Worksheet 2 (part A)

AppInventor Environment

Activity: You will be introduced to the main parts of AppInventor.
Time: 10 Minutes

Sign in with your Google account at http://ai2.appinventor.mit.edu

In all the App Inventor apps you'll build, you'll use the same three steps with three different screens:

1. **Design** the app screen by using the *app Designer*.
2. **Tell** the app **what to do** by programming the *Blocks Editor*.
3. **Test** the program using your *phone* or *emulator*.

Designer

The AppInventor Designer lets you

a) *Create* a new project,

b) *Add* components and

c) *design* the look of the app.

Blocks Editor

- The Blocks Editor lets you *control* how the app works by using *programming blocks*. (You can toggle between Designer and Blocks as you program.)

Testing[*]

- Finally, you'll test the program on your phone.
- You can see if the app looks and works OK.

[*] Testing on your smartphone

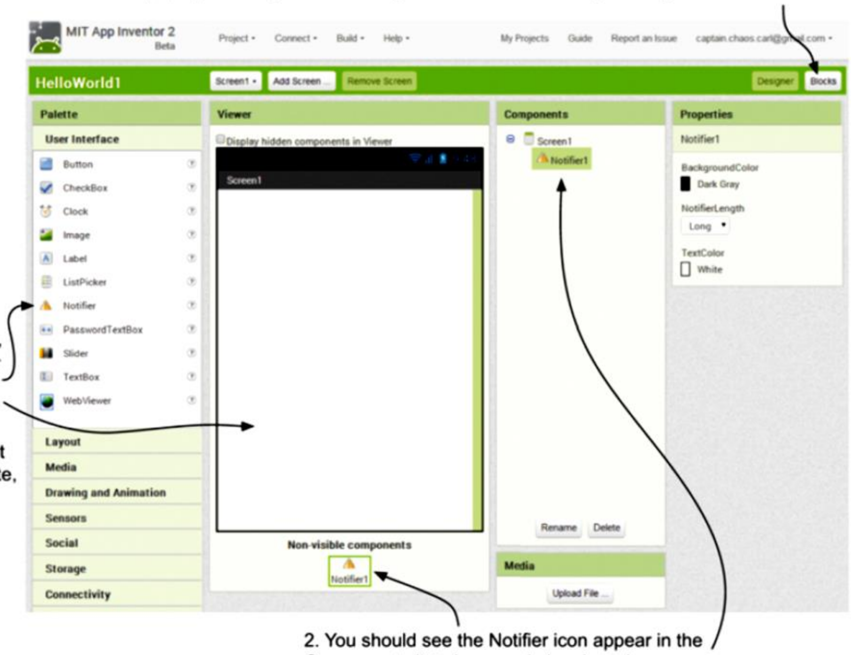
- You will need an Android smartphone to download from Google Play the "MIT AI2 Companion"

Worksheet 2 (part B)

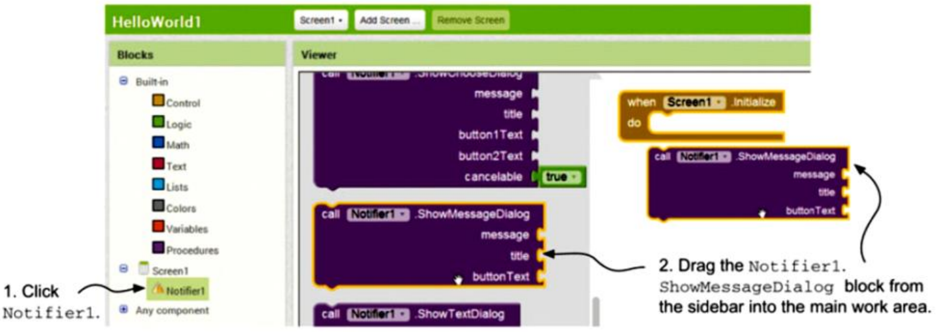
My first app: Hello World!

Activity: You will create an app that displays a pop-up message saying “Hello World!”
Time: 30 Minutes

Follow your facilitator and complete the tasks below (put a mark if completed).
Don’t hesitate to ask if there is something you are not sure of.

