

## 1

### Getting started

7

Introducing Android	8
Beginning your first app	10
Exploring the designer	12
Adding components	14
Applying behaviors	16
Running your first app	18
Testing by emulator	20
Installing your first app	22
Managing projects	24
Summary	26

## 2

### Designing interfaces

27

Enabling buttons	28
Reading text input	30
Inserting images	32
Painting canvas	34
Picking list items	36
Checking boxes	38
Storing data	40
Telling the time	42
Configuring screens	44
Summary	46

## 3

### Controlling progress

47

Composing programs	48
Defining variables	50
Performing operations	52
Branching flow	54
Providing alternatives	56
Notifying messages	58
Looping within a range	60
Looping through a list	62
Looping while true	64
Summary	66

# 4

## Calling functions

67

Calling object methods	68
Creating procedures	70
Passing arguments	72
Returning results	74
Ignoring results	76
Calling subroutines	78
Validating input	80
Doing mathematics	82
Generating random numbers	84
Summary	86

# 5

## Managing text

87

Manipulating strings	88
Querying strings	90
Comparing strings	92
Trimming strings	94
Splitting strings	96
Extracting substrings	98
Summary	100

# 6

## Handling lists

101

Making lists	102
Querying lists	104
Adding list items	106
Selecting items	108
Swapping items	110
Manipulating lists	112
Separating lists	114
Summary	116

# 7

## Embracing media

117

Playing sounds	118
Playing video	120
Snapping photos	122
Picking images	124
Switching screens	126
Animating components	128
Detecting collisions	130
Dragging objects	132
Dropping objects	134
Summary	136

# 8

## Sensing conditions

137

Pin-pointing location	138
Recognizing orientation	140
Feeling movement	142
Picking contacts	144
Calling phone numbers	146
Texting messages	148
Tweeting updates	150
Storing data online	152
Summary	154

# 9

## Deploying apps

155

Planning the program	156
Assigning static properties	158
Designing the interface	160
Initializing dynamic properties	162
Adding runtime functionality	164
Completing functionality	166
Distributing the application	168
Selling your app	170
Summary	172

# +

## Handy reference

173

Procedures blocks	174
Variables blocks	175
Text blocks	176
Lists blocks	178
Control blocks	180
Math blocks	182
Logic blocks	185
Colors blocks	186

## Index

187

# Preface

The creation of this book has provided me, Mike McGrath, a welcome opportunity to demonstrate how MIT App Inventor can be easily used to create applications for today's portable devices. All examples I have given in this book demonstrate app features using the current online version of App Inventor 2 hosted by the MIT Center for Mobile Learning.

## Conventions in this book



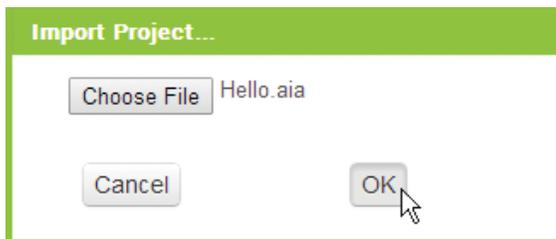
Hello.apk

The examples provide screenshots of the actual App Inventor blocks used to implement each step for clarity. Additionally, in order to identify each example described in the steps, a colored icon and an Android Application Package file name appears in the margin alongside the steps.

## Grabbing the source code

For convenience I have placed source code files from the examples featured in this book into a single ZIP archive. You can obtain the complete archive by following these easy steps:

- 1 Browse to [www.ineasysteps.com](http://www.ineasysteps.com) then navigate to [Free Resources](#) and choose the [Downloads](#) section
- 2 Find [Building Android Apps in easy steps, 2nd Edition](#) in the [Downloads](#) list, then click on the hyperlink entitled [All Code Examples](#) to download the archive
- 3 Now, extract the archive contents to any convenient location on your computer – the source code for each example is contained in its own individual AIA archive
- 4 You may then upload any AIA archive example into App Inventor using the “Import project (.aia) from my computer menu” option on the My Projects page Project menu



I sincerely hope you enjoy discovering the powerful, exciting possibilities of MIT App Inventor and have as much fun with it as I did in writing this book.

Mike McGrath

# 1

# Getting started

*Welcome to the exciting world of application development for Android. This chapter shows how to establish an app development environment and demonstrates how to create a simple Android app.*

- 8** Introducing Android
- 10** Beginning your first app
- 12** Exploring the designer
- 14** Adding components
- 16** Applying behaviors
- 18** Running your first app
- 20** Testing by emulator
- 22** Installing your first app
- 24** Managing projects
- 26** Summary

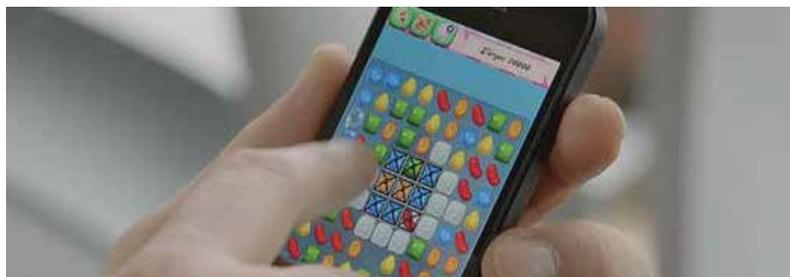


# Introducing Android

Android is an operating system for mobile phones and tablets, in much the same way that Microsoft Windows is an operating system for PCs. The Android operating system is maintained by Google and comes with a range of useful features as standard.

