

Android Studio

CSC3054 / CSC7054

Advanced Features of the Emulator

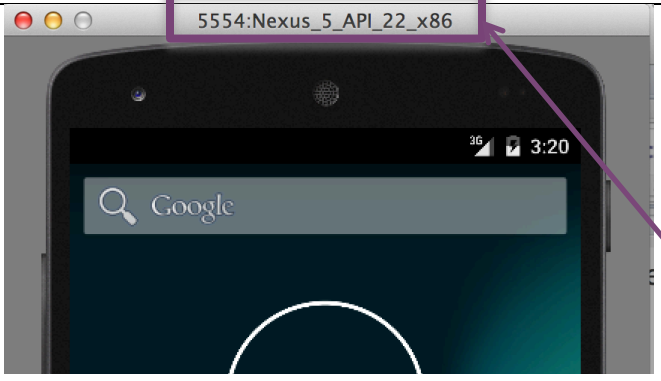
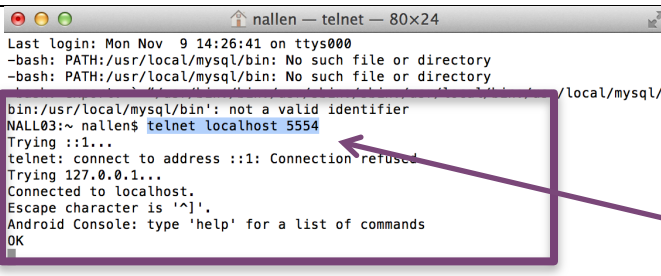
Advanced features of the Emulator

So let's take a look at some of the advanced features that the Android Emulator does support. It allows you to:

- Configure the emulator to emulate the speed and latency of different mobile networks.
- Configure the emulator to emulate different battery states (e.g. low on battery power/ currently charging the device).
- Inject mock location coordinates to make testing of location aware applications easier.
- Send text messages
- Make phone calls

These features would be used to help test code that must respond to the environment or environmental events. For instance, applications are often designed to do different things depending on the battery level. Let's take a look.

Connecting to the Emulator

Step 1		<p>Open an emulator running a virtual device.</p> <p>The port number on which the emulator's listening in the emulator window's title bar.</p> <p>In this case it's 5554.</p>
Step 2		<p>Open the terminal or command line window and use <code>telnet</code> to connect to the emulator.</p> <p>Type the command:</p> <pre>telnet localhost 5554</pre>

You can also connect to the emulator through the terminal window located within Android Studio

The screenshot displays the Android Studio 1.4 Beta interface. The top toolbar shows various development tools. The left sidebar contains the Project, Structure, and Gradle toolbars. The main editor area shows the MainActivity.java file with the following code:

```
public class MainActivity extends Activity {  
    //currency and percentage formatters  
    private static final NumberFormat currencyFormat = NumberFormat.getCurrencyInstance();  
    private static final NumberFormat percentageFormat = NumberFormat.getPercentInstance();  
  
    private double billAmount = 0.0; //bill amount entered by the user  
    private double customPercent = 0.18; //initial custom tip percentage  
    private TextView amountDisplayTextView; //shows formatted bill amount  
    private TextView percentageCustomTextView; // shows custom tip percentage  
    private TextView tip15TextView; //show 15% tip  
    private TextView total15TextView; // shows total with the 15% tip  
    private TextView tipCustomTextView; // shows the custom tip amount  
    private TextView totalCustomTextView; //shows total with the custom tip  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState); //call superclass's  
        setContentView(R.layout.activity_main); //inflate the G  
  
        //get reference to the TextViews  
        //that MainActivity interacts with programmatically  
    }  
}
```

The bottom panel shows the Terminal window with the following output:

```
bash: PATH:/usr/local/mysql/bin: No such file or directory  
bash: PATH:/usr/local/mysql/bin: No such file or directory  
bash: export: `="/usr/bin:/bin:/usr/sbin:/sbin:/usr/local/bin:/usr/local/mysql/b  
in:/usr/local/mysql/bin': not a valid identifier  
nall03:TipCalculatorNA nallen$
```

