

INTRO TO ANDROID

CSE 590 Ubiquitous Computing | Lecture 1 | Mar 29

Jon Froehlich • Liang He (TA)

SCHEDULE TODAY: 6:30-9:20

06:30-06:45: Ice breaker

06:45-07:45: Intro to UbiComp

07:45-07:55: Short break

07:55-08:05: This class

08:05-09:20: Intro to Android

Spring 2018

- Home
- Announcements
- Assignments
- Discussions
- Grades
- People
- Pages
- Files
- Syllabus**
- Outcomes
- Quizzes
- Modules
- Conferences
- Collaborations
- Chat
- Attendance
- UW Libraries
- Add 4.0 Grade Scale
- Panopto Recordings
- Settings

CSE P 590 A Sp 18: Special Topics In Computer Science

[Jump to Today](#) [Edit](#)

Course Overview:

The aim of this class will be to introduce you to ubiquitous computing. We will focus on how traditional topics of computing have evolved to support the vision of a connected, portable, and a human-centric computing environment. The course will include discussion into contribution of various fields, including human-computer interaction, embedded computing, computer vision, distributed systems, machine learning, and electrical engineering. The students will gain practical experience in developing sensing systems in different application domains, such as activity recognition, health sensing, gestural interaction, etc.

The course will be a combination of lectures, tutorials, class discussions, and demonstrations.

Acknowledgments: This course has been informed by [Professor Shwetak Patel's Spring 2015 offering](#) and [Professor Mayank Goel's Fall 2015 offering](#).

Who We Are and Contact Info:

- **Instructors:** Professor [Jon Froehlich](#) and [Liang He](#) (TA)
- **Email:** jonf@cs.uw.edu and lianghe@cs.washington.edu
- **Lectures:** Thurs, 6:30-9:20PM in Johnson Hall 175 ([map](#))
- **Communication:** We will communicate largely via Canvas but may also send mails via the course mailing list: csep590a_sp18@uw.edu
- **Office hours:** By appointment.

Course Materials:

You must have a **laptop** for use in and out of class capable of running [Android Studio](#) and an Android emulator. We will be providing you with an Android device (a Huawei Honor7 running Android OS 7 aka Nougat), an Arduino board, and an electronics kit. All materials must be returned on the final day of class (June 7).

We will not be using a textbook; however, if you're interested, please see John Krumm's *Ubiquitous Computing Fundamentals*.

Assignments:

Course Status

[Unpublish](#) [Published](#)

[Import from Commons](#)

[Choose Home Page](#)

[View Course Stream](#)

[Course Setup Checklist](#)

[New Announcement](#)

[Student View](#)

[View Course Analytics](#)

March 2018						
25	26	27	28	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31
1	2	3	4	5	6	7

Course assignments are not weighted.

Spring 2018

Home

Announcements

Assignments

Discussions

Grades

People

Pages

Files

Syllabus

Outcomes

Quizzes

Modules

Conferences

Collaborations

Chat

Attendance

UW Libraries

Add 4.0 Grade
Scale

Panopto
Recordings

Settings

A1: Step Tracker

Published

Edit

⋮

Related Items

SpeedGrader™

Goal

In this assignment, you will build your own step tracker using your Android smartphone's built-in accelerometer (and gyro, optional) along with basic signal processing algorithms and heuristics. You should **not** use machine learning for this assignment (e.g., supervised learning).

There will be a midpoint check-in on April 5 ([link](#)). You will demonstrate a working version of your prototype to the TA, Liang He, during classtime on April 12.

Learning Goals

- Introduce and learn the basics of Android development (e.g., Android Studio, using the emulator, deploying to a device, debugging, the Android architecture).
- Introduce and learn basic methods for accessing and processing built-in sensors on Android.
- Introduce and learn how to apply basic signal processing algorithms in real-time.
- Reflect on and apply basic theories/design principles of persuasive technology (e.g., see [Consolvo](#), [et al.](#), [CHI'06](#), [Campbell](#), [et al.](#), [CSCW'08](#)).