Standard Android features include Google Search and Google Maps, which means you can easily search for information on the web and find directions from your phone – as you would on your computer. This is handy for discovering things like train times and getting directions when out and about. Other Google services, such as Gmail and Google Earth can also be accessed from devices running the Android operating system. You can easily check Facebook and Twitter profiles too, through a variety of applications (apps) – making it ideal for social networking.

There is a huge range of custom apps available to download from the Google Play Store. For example, there are camera apps such as “Camera 360” – that allow you to take photos with artistic effects, and music player apps such as “Winamp” – that allow you to import MP3s and create playlists, and popular game apps, such as “Candy Crush Saga” – that provide great fun entertainment.



Android is an open-source operating system, built on the open-source Linux Kernel, which means it can be easily extended to incorporate new cutting-edge technologies as they emerge. Android was brilliantly designed, from the ground-up, to enable developers to create compelling apps that can fully exploit all the host device’s capabilities. For example, an app can access all of a phone’s core functionality such as making calls, sending text messages, or taking photos. The Android platform will continue to evolve as the developer community works together to build innovative mobile applications and you can be part of this exciting innovation process with MIT App Inventor 2.



The New icon pictured above indicates a new or enhanced feature introduced in the latest version App Inventor 2.

...cont'd

## MIT App Inventor 2

App Inventor enables you to develop applications for Android devices using a web browser and a connected device or a device emulator – without writing a single line of code. It is a web-based tool developed by MIT in which the App Inventor servers store your work and help you keep track of your projects. When the app is finished you can package it to produce an “application package” (Android **.apk** file) that can be shared around and installed on any Android phone or tablet like any other app. App Inventor 2 is supported by a wide range of operating systems and web browsers, with these minimum specifications:



Computer operator system requirements...

- **Windows:** Windows XP, Vista, Windows 7, Windows 8+
- **Linux:** Ubuntu 8+, Debian 5+
- **Mac:** Mac OS X 10.5+

Web browser requirements...

- **Google Chrome:** 4.0+
- **Mozilla Firefox:** 3.6+
- **Apple Safari:** 5.0+

App Inventor 2, which is demonstrated in this book, runs completely in the browser – unlike App Inventor 1, which was demonstrated in the previous edition of this book. This means you can begin building apps online immediately but to test your apps you will need to install one or two pieces of software:

- **MIT AI2 Companion:** for installation on an Android device to enable live testing on that device via a WiFi network
- **aiStarter:** for installation on a computer system to enable the web browser to communicate with an Android device emulator

The procedure of testing apps on an virtual emulator and live testing on a physical device are demonstrated in this chapter.



At the time of writing Microsoft Internet Explorer is not supported by MIT App Inventor 2.

# Beginning your first app

The App Inventor 2 interface, where you visually design and build apps for Android devices, is online at [ai2.appinventor.mit.edu](https://ai2.appinventor.mit.edu). App Inventor requires you to sign in with a Google account.

- 1 Open a web browser and navigate to [accounts.google.com](https://accounts.google.com) then complete the form there to create a Google account

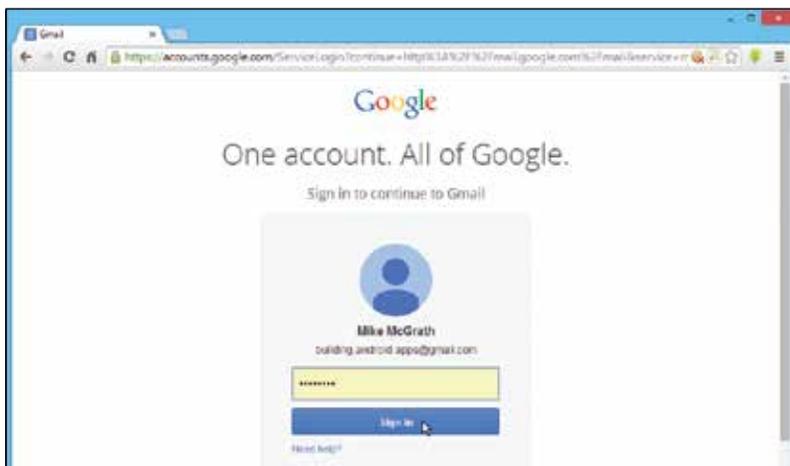


- 2 Next, attempt to navigate to the App Inventor 2 interface at [ai2.appinventor.mit.edu](https://ai2.appinventor.mit.edu) – you will be redirected to a Google Sign In page

- 3 Now, enter your Google account details to sign in

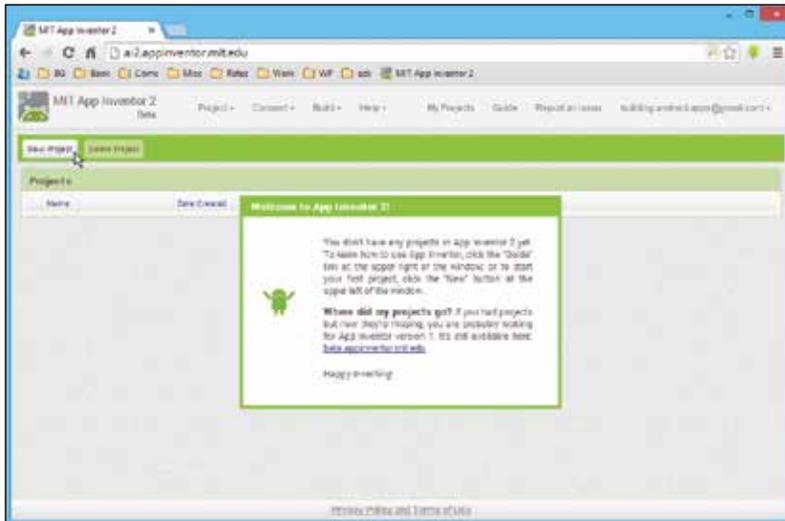


Use your existing Google account to sign in if you already have one.



