114.3582733
Tracker8	latitude:39.26651	longitude:113.579198
Tracker9	latitude:39.98489813	longitude:116.311883625

# TESTING FOR PART 1:

## PART 1 TESTING DESCRIPTION :

1. For this part I understand I was supposed to get data using map function or something else and print the trackers from the range 0-9 with their respected latitude and longitude but without the altitude which was getting printed on the console and I was supposed to print the with proper vertical alignment in cells.
2. For the first part, the first thing I did was to create a GUI Frame, to do that I research and looked at how people create GUI's and frames as well as the week 9 examples, tested them and the codes to see if it is working on my local and then finally created my own frame with the name GUI and added a size for it.
3. After testing example.java, I realised that I can used the provided code for Initialise the GPS Service and Attaching a handler method to each stream to be able print the trackers in my GUI and therefore I added it to my GpsGui.java



3. Further, for printing the tracker number 0-9, through research and reading I realised that I can use a cell string 'tNum' and map function to be able to get gps event ev, use split function to remove space and print according to what I wanna print or remove, further use an array to print elements from the string array in the GUI which I stored, as an example tracker was the first element therefore expected[0] and hold("") to make sure they all print at once without any hold. Did testing using slip function adding different things within it to see how it impacts the GUI printing, did the same with hold and other functions.

4. In sequence, I used label function in gui to add array string for tracker for printing using frame.add() functions and then added colour to the text, changed its font and aligned it using functions like : setPreferredSize(new Dimension(), Font ft = new Font("Serif", Font.BOLD, 16);, lA.setFont(ft); and lA.setForeground(Color.BLUE); and printed it as a flow layout.

5. I did the same for printing latitude and longitude and also changed their names in Gps Event file to print the full word.

I was able to do all this through reading about the different function and testing them with some other examples and then modifying yo be able to use with in my GUI.

# TESTING FOR PART 2:

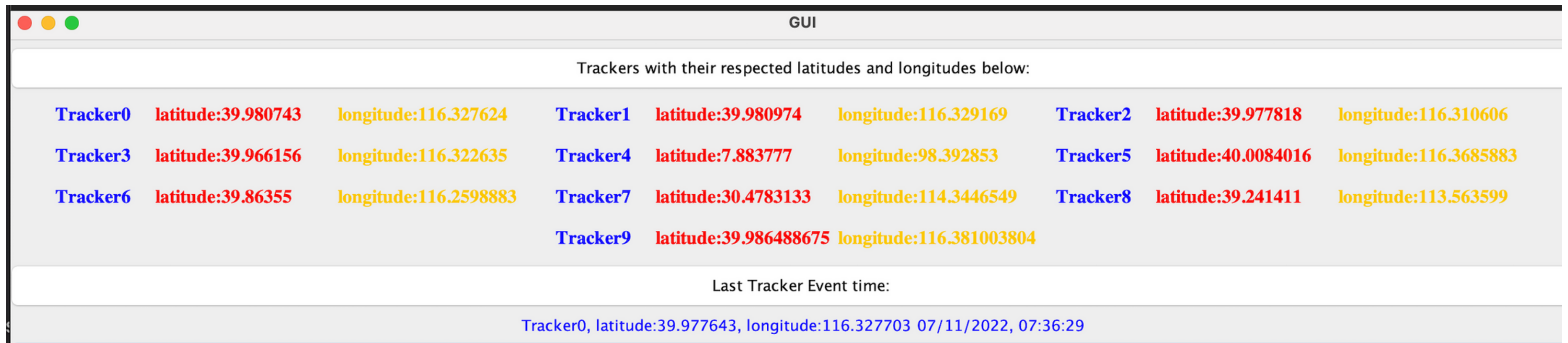
PART 2 DESCRIPTION :

[FROM THE ASSIGNMENT PAGE]

2. A display field shows each event as it is passed to the GUI, at the time it occurs.
  - This is presented as a comma-delimited string of 4 items
  - Only a single entry is presented at a time, and cleared after 3 seconds if not overwritten.