Parts

- [6 pts] Design an **algorithm to robustly track steps** using the Android smartphone's built-in accelerometer (and, possibly gyroscope). At a minimum, your signal processing approach should include a *smoothing filter* and a *peak detection* algorithm. The algorithm should recognize a new step with ≤ 2 sec latency.
- [3 pts] Design a **debug visualization interfaces** that shows a line graph of the real-time accelerometer signal, the smoothed signal, and debug information about your tracker algorithm. This is part of your midpoint check-in due next week, see: <https://canvas.uw.edu/courses/1199409/assignments/4187237>. You can roll-your-own simple line graph or use an existing library (like [GraphView](#)).
- [2 pts] The **debug interface should also show**: (i) the number of steps your algorithm has tracked; (ii) the number of steps tracked by Android's [Step Detector Sensor](#); (iii) and the number of steps tracked by Android's [Step Counter Sensor](#).
- [3 pts] Design a **creative feedback interface** that highlights both when a new step is sensed as well as provides a cumulative count. For example, perhaps you make a fractal tree that dynamically grows based on the number

Spring 2018

Home

Announcements

Assignments

Discussions

Grades

People

Pages

Files

Syllabus

Outcomes

Quizzes

Modules

Conferences

Collaborations

Chat

Attendance

UW Libraries

Add 4.0 Grade
Scale

Panopto
Recordings

Settings

Published

Edit

⋮

This is a graded discussion: 2 points possible

due Apr 4



R1: History and Vision for UbiComp

Jon Froehlich

Mar 29 at 9:39am

Each week, we will have one required reading and a set of optional papers, which I encourage you to skim over.

For the required reading, you must post a reading response below that:

- Summarizes the paper in ~3 sentences or less (in your own words).
- Enumerates key questions that you had while reading (could be technical, could be conceptual)
- Enumerates key things you liked as well as key criticisms ("I think this paper/system/research could have been improved if (or by)...").

For bonus points, individual students can volunteer to lead discussions of the *Required* papers or those marked *Optional for Discussion* below. See: https://canvas.uw.edu/courses/1199409/discussion_topics/4253123.

Required

- Weiser, Mark. (1991). "The Computer for the 21st Century." *Scientific American* 265, no. 3: 94-105.
<http://www.jstor.org/stable/24938718> .

Optional for Discussion

- Dourish, Paul (2004). "Chapter 2: Getting in Touch" in his book *Where the Action Is: The Foundations of Embodied Interaction*. [Dourish WhereTheActionIs Chapter2 Book2004.pdf](#)
- Rogers, Y., (2006). "Moving on from Weiser's vision of calm computing: engaging ubicomp experiences." *UbiComp'06*, https://link.springer.com/chapter/10.1007/11853565_24 .

Go Even Deeper

- Abowd, Gregory D. and Mynatt, Elizabeth D. (2000). "Charting past, present, and future research in ubiquitous computing." *ACM Trans. Comput.-Hum. Interact.* 7, 1 (March 2000), 29-58. DOI=<http://dx.doi.org/10.1145/344949.344988> .
- Weiser M., Brown J.S. (1997). "The Coming Age of Calm Technology." In: *Beyond Calculation*. Springer, New York, NY. https://link.springer.com/chapter/10.1007/978-1-4612-0685-9_6 .
- Want, R. (2010). "Chapter 1: An Introduction to Ubiquitous Computing" in the John Krumm edited book *Ubiquitous Computing Fundamentals*.
[Want AnIntroductionToUbiquitousComputing_KrummUbiCompFundamentalsBook2010.pdf](#)

What We Do

MAKERSPACE

[What We Do / MakerSpace](#)[About](#)[Getting Started](#)[Eligibility and Policies](#)[Tools and Resources](#)[Tool Lending Library](#)[Events](#)[Showcase](#)[FAQs](#)[Contact Us](#)

The CoMotion MakerSpace is a community-focused workshop. We have a wide range of capabilities with sewing machines, VR headsets, woodworking tools, and 3D printers under one roof. Make anything – be it a prototype, a community, or a connection.

[Get Involved](#)

More than just machines.



FREE FOR UW STUDENTS, FACULTY, AND STAFF.

OPEN TO THE PUBLIC.

MakerSpace

GETTING STARTED

MakerSpace / Getting Started

[About](#)

[Getting Started](#)

[Eligibility and Policies](#)

[Tools and Resources](#)

[Tool Lending Library](#)

[Events](#)

[Showcase](#)

[FAQs](#)

[Contact Us](#)



Come on over!

Our hours are 2-8 PM Sunday to Friday.

We're located on the second floor of Fluke Hall.



The Safety Orientation

You'll need to pass our Safety Quiz in order to use any of the MakerSpace's equipment.