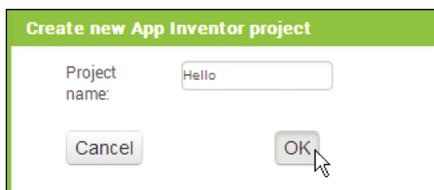
...cont'd

After signing in you will be redirected to the App Inventor 2 interface. Upon your first visit, App Inventor opens in “Projects mode” with no previously created projects and looks like this:

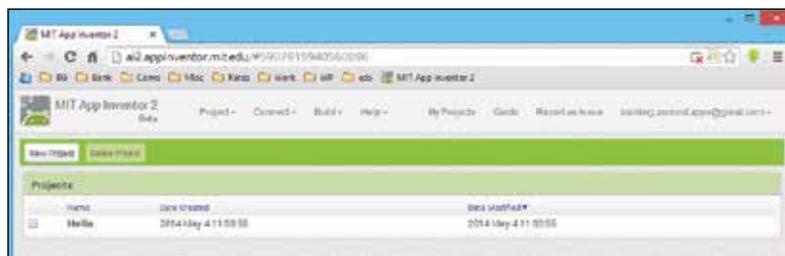


App Inventor 2 runs entirely in the web browser – unlike the previous version that required Java support.

- 4 Click the “New Project” button to launch the New App Inventor Project dialog
- 5 Type the Project Name **Hello** and click OK – to see a new empty project appear



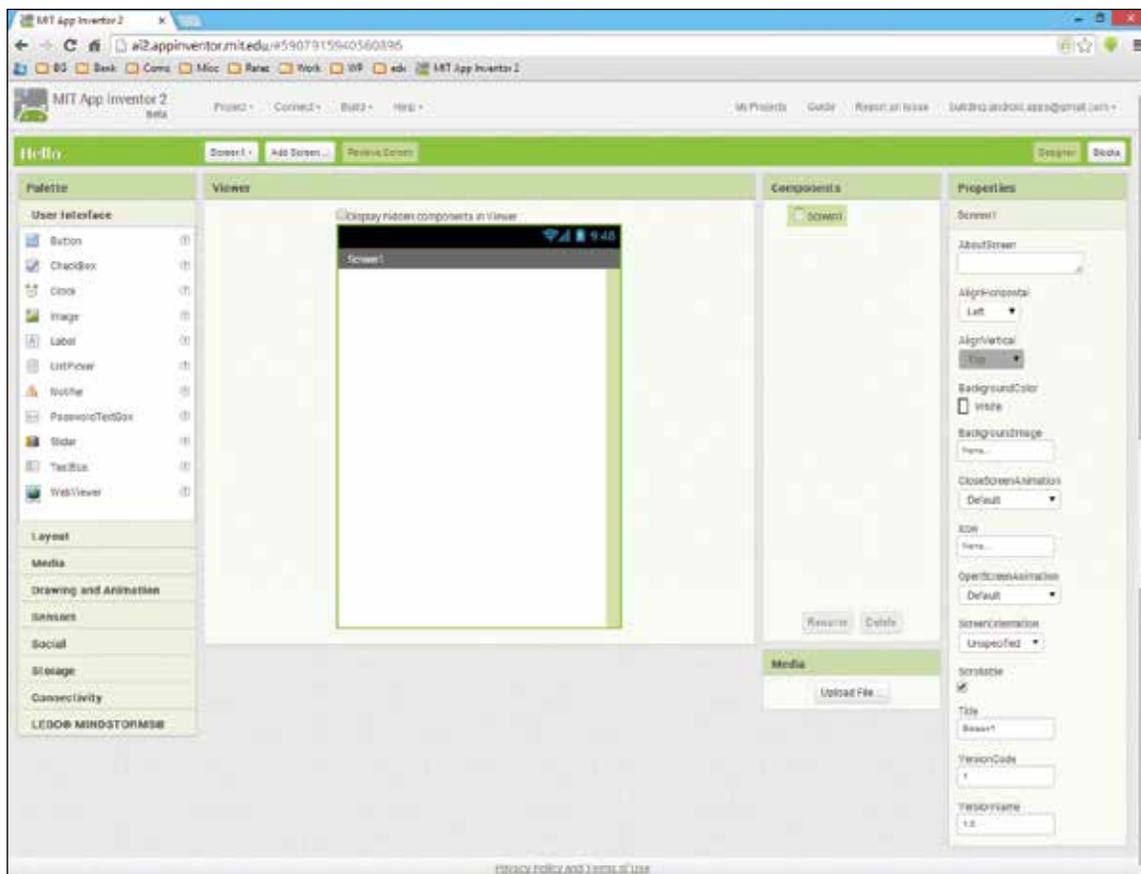
Name new projects capitalized with an uppercase first character.