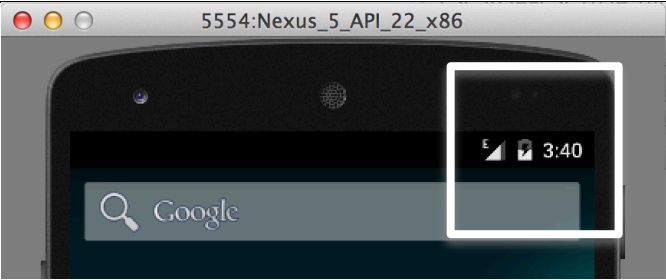
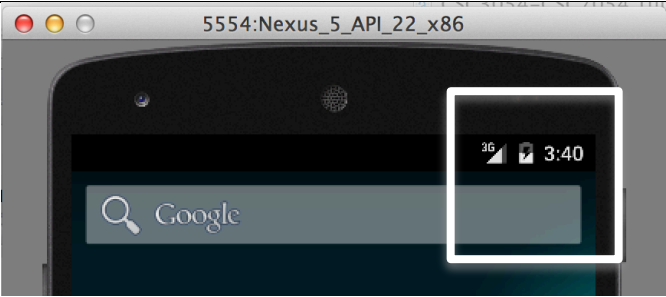
A purple speech bubble points to the Terminal window with the text "The terminal window within Android Studio".

Changing the network characteristics: Network Speed

It is possible to set the network characteristics to emulate a slower edge network, network speed edge. You can set a transfer rate or range at emulator startup or you can use the console to change the rate, while the application is running in the emulator. To set the network speed at emulator startup, use the `-netspeed` emulator option with a supported `<speed>` value, as listed in the table below. Here are some examples:

```
emulator -netspeed gsm
emulator -netspeed 14.4 80
```

The format of network `<speed>` is one of the following (numbers are kilobits/sec):

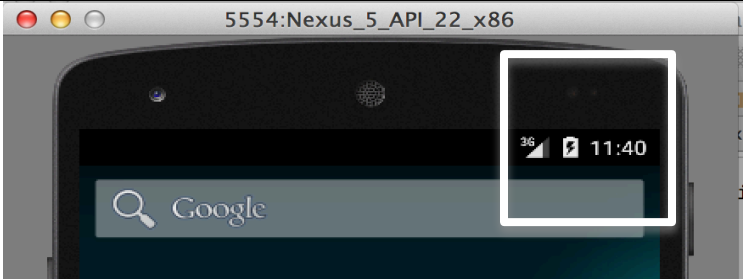
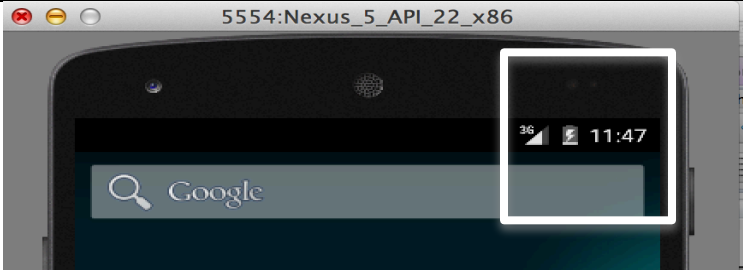
Value	Description	Comments
<code>gsm</code>	GSM/CSD	(Up: 14.4, down: 14.4)
<code>hscsd</code>	HSCSD	(Up: 14.4, down: 43.2)
<code>gprs</code>	GPRS	(Up: 40.0, down: 80.0)
<code>edge</code>	EDGE/EGPRS	(Up: 118.4, down: 236.8)
<code>umts</code>	UMTS/3G	(Up: 128.0, down: 1920.0)
<code>hsdpa</code>	HSDPA	(Up: 348.0, down: 14400.0)
<code>full</code>	no limit	(Up: 0.0, down: 0.0)
<code><num></code>	Set an exact rate used for both upload and download.	
<code><up>:<down></code>	Set exact rates for upload and download separately.	
Step 1		Change the network speed from fast to slow by typing <code>network speed edge</code> in the terminal window
Step 2		Change it back to 3G by typing <code>network speed full</code> in the terminal window

Changing the network characteristics: Battery Power

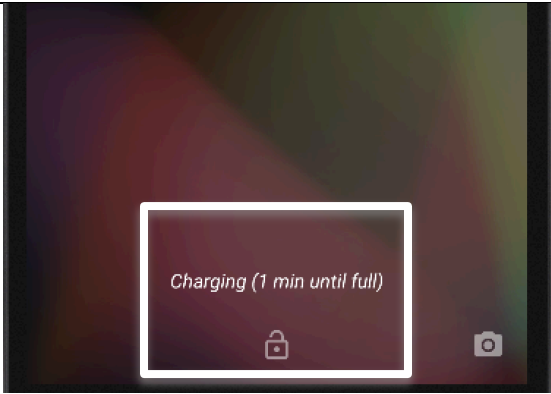
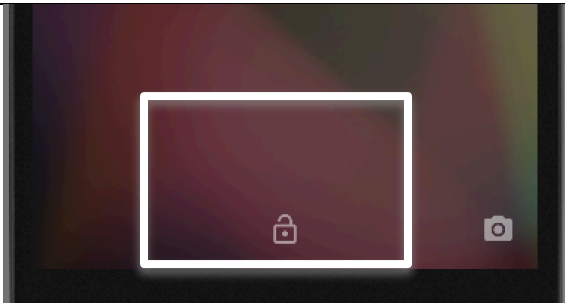
It is possible to change the battery status indicator in the notification bar to reflect a phone that is running low on battery, power capacity e.g. one that is operating at 10%. The `power` command controls the power state reported by the emulator to applications. The syntax for this command is as follows:

```
power <display|ac|status|present|health|capacity>
```