Orientations are held at 5 PM on Tuesday/Wednesday/Thursday. RSVP is not needed.



Join Our Community

Get even more involved by joining our [Facebook group](#)! We also have a [Facebook page](#) where we'll post interesting goings-on at the MakerSpace.

We also have an email list, which you'll be automatically added to after you've attended an orientation.

CSE590 UbiComp Background Survey | Spring 2018

This survey is intended to help us better understand your background and goals. Having more information about you will hopefully improve our ability to teach you and, consequently, improve the overall quality of the course. Note: we may share anonymized responses in class.

- Jon and Liang

* Required

What is your name? *

Your answer

What was your undergraduate degree in? *

- ☐ Computer Science
- ☐ Design
- ☐ Architecture
- ☐ Art
- ☐ Engineering
- ☐ Humanities
- ☐ Science

ANDROID DEVICE

HUAWEI HONOR7X



Android OS 7 (Nougat)

1080x2160 display

Main camera (16MP + 2MP)

Back camera (8MP)

32GB ROM, Micro-SD up to 256

3GB RAM

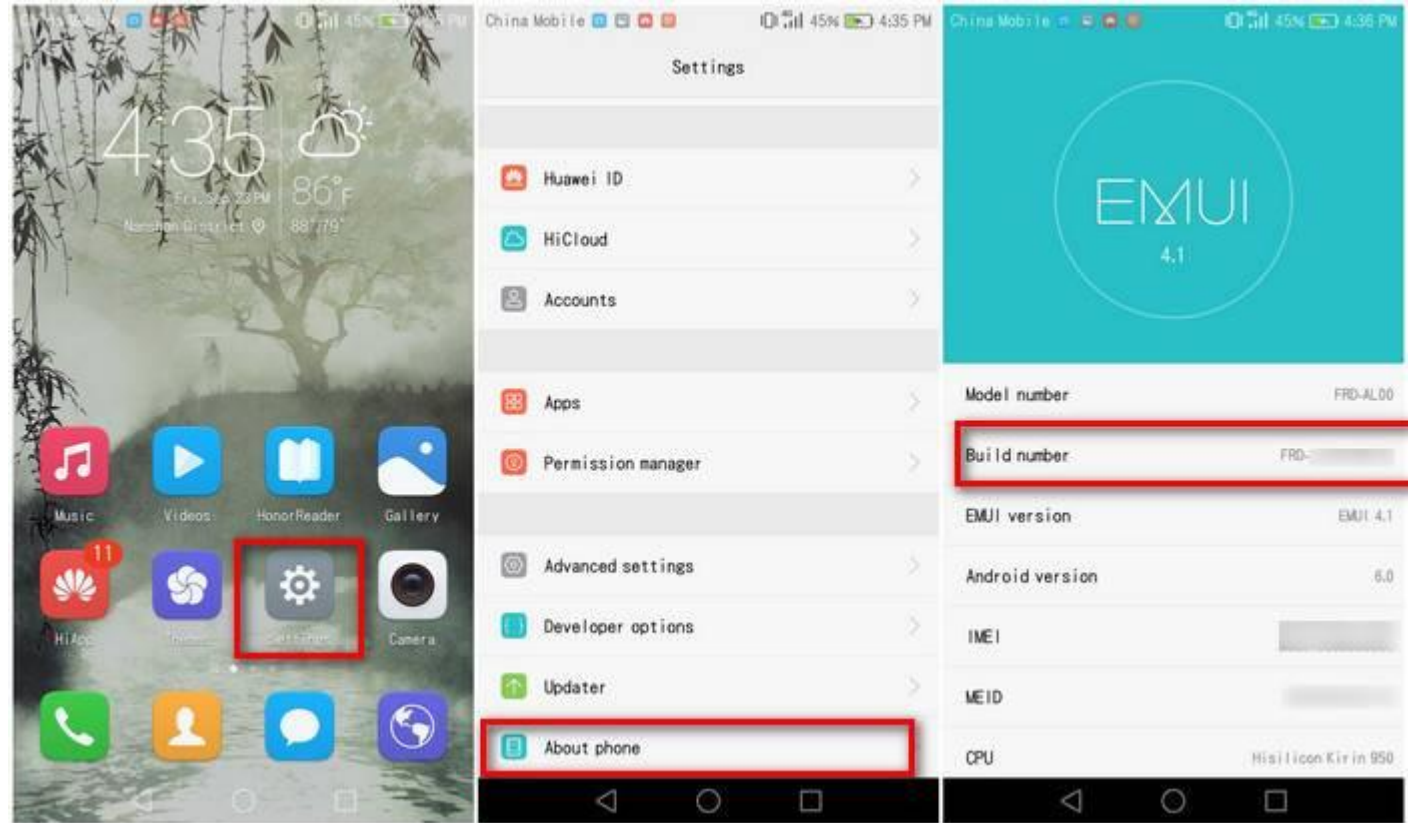
ENABLE DEVELOPER OPTIONS

Turn on your phone.