# Exploring the designer

Having created an App Inventor project, as described on the previous page, it can be opened for work in “Designer mode”.

- 1 On the App Inventor Projects page click on the project item you wish to open – see App Inventor switch to its Designer mode with the chosen project opened



...cont'd

- 2 Examine the green App Inventor title bar to see it contains the app name, buttons to work on multi-screen apps, and buttons to switch between Designer mode and a “Blocks” editor mode
- 3 Now, explore the App Inventor Designer to see it comprises these five columns:
  - **Palette:** containing individual components that can be added to the interface, grouped by category – such as “User Interface”
  - **Viewer:** visually representing components added to the interface – a screen container component is provided by default
  - **Components:** listing components added to the interface – arranged hierarchically beneath the screen component
  - **Media:** listing media resources used by the app – such as images, audio, and video files
  - **Properties:** listing editable characteristics of the component currently selected in the Viewer column – such as “Screen1”
- 4 Finally, notice that the top App Inventor menu bar remains constant in any view – click the “My Projects” menu item at any time to return to the Projects page



The Media column is located directly below the Components column.



You can also return to the Projects page at any time by choosing the “My Projects” item on the “Projects” menu.

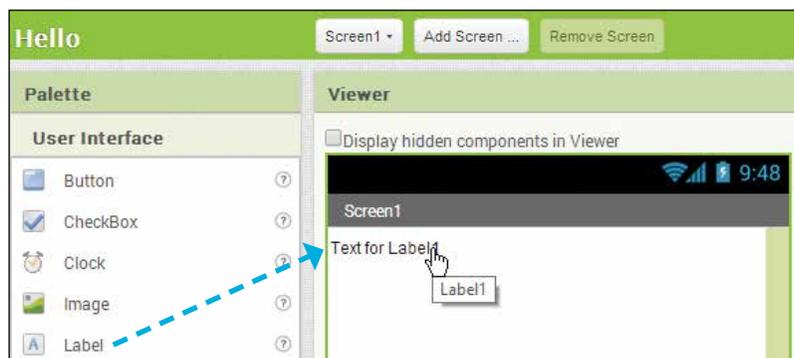


Notice that when a component gets dropped onto the Viewer it gets added to the list of items in the Components column and its characteristics appear in the Properties column.

# Adding components

The first stage in creating an app is to design the user interface by adding components from the Palette column in Designer mode. Follow these easy steps to add components to the Hello project, begun on the previous page, on your own system:

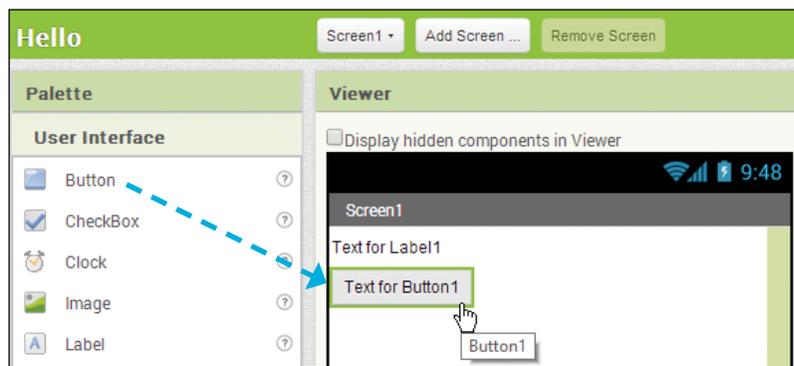
- 1 Click on the Palette column's User Interface category to reveal the component items it contains
- 2 Next, click on a Label component in the Palette column's User Interface category and drag a copy of it to the Viewer column – release the mouse button to drop it



- 3 Now, click on a Button component in the Palette column's User Interface category and drag a copy of it to the Viewer column – release the mouse button to drop it



At any time in Designer mode you can click on a component in the Viewer to "select" it and see it get highlighted in the Components column, and then see its characteristics appear in the Properties column.



...cont'd

- 4 Click on the Label component added to the Viewer to select it – see its characteristics appear in the Properties column

Items listed in the Properties column can be edited by clicking on them to change text values, check/uncheck boxes, or choose from alternatives that appear in pop-up dialog boxes.

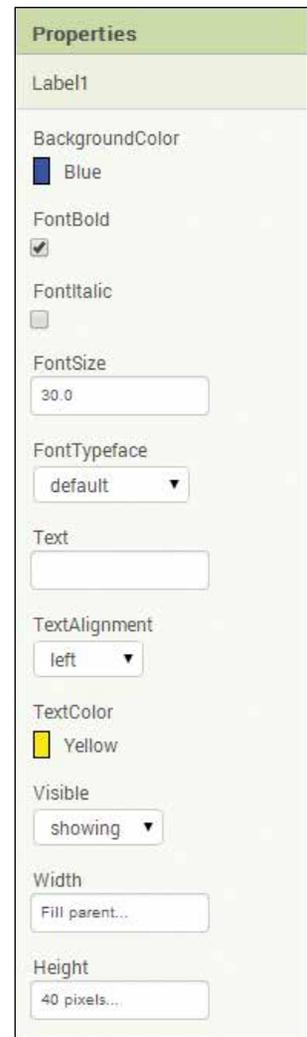
- 5 Edit the Label component's characteristics in the Properties column to set its BackgroundColor to "Blue", FontBold to checked, FontSize to "30", Text to blank, TextColor to "Yellow", Width to "Fill parent", and Height to "40" – see the Label's look change in the Viewer