The `event` command supports the subcommands listed in the table below.

Subcommand	Description
<code>display</code>	Display battery and charger state.
<code>ac <on off></code>	Set AC charging state to on or off.
<code>status <unknown charging discharging not-charging full></code>	Change battery status as specified.
<code>present <true false></code>	Set battery presence state.
<code>health <unknown good overheat dead overvoltage failure></code>	Set battery health state.
<code>capacity <percent></code>	Set remaining battery capacity state (0-100).
Step 1 	<p>In the terminal window – type the command</p> <pre>power capacity 100</pre> <p>Notice that the device is at full charge.</p>
Step 2 	<p>In the terminal window – type the command</p> <pre>power capacity 10</pre> <p>Notice that the charging status reduced to 10%</p>



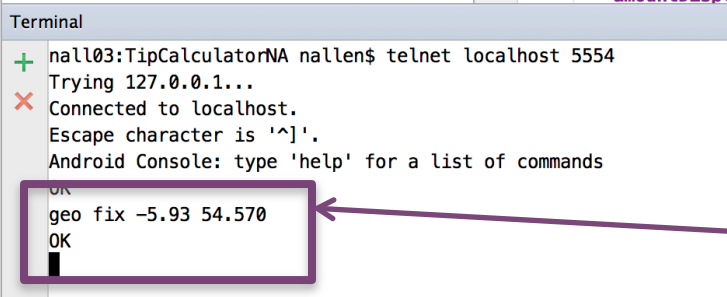
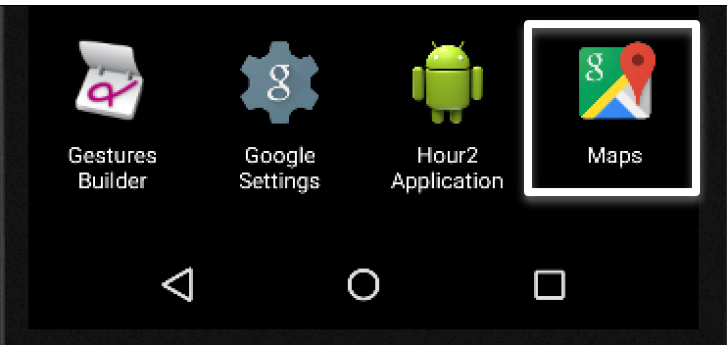
Step 3		<p>In the terminal window – type the command</p> <pre>power status charging</pre> <p>Notice that a 'Charging' message will appear on the emulated device.</p>
Step 4		<p>In the terminal window – type the command</p> <pre>power status not-charging</pre> <p>Notice that the 'Charging' message will disappear from the emulated device.</p>