Tap on Settings Icon

Then, tap on About phone.

Scroll Down to locate Build Number. Now, tap seven times on it to enable Developer Options



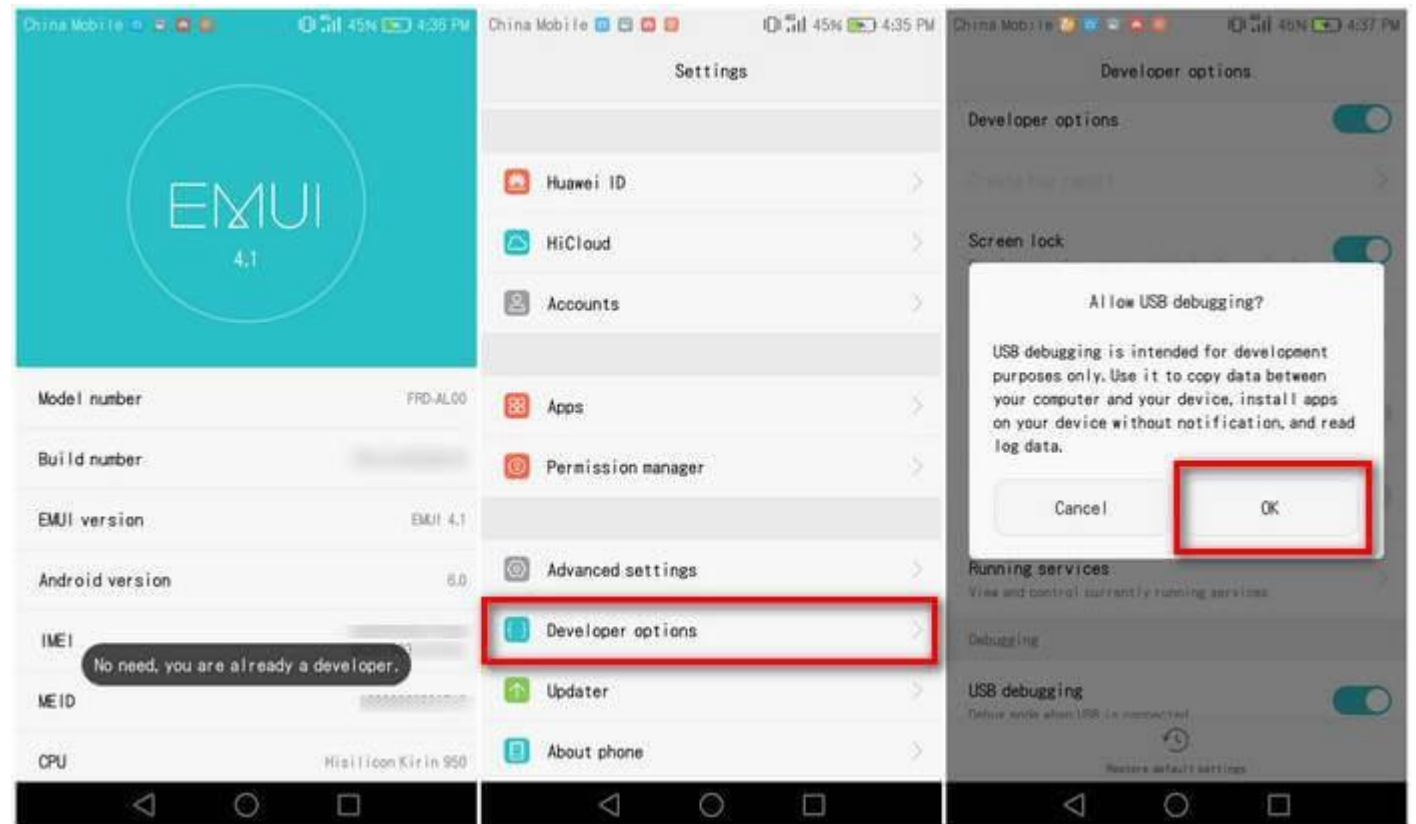