- 6 Now, click on the Button component in the Viewer to select it – see its characteristics appear in the Properties column

- 7 Edit the Button component's characteristics in the Properties column to set its FontSize to "30" and Text to "Click Me" – see the Button's look change in the Viewer



The components added to the Viewer have been automatically named "Label1" and "Button1" by App Inventor but as yet have no functionality. The next stage in creating an app is to add functionality – making the Button respond to a user action by displaying a response message in the Label component.



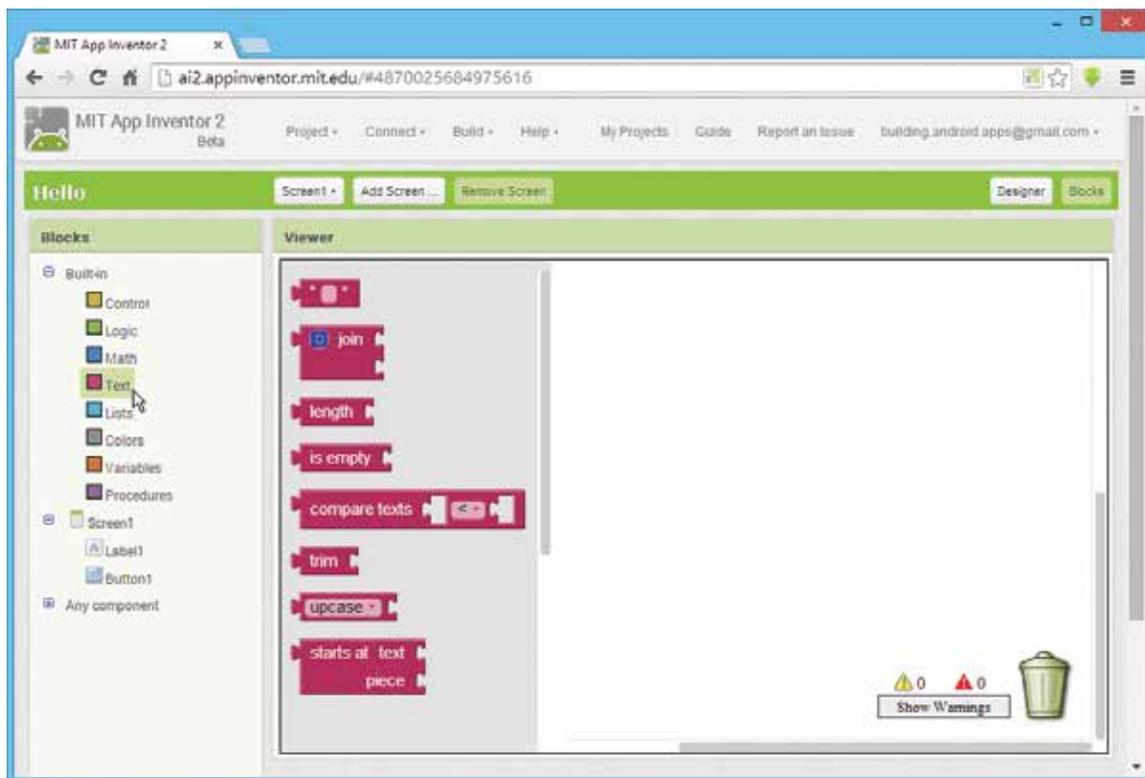
# Applying behaviors

The second stage in creating an app is to apply functionality to components added to the user interface in App Inventor Designer. This stage employs the App Inventor “Blocks mode” editor that allows you to easily add functionality without writing any program code. Follow these easy steps to add “Click” behavior functionality to components of the Hello project, added on the previous page:



1

On the App Inventor menu bar, click on the “Blocks” button – to switch to the Blocks mode editor



The Blocks mode editor has a “Blocks” column listing “drawers” (nodes) that each provide various individual colored blocks of code text that can be snapped together, like jigsaw-puzzle pieces, to form instructions to be performed by the app. The selection of blocks available for any listed drawer appear when you click on that drawer in the Blocks column. The blocks are assembled by choosing an available block then dragging it onto the Blocks editor’s “Viewer” area.

...cont'd

- 2 Click on the **Button1** drawer – to reveal its blocks
- 3 Now, drag the top block containing the text **when Button1.Click do** onto the Blocks editor's Viewer area



- 4 Click the **Label1** drawer and drag the **set Label1.Text to** block to the Viewer – snapping it into the previous block



- 5 Click the Built-in **Text** drawer and drag its " " basic text block onto the Viewer and snap it into the previous block
- 6 Finally, click on the text block's " " content area, to make it editable, and change the content to **"Hello World!"**



App Inventor 2 automatically adds a corresponding component drawer directly in the editor's Blocks column whenever you add a component in Designer mode – unlike the previous version that placed them into a "My Blocks" drawer.



If you snap together blocks that form an invalid instruction, the Blocks editor will advise you of the error.