## TESTING FOR PART 2:

## PART 2 RESULTS :



# TESTING FOR PART 2:

## PART 2 TESTING DESCRIPTION :

1. For this part I understand I was supposed to find the recent tracker which has the most recent data(i.e., latitude and longitude and print with the time it got updated and update the whole thing every 3 seconds.
2. After finishing this part the next thing I did was set up a layout for the different parts making sure that all of the parts can get their own space and it is easy to tell which part is doing what on the GUI. In order to do that I had to test different things, firstly I tried JLabel to just highlight the headlines, then tested Grid Layout and then further I tested different buttons for the heading and settled on that.
3. Moving on I realised that I have to display the current time so I went and researched how to print date and time in a way I can print it as a comma-delimited string, I found some things which then I tested on my application and used to print the time for the second part according to the need of this

# TESTING FOR PART 3:

## PART 3 DESCRIPTION :

[FROM THE ASSIGNMENT PAGE]

3. A single (1) display that combines all input streams, and only outputs GPS events in the defined range of Latitude and Longitude.
  - The defined range is controlled by a control panel consisting of a latitude input field, a longitude input field, and a button to set the restriction.
    - The latitude and longitude restriction will probably require the use of the **snapshot** primitive.
  - Labels show the current setting of the restriction.
  - The GUI element displays the data in an identical form to the display field in (2), as long as the data is in the range.

# TESTING FOR PART 3:

## PART 3 RESULTS :

Control Panel for Latitude and Longitude:

Latitude Lower Bound:

-90.000

-90.000

Latitude Upper Bound:

90.000

90.000

Longitude Lower Bound:

-180.000

-180.000

Longitudes Upper Bound:

180.000

180.000

SET RANGE

Filtered Event:

Tracker0	latitude:39.980743	longitude:116.327624	Tracker1	latitude:39.980974	longitude:116.329169	Tracker2	latitude:39.977818	longitude:116.310606
Tracker3	latitude:39.966156	longitude:116.322635	Tracker4	latitude:7.883777	longitude:98.392853	Tracker5	latitude:40.0084016	longitude:116.3685883
Tracker6	latitude:39.86355	longitude:116.2598883	Tracker7	latitude:30.4783133	longitude:114.3446549	Tracker8	latitude:39.241411	longitude:113.563599
			Tracker9	latitude:39.986488675	longitude:116.381003804			

# TESTING FOR PART 3:

## PART 3 TESTING DESCRIPTION :

1. I understand that for this part I had to filter the tracker data in a way that it only shows data from the latitude or longitude max or minimum range typed in the textbox and whatever the user types in the text box should get printed on the gui next to the text box, further as the user presses the button to set range it should filter the data according to the typed range of longitude and latitude and give the result to the user under the filter trackers.
2. For this part I created textboxes for minimum or low bound latitude, maximum or high bound latitude, minimum or low bound longitude and maximum or high bound longitude. I did this by testing some online examples on how to create a text box on a GUI for strings and print it for the frame. Further, learned to create buttons, add labels, colours and align text through testing as well.
3. For the second part I used the similar code from the first part to print all the trackers under filtered events and aligned them.



# TESTING FOR PART 4:

## PART 4 DESCRIPTION :

### [FROM THE ASSIGNMENT PAGE]

4. A distance field for each tracker that outputs the total distance travelled over the last 5 minutes for each tracker.
  - Your GUI should display a label for each tracker, with the current distance travelled next to it.
  - This only includes events that are within the currently set Lat/Long range.
  - The value should be a distance in meters, rounded up to the nearest integer meter.
    - **Important:** The tracker altitude data is in feet. You should use `map` to transform this data.
  - Distance calculations may be calculated without considering the curvature of the Earth but must take the altitude into account.  
(See Latitude and Longitude section below)
  - **Hint:** You might want to look at the FRP primitive `snapshot`.
  - **Hint:** A sliding window of 5 minutes, however you implement it, will make this calculation easier. You only calculate distances between known positions, i.e. two distinct events



# TESTING FOR PART 4:

## PART 4 RESULTS :

Total distance travelled over the last 5 minutes by each tracker:											
Total Distance travelled: Tracker0 43m			Total Distance travelled: Tracker1 43m			Total Distance travelled: Tracker2 43m			Total Distance travelled: Tracker3 43m		
Total Distance travelled: Tracker4 43m			Total Distance travelled: Tracker5 43m			Total Distance travelled: Tracker6 43m			Total Distance travelled: Tracker7 43m		
			Total Distance travelled: Tracker8 43m			Total Distance travelled: Tracker9 43m					

# TESTING FOR PART 4:

## PART 4 TESTING DESCRIPTION :

1. For this part, I understand I was supposed to print the distance which it travelled over the last 5 minutes only for filtered trackers according to the latitude and longitude range which has been set printing the value in meters rounded up to the nearest integer, map the altitude data to transform it from feet to meters taking altitude into account without earth's curvature using FRP primitive snapshot with a sliding window of 5 minutes for two known positions.
2. For this part I used the similar code from the first part to print all the trackers under total distance travelled and added a label for printing the distance.
3. All the overall layout has been developed through research and testing to be implemented in a way most suitable for the application.

---

**T H A N K   Y O U !**

---