The background is a dark navy blue. In the top-left corner, there are several parallel teal lines that form a corner-like shape, extending towards the center. In the bottom-left corner, there are more parallel teal lines, some horizontal and some diagonal, creating a stepped effect. In the bottom-right corner, there are several parallel teal lines that form a diagonal shape, extending from the bottom towards the right edge.

Android Application Development Tutorial

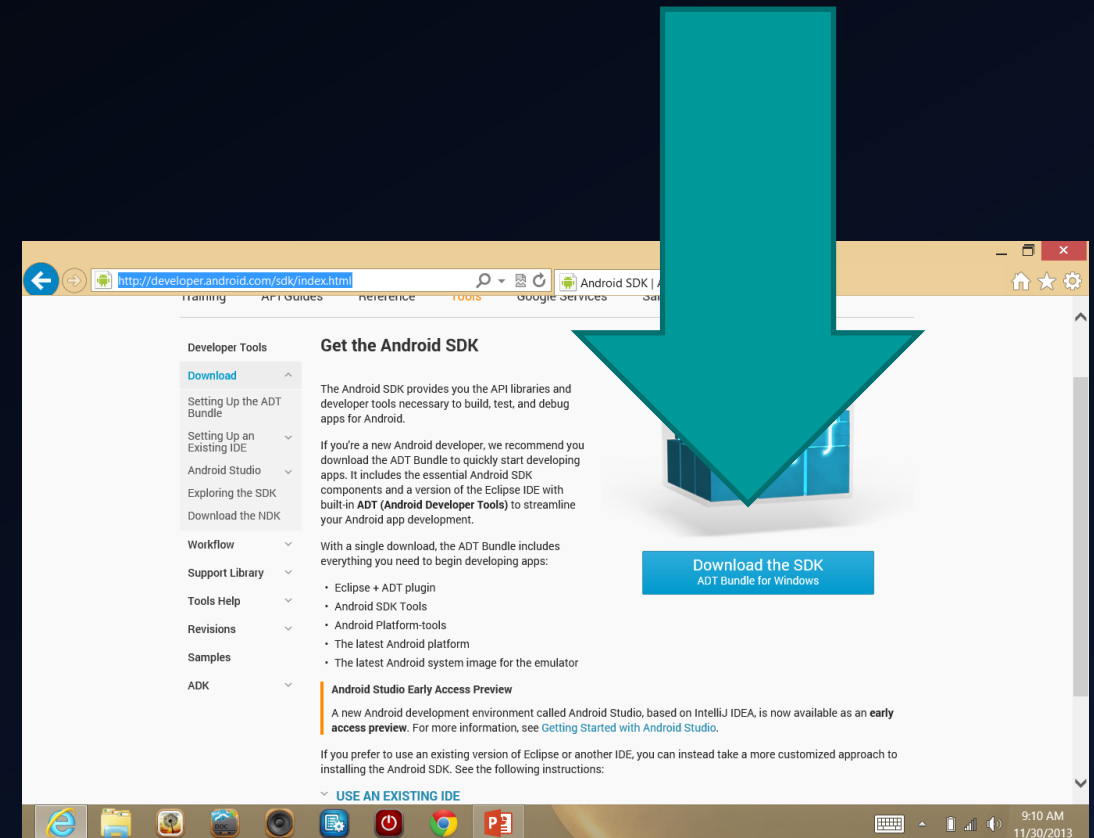
BY LEO AND GREG NEAT (MOSTLY LEO)



Step 1: Setting Up the Environment

Downloading Android SDK

- SDK is just about everything you need for development in one package
- Go to <http://developer.android.com/sdk/index.html>
- Consists of the eclipse package and all of the android software
- Run through the installation process
- Note: Make sure the os, Java and Eclipse all match # bits (32 or 64)



Opening Eclipse

- Go to where you saved the File
- Click android adt bundle, android adt bundle, eclipse, the eclipse application
- There are two, click the larger file, circular icon (305k)
- Then the IDE should open up



Setting Up the Application Creation

Setting Up the Application

- Click file, New, Android application,
- Name the application what ever you want as long as its unique
- Keep the package name but change the *example* part of it because android will not let you upload any thing with .example.
- Change it to your name instead of *example*
- Then set the minimum to froyo and the target for kit kat(4.4) (These are Android OS's)
- Then click next, the other fields are obsolete
- Use the default settings for the next page and just click next

The Opening Of a New Application cont.