# Running your first app

The recommended way to build apps with App Inventor requires your computer to share a WiFi connection with your Android device, so you can run the app as you build it for “live testing”, and the **MIT AI2 Companion** to be installed on the Android device to allow a “handshake” connection to be established between App Inventor and the device using a generated code.

- 1 On your Android device, go the Google Play Store and install the MIT AI2 Companion



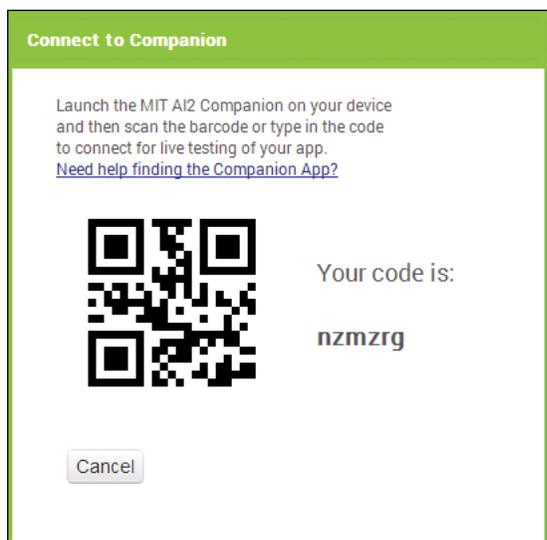
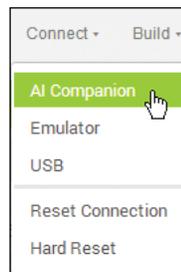
Installation of the MIT AI2 Companion is best done from the Play store. Loading the Companion directly (“side loading”) means it will not receive automatic updates.



- 2 On your computer, with the Hello app open in App Inventor, choose Connect > AI Companion on the menu to see a generated 6-character code



The option shown to connect via USB is fraught with driver issues so is best avoided.



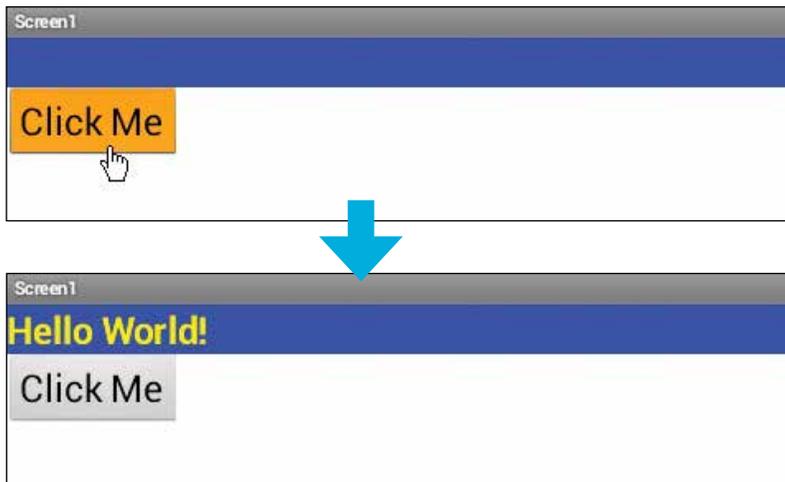
...cont'd

- 3 Back on your Android device, launch the MIT AI2 Companion and see it request you type in a handshake code
- 4 Precisely enter the 6-character code generated by App Inventor
- 5 Tap the “connect with code” button to see the app that is loaded in App Inventor run in your Android device



The computer running App Inventor in a browser must be connected to the same WiFi network as the Android device running the MIT AI2 Companion.

- 6 Tap the Button component in the Hello app running on the Android device to see the behavior defined in the Blocks editor apply the Label text greeting message



The running app can be updated live – try changing the label color in App Inventor’s Designer mode to see it immediately change in the Android device.

# Testing by emulator

As an alternative to live testing apps with an actual Android device, an Android device emulator can be used on your computer. This requires that **aiStarter** be installed on your computer from [appinventor.mit.edu/explore/ai2/setup-emulator.html](http://appinventor.mit.edu/explore/ai2/setup-emulator.html) Once installed, the aiStarter program typically starts automatically when you start your computer, or it can be started manually.

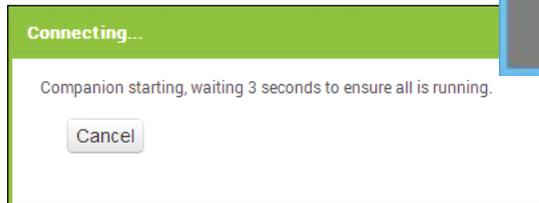
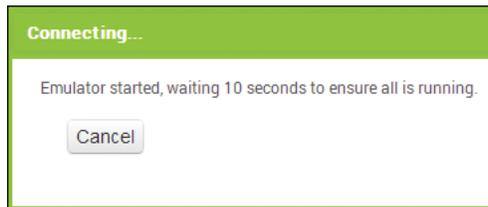
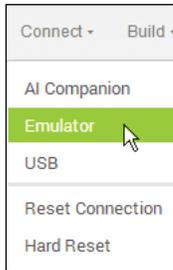
- 1 Ensure the aiStarter program is running – on Windows you will see a command window appear like this one



- 2 With the Hello app open in App Inventor choose Connect > Emulator on the menu to see the emulator app start up followed by an AI Companion app start up

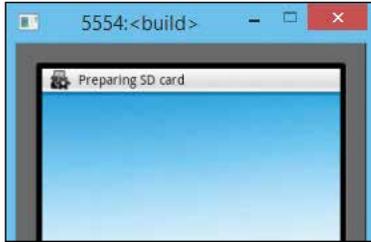


The AI Companion app here gets automatically installed when the aiStarter gets installed on the computer.