Geo Location Provider Emulation

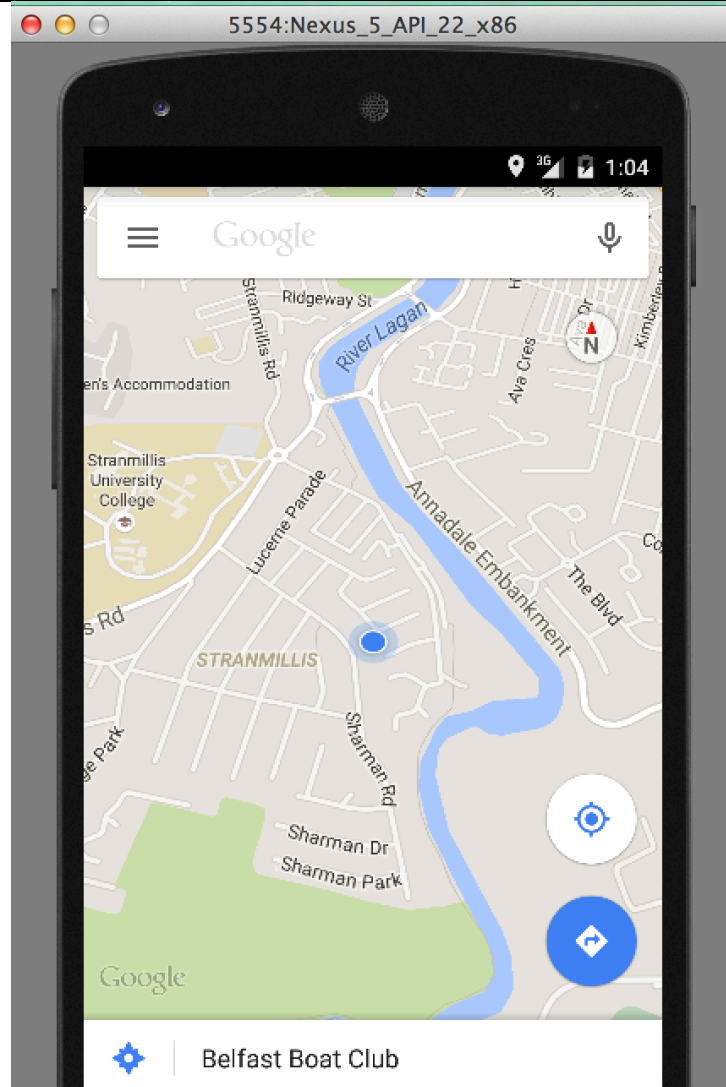
You can set the geographic location reported to the applications running inside an emulator. Use the `geo` command to send a simple GPS fix to the emulator, with or without NMEA 1083 formatting:

```
geo <fix|nmea>
```

The `geo` command supports the subcommands listed in the table below.

Subcommand	Description	Comments
<code>fix <longitude> <latitude> [<altitude>]</code>	Send a simple GPS fix to the emulator instance.	Specify longitude and latitude in decimal degrees. Specify altitude in meters.
<code>nmea <sentence></code>	Send an NMEA 0183 sentence to the emulated device, as if it were sent from an emulated GPS modem.	<code><sentence></code> must begin with '\$GP'. Only '\$GPGGA' and '\$GPRCM' sentences are currently supported.
Step 1		<p>Set the emulator's location to Belfast Boat Club, UK by typing in the following command:</p> <pre>geo fix -5.93 54.570</pre>
Step 2		<p>Open up Google Maps on the emulator</p>

Step 3



When you open up the maps application, you can see that it shows the location as Belfast, Boat Club.

Send a Text Message

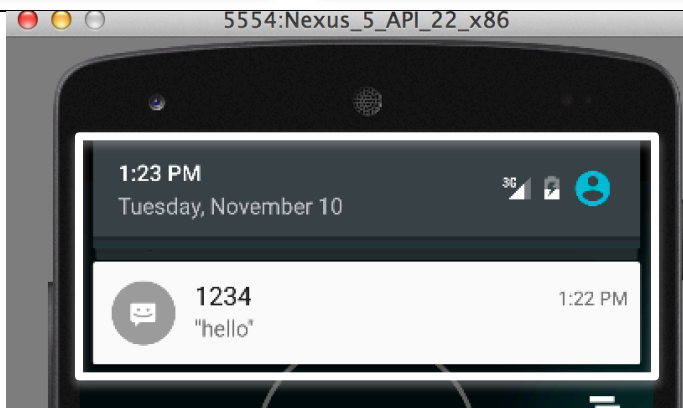
The Android emulator also allows you to emulate networked interactions, such as receiving a SMS message. Once you connect to an emulator instance, you can generate an emulated incoming SMS using the following command:

`sms send <senderPhoneNumber> <textmessage>` where `<senderPhoneNumber>` contains an arbitrary numeric string.

The console forwards the SMS message to the Android framework, which passes it through to an application that handles that message type.