- The Next page is pretty self explanatory, It is just creating the launcher icon for your application
- Once you hit next, you will get a new page that you should leave as the default for this level of coding.
- The Next page is where you pick the name for your java and xml files, they should be changed because they have to be unique, but the xml file you should keep as activity_whateveryourfilenameis
- Then Click Finish

Writing the software

- You will have 4 files appear, the xml file, the r file, the build file, and the java file (left hand margin) **I didn't see this**
- Automatically close the build file and the r file because they are automatically generated and should not be altered
- “Clean” your file first, this is done to make sure the auto generated files are correctly made
- You can do this by clicking the *Project* top tab, then clean....

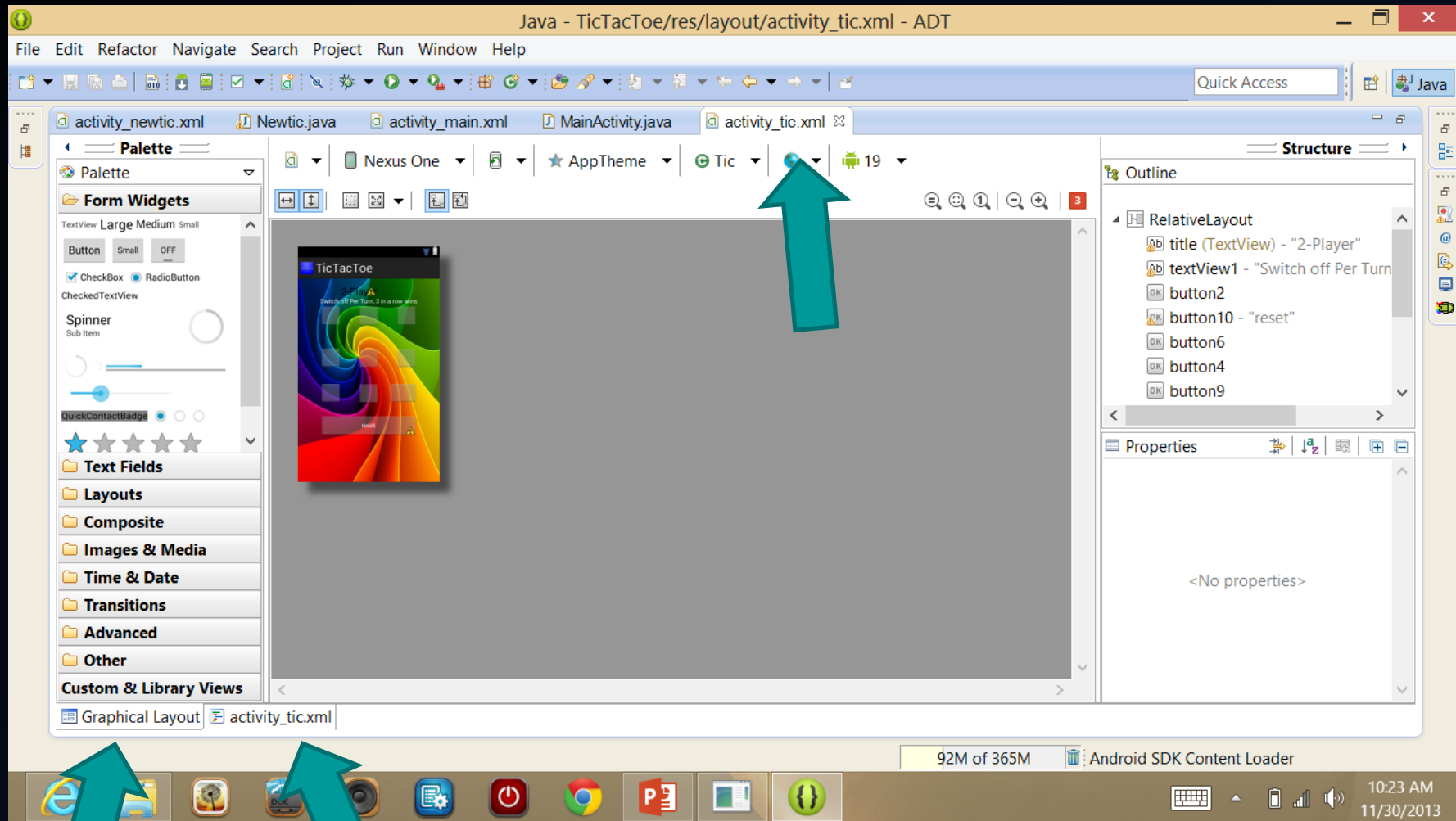
Writing the software: the files

- The r file: The r file is a bridge between java and xml it is used for functions but should never be altered (generated automatically)
- The build file: really dose not matter, just don't change it and it should be good (generated automatically)
- The java file: This is where the majority of your code will be written and can be found it the path of src , your file name, then it will be the java file
- The xml file: this is graphics part of the application and resembles html, this is found in the path res, layout, then your xml file