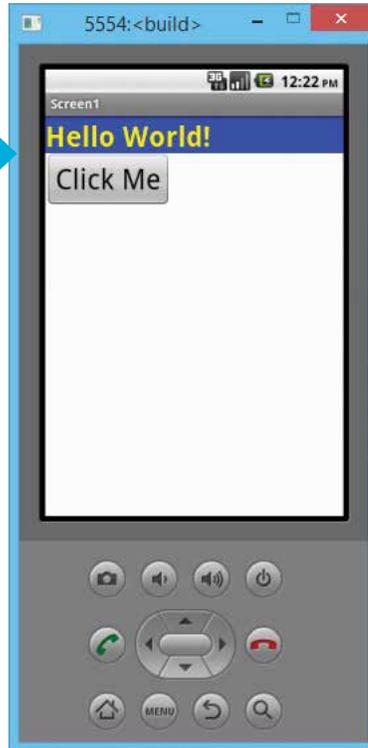
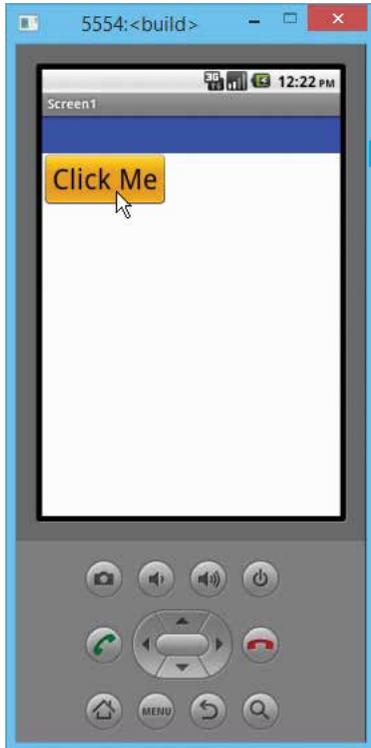


...cont'd

- 3 Wait until the initial black screen of the emulator changes to color, then wait until the emulator's SD card loads
- 4 Continue to wait while the AI Companion loads the Hello app from App Inventor



- 5 Click the Button component in the Hello app running on the emulator to see the behavior defined in the Blocks editor apply the Label text greeting message



Starting the emulator can take a couple of minutes but you should simply wait until your app appears in the emulator screen – don't use the emulator before.



Help for trouble shooting connection problems can be found online at [appinventor.mit.edu/explore/ai2/connection-help.html](http://appinventor.mit.edu/explore/ai2/connection-help.html)

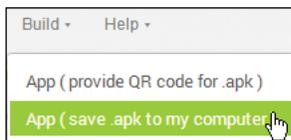
# Installing your first app

When you have completed building an app in App Inventor it can be compiled into an APK Android package and downloaded for direct installation (“side loaded”) onto any Android device.



Hello.apk

- 1 With the Hello app open in App Inventor choose the Build > “App (save .apk to my computer)” menu option



- 2 Wait while App Inventor compiles the project then downloads the APK package to your computer

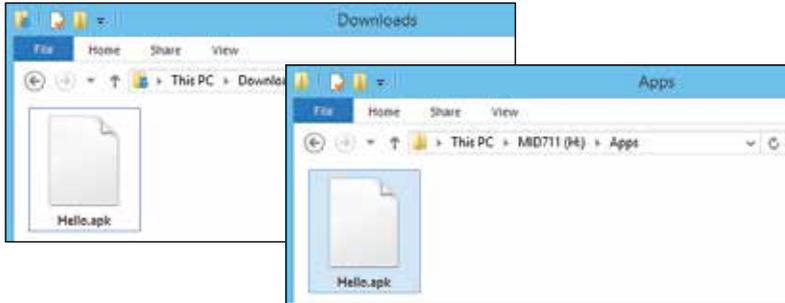


- 3 Next, connect an Android device to one of your computer’s USB ports and turn on USB storage



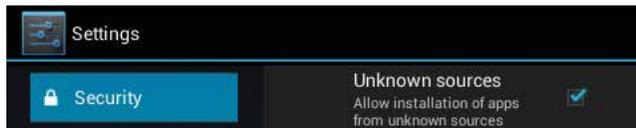
...cont'd

- 4 Now, copy the APK package from the default download folder on your computer to a convenient folder on the connected Android device

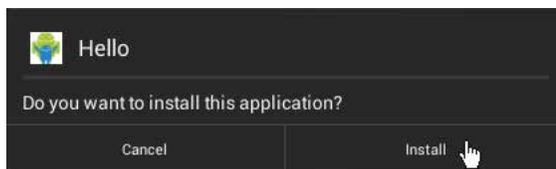


Allowing a connected Android device to act as USB storage means you can access folders on the device's SD card using the File Explorer on the connected computer.