ENABLE USB DEBUGGING

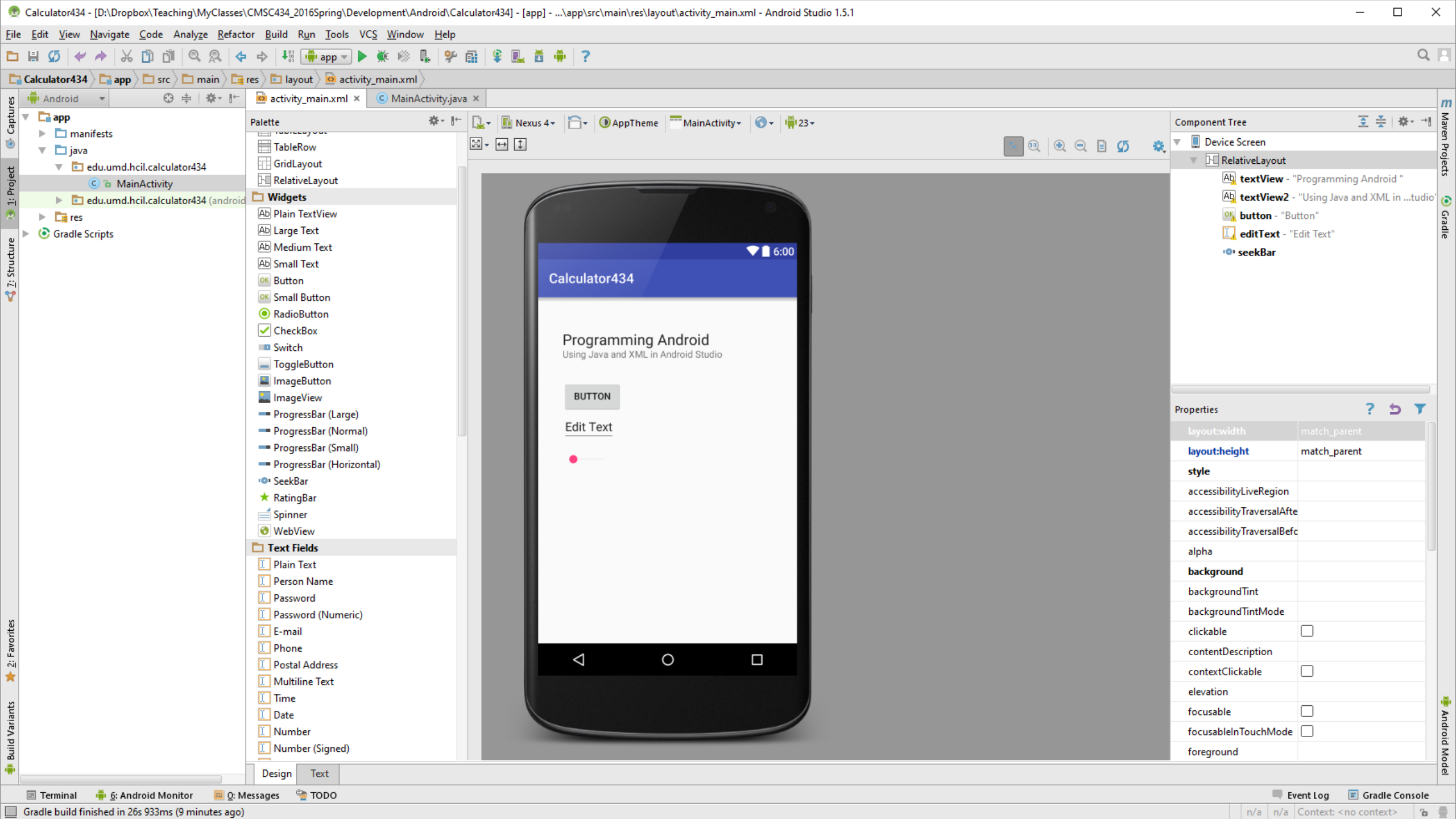
Tap on Settings icon

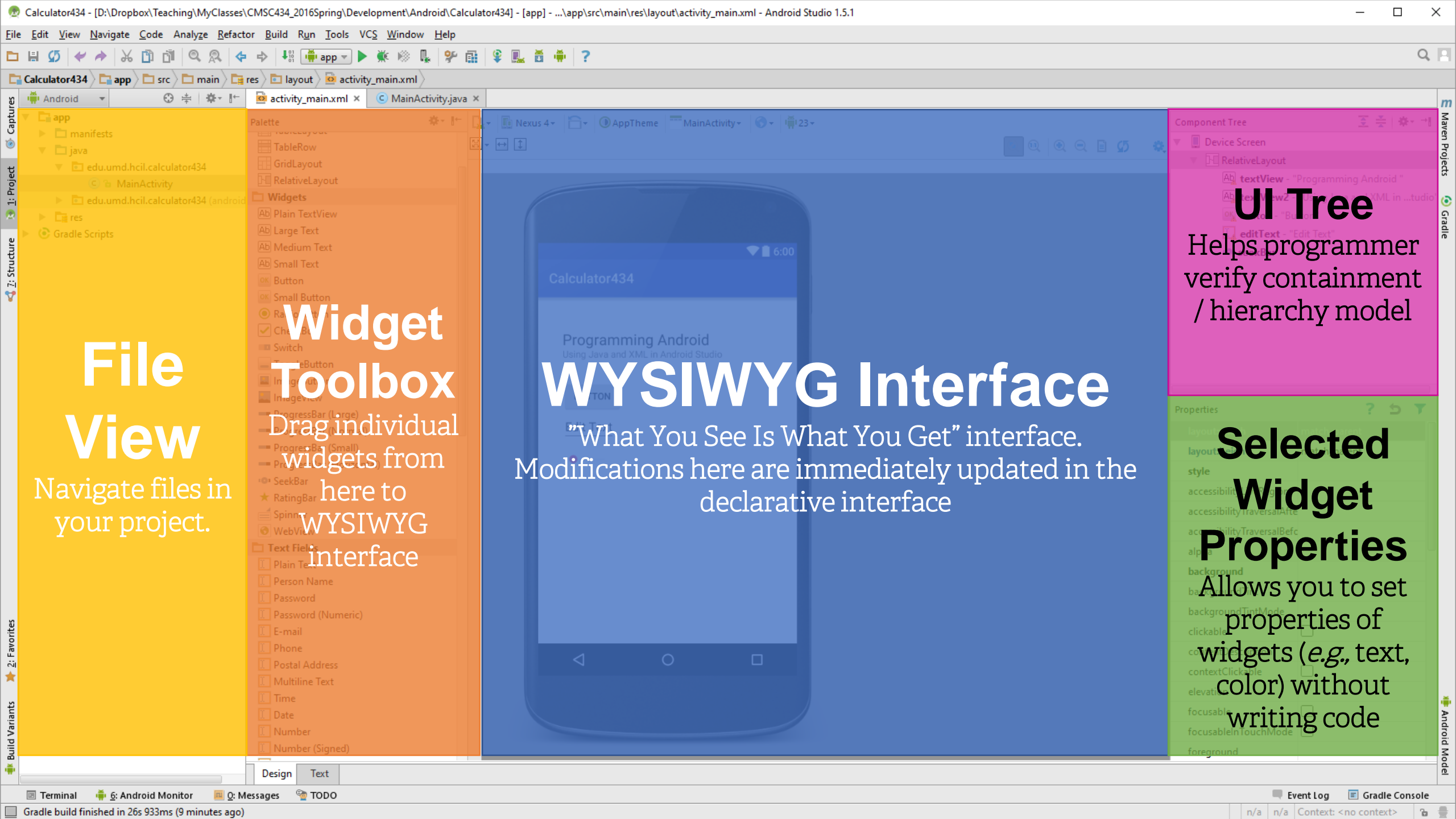
Then, Tap on Developer Options

Under Developer Options, you get USB Debugging Option.

Check the box, and give further confirmation to enable it.







File View

Navigate files in your project.