Writing the software: the xml file

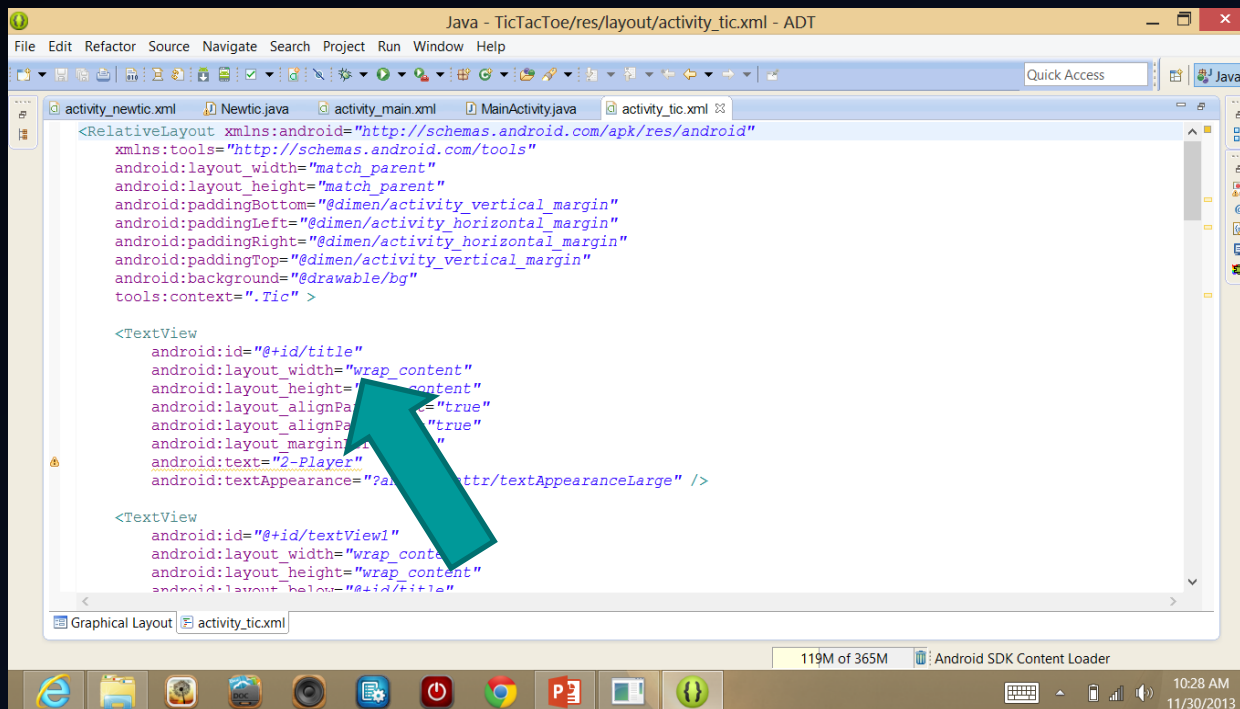
- Always start with your xml file by clicking on the top tag that says yourname.xml
- Make what ever button, text you need in order to have your successful program
- This is done by one of two ways
- The coding way is to click the tab at the bottom of the xml document that says the xml files name
- This will give you the opportunity to do directly code the graphics
- The other way, more commonly used is the graphics way, this is done by clicking the bottom tab, graphical layout
- Drag the desired items onto the output frame. The xml code is generated automatically from your dragging.