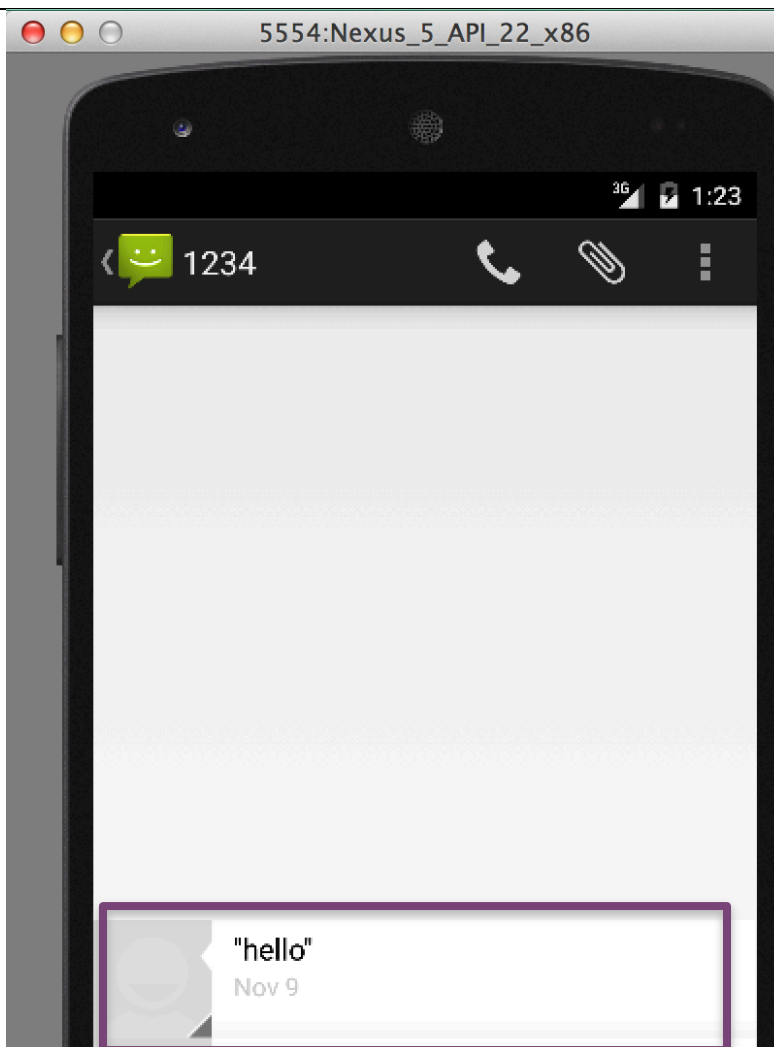
Step 1		<p>In the terminal window, reconnect to the emulator:</p> <pre>telnet localhost 5554</pre>
Step 2		<p>Give the emulator a command, which will cause it to emulate an incoming SMS message.</p> <pre>sms send 1234 "hello"</pre> <p>Press return</p>
Step 3		<p>The emulator's notification bar will contain an icon indicating that the emulated phone just received a SMS message.</p>

Step 4



Pull down on the notification bar to start the messaging application

Step 5



Take a closer look at the SMS message.

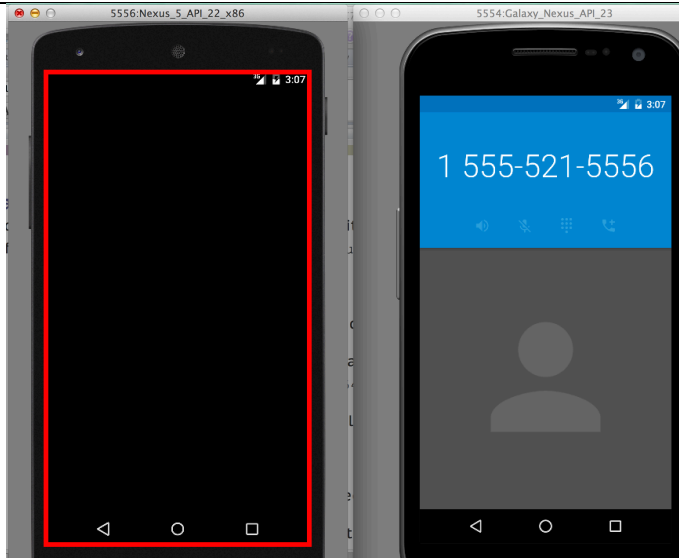


Make and Receive a Phone Call

Android allows two emulators to interact with each other. Please note, you MUST have two different AVDs running for this to work e.g. Nexus 5 and Galaxy Nexus

Step 1		Run two emulator instances
Step 2		Open the phone application and start to dial the number of the second.
Step 3		The number of the second emulator is the port number shown at the top of that emulator window, in this case, 5554.

Step 4



The second emulator will receive the call

It will ring its user.

Pick up the incoming call in the second emulator.

The emulator's interface will change to reflect that the call is connected. If one of the parties hits the hold button, it will be reflected in both phone applications. Once the users are done with the call they can hang up and both emulators will show that the call has been disconnected.

This is just a few of the things you can do with the emulator. There are many other interesting features as well. To learn more, look at the emulator page on the Android developer's website.