Widget Toolbox

Drag individual widgets from here to WYSIWYG interface

WYSIWYG Interface

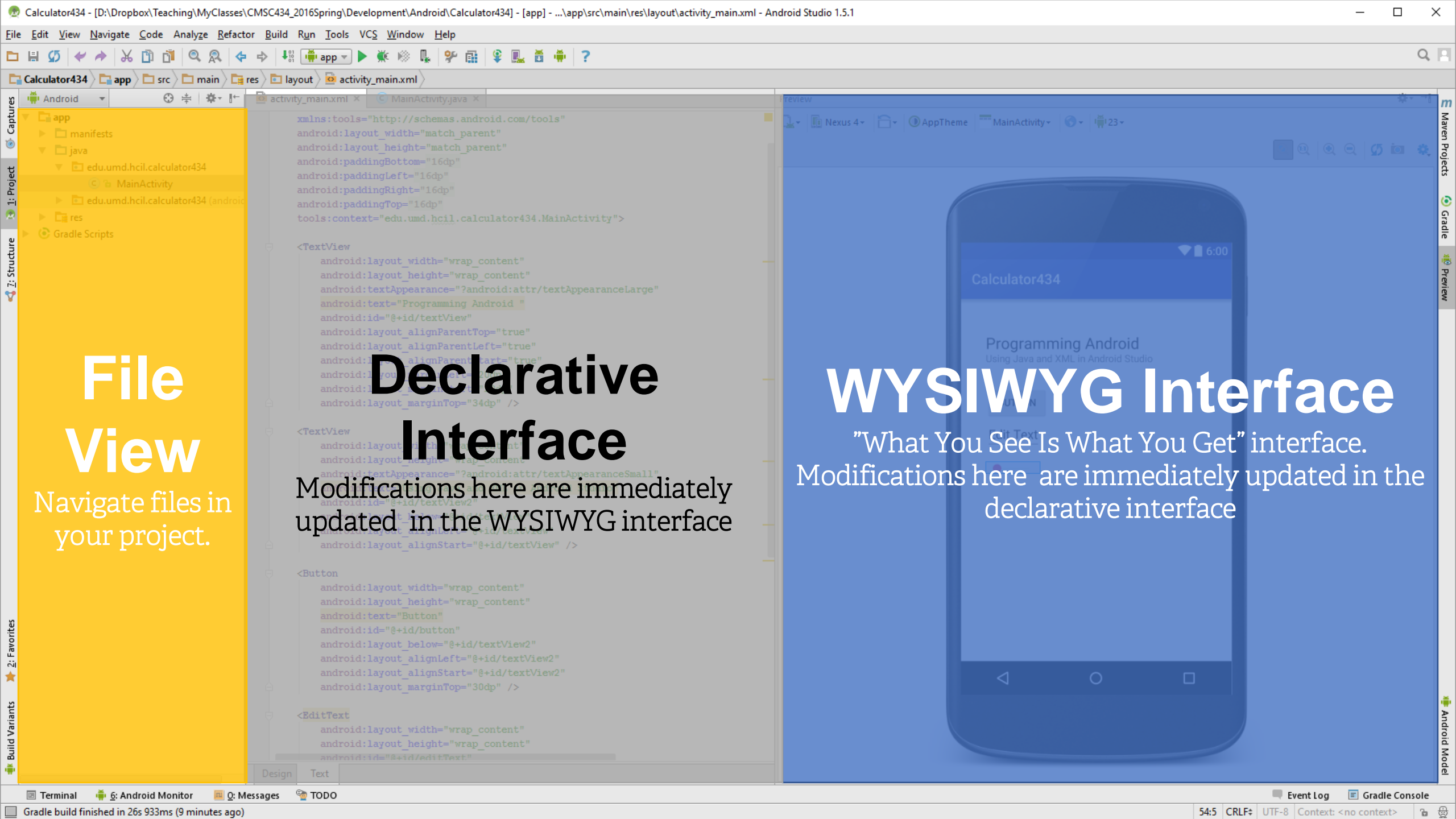
"What You See Is What You Get" interface. Modifications here are immediately updated in the declarative interface

UI Tree

Helps programmer verify containment / hierarchy model

Selected Widget Properties

Allows you to set properties of widgets (e.g., text, color) without writing code



File View

Navigate files in your project.

Declarative Interface

Modifications here are immediately updated in the WYSIWYG interface

WYSIWYG Interface

"What You See Is What You Get" interface. Modifications here are immediately updated in the declarative interface

BUILD TWO SIMPLE APPS

1. A “hello world” app that changes from “hello world” to “goodbye world” on a button press.

<https://developer.android.com/training/basics/firstapp/index.html>

2. A simple app that interacts with the sensor-based event system:

https://developer.android.com/guide/topics/sensors/sensors_overview.html