Writing the software: the xml file



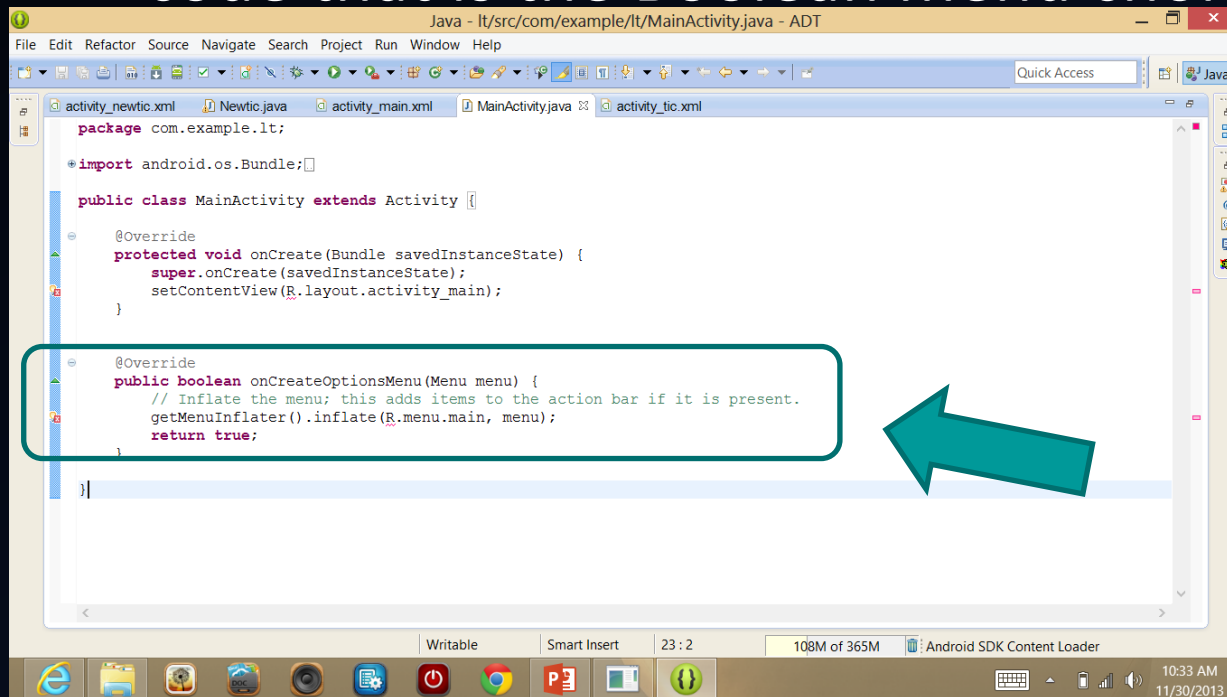
Writing the code: the xml file

- The xml options are found on the side of the graphical layout and have many types of user interfaces
- In the Xml actual code, the top line of what ever you created will have an id, this is how you will refer to it in the java file.
- The text boxes can be edited in the xml files to display whatever you want. They are all strings.



Writing the software: the java file

- Once you finish the layout of your app in xml, you will need to go to your java file
- Open the java file and you can start off by deleting the section of code that is the Boolean menu choice that is auto generated



Writing the application: the java file

- The java method *oncreate* is where you will be doing most of your coding (its like `main()` or `begin()`).
- This method is the one that runs on the opening of the app
- In there you can put what ever you want but you need to construct the things that you made in the xml file in order to use them
- Conceptually that is the end of the development process but there is a lot more details

Writing the application: the java file

- How you construct some thing you made in xml, in java is the universal line of code that you put into onCreate.

So it can be used in multiple methods

The R file

What ever your id was in xml

- `final Button one=(Button) findViewById(R.id.button1);`

What ever name you want it to be in java

The type of object

The constructing

All one line of code

Writing the application: the java file

- Most of your actions are done with Buttons In the beginning stages of development. They will use the on click command.
- This is :

```
yourButtonName.setOnClickListener(new View.OnClickListener()  
    {  
        public void onClick(View v)  
        { // put code here }); //this is all one line of code!!!
```