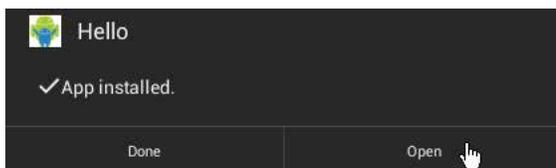
- 5 Ensure that the Settings on the Android device allow "installation of apps from unknown sources"



- 6 Tap the Hello.apk package icon on the Android device to call up the Android installer, then tap Install



- 7 When installation completes, tap Open to launch the application or tap the Hello icon that has been added



The Unknown Sources option on the device's Settings must be enabled to allow installation of apps from outside the Google Play Store.

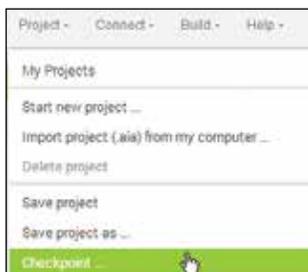


Automatic saving also enables changes to be instantly implemented during live testing.

# Managing projects

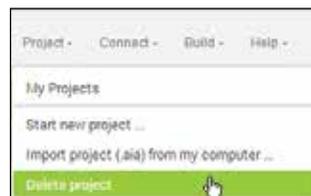
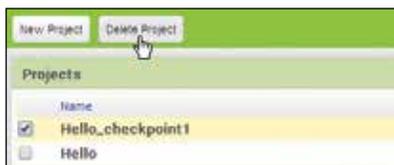
The “Projects” menu on the App Inventor menu bar contains a number of useful options for managing your application projects. Whenever you make any change to an app you are developing, App Inventor automatically saves the project, ensuring it is constantly up-to-date – so the “Save project” option is seldom needed. Its “Save project as” option is useful to create a new copy of the project, and you then continue working on that new copy. Its “Checkpoint” option, on the other hand, is useful to create a new backup copy of the project, and you then continue working on the original project:

- 1 With the Hello project open, click on the Project > Checkpoint option and accept the suggested default name to create a backup of this project



As you develop more applications with App Inventor each project gets automatically saved in the list on the “My Projects” page. While in App Inventor’s Projects mode, projects can be added to the list at any time using the “New Project” button or “Start new project” option, and existing projects can be deleted using the “Delete Project” button or “Delete project” option:

- 2 In the My Projects list, check the box against the Hello project backup then click the Delete Project button or choose the Projects > “Delete project” option to remove it



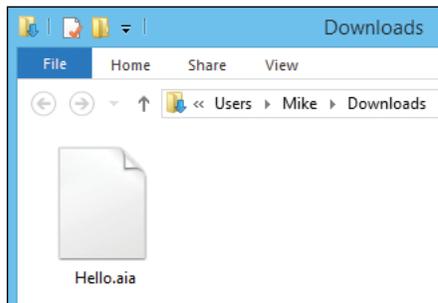
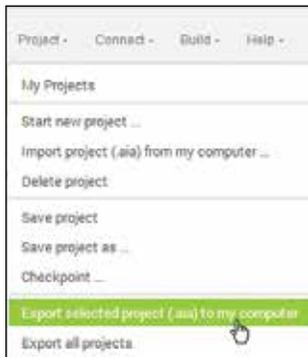
The New Project and Delete Project buttons are only accessible on the My Projects page, but the Start new project and Delete project options are accessible on the Projects menu while in Designer mode or Blocks editor mode.

...cont'd

Besides sharing your apps with others in an executable form as an Android package (.apk) for installation on an Android device, you can share your apps in source code form as an App Inventor Archive (.aia) that can be loaded into App Inventor and remixed.

3

In the My Projects list, check the box against the Hello project then choose the Projects > “Export project (.aia) to my computer” option to download the project source code

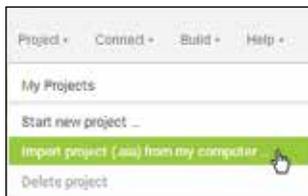


4

Delete the Hello project from the My Projects list using either of the methods described in Step 2

5

Click Projects > “Import project (.aia) from my computer” then choose the downloaded AIA source file to restore the project to the My Projects list



App Inventor 2 compresses resources into a single AIA archive – unlike the previous version that compressed them into a ZIP archive. Both archive types can be explored with a file archiver such as the free 7-Zip tool from [7-zip.org](http://7-zip.org)



The AIA sources of each example in this book are contained in the free ZIP archive described on page 6, so you can upload and remix them.

# Summary

- Android is an operating system for mobile phones and tablets
- App Inventor enables you to develop applications for Android devices without writing a single line of code
- MIT App Inventor 2 runs completely in a web browser
- App Inventor first opens on the Projects page where new projects get started
- The App Inventor Designer mode is where you add components to the application to design the user interface
- Designer mode contains Palette, Viewer, Components, Media, and Properties columns
- Interface components can be dragged from the Palette onto the Viewer and their characteristics edited in Properties
- The App Inventor Blocks editor mode is where you add functionality to the application to create behaviors
- Blocks editor mode contains a Blocks column, listing drawers that each provide blocks of code, and a Viewer area
- Blocks can be dragged from the Blocks column and snapped together on the Viewer area to form instructions
- MIT AI2 Companion enables an Android device to be used for live testing an app via a shared WiFi network connection
- Installation of aiStarter enables a virtual device emulator to be used to test an app on a single computer
- Completed apps can be compiled as an APK package for download and then be side loaded onto Android devices
- Project source code can be downloaded as an AIA archive for sharing and be uploaded for remixing
- Projects are managed from the My Projects list view