Writing the application: the java file

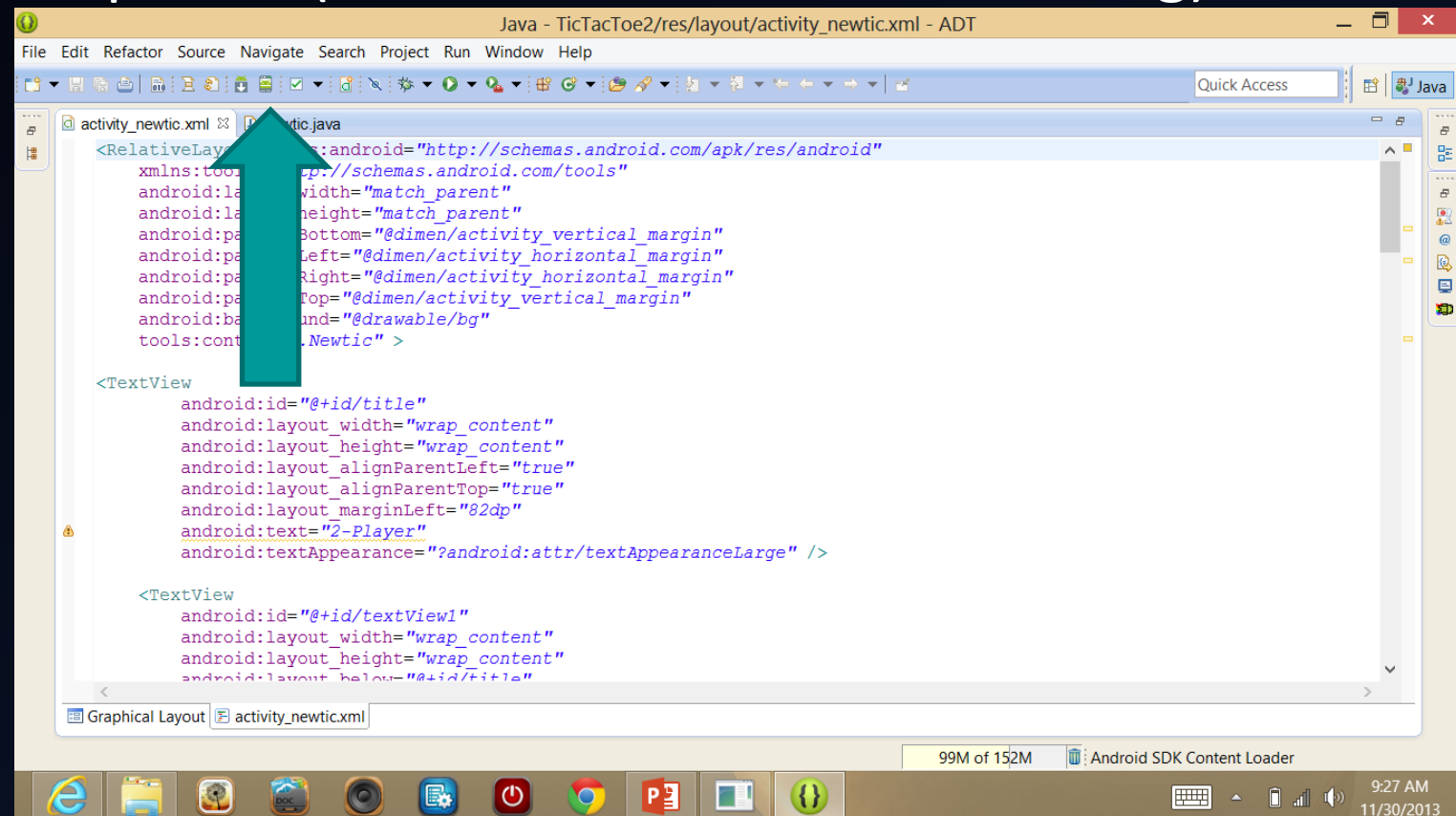
- Save both files so they know about each other. The astrix indicates the file(s) need(s) to be saved.
- Right click to import stuff like Buttons
- The classes available (like Buttons, Text,...) are listed in the xml file
- R.id. Provides a list of names of xml variables to connect to java variables. This is the file that is generated automatically.

Creating the Emulator



Creating an Emulator

- Finally, click the icon on the top tab that is 8 from the left called *Android Virtual Device Manager*, it looks like a small phone (make sure tool bar is showing)



Creating an emulator cont.

- An Emulator is a fake android device that is used for running the applications you write
- ADT manager should open up once clicked
- Click the *new* button, this is making a new emulator
- Name it what ever you want
- Select a small device initially (the size is directly proportional to the ram usage)
- Select the only available option for target and then press ok
- Now your emulator is created and you are ready to run your app.
- Note: if there are no compiling errors, make a new emulator with a different device size.
- Note2: the emulator can be really slow, so be patient. Folks with smart phones will skip this step and run their app directly on their phone.

Test Directly on Phone

- Still working on this

Put App in App Store

- I paid the \$25 fee for our group
- Gmail account for the group:
 - Name: Neato Neat
 - Username: appdeveloping@gmail.com
 - Password: shisho89
 - Birthday: 12/25/89
 - Male
 - Phone: 818 – 331 – 6245
 - Note: there is a misspelling in the username

Put App in App Store (cont)

- Good youtube video to help:
<http://www.youtube.com/watch?v=ReRCJgS-g9o>
- Important link referred to in the above video:
- <http://forum.xda-developers.com/showthread.php?t=1385846>
- Use group account (Neato Neat – previous slide) to publish account
- After publishing, search for App on Google Play