TASK	DONE?
Starting a new project 1. From the menu click “Project” and then choose “Start New Project”. 2. Enter the project name, `HelloWorld`, and then click OK.	
TIP: • No spaces are allowed in project names or component names. One way to separate words is to use *camel case*, i.e. capital letters to indicate the start of each word. • App Inventor projects are automatically saved every few seconds as you work on them.	
Adding a Notifier component to the project  You can use the useful `Notifier` component for all kinds of pop-up messages and warnings. 3. You’re finished in the Designer, so open the Blocks Editor by clicking the Blocks button. 1. From the Palette, drag the `Notifier` component onto `Screen1` here. Tip: If you can’t spot Notifier in the Palette, make sure you’re looking in the User Interface section. 2. You should see the Notifier icon appear in the Components list above and also down here. MIT App Inventor 2 Beta Project Connect Build Help My Projects Guide Report an Issue HelloWorld1 Screen1 Add Screen Remove Screen Designer Blocks Palette User Interface Button CheckBox Clock Image Label ListPicker Notifier PasswordTextBox Slider TextBox WebViewer Layout Media Drawing and Animation Sensors Social Storage Connectivity Viewer Display hidden components in Viewer Screen1 Non-visible components Notifier1 Components Screen1 Notifier1 Rename Delete Media Upload File ... Properties Notifier1 BackgroundColor Dark Gray NotifierLength Long TextColor White	

Writing the program using blocks

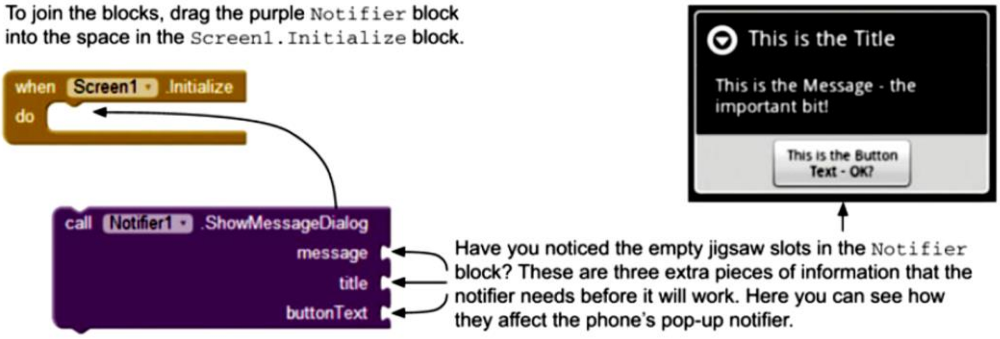


1. Click Notifier1.

2. Drag the Notifier1. ShowMessageDialog block from the sidebar into the main work area.

Now you join the blocks

To join the blocks, drag the purple Notifier block into the space in the Screen1.Initialize block.



This is the Title

This is the Message - the important bit!

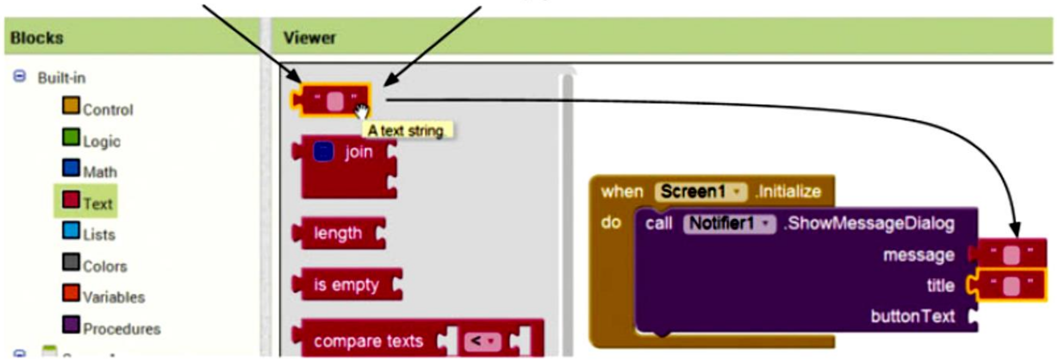
This is the Button Text - OK?

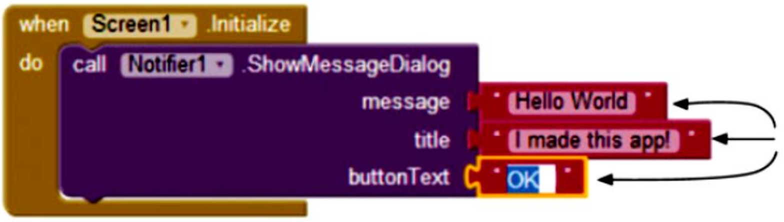
Have you noticed the empty jigsaw slots in the Notifier block? These are three extra pieces of information that the notifier needs before it will work. Here you can see how they affect the phone's pop-up notifier.

The message, title, and button text that the notifier needs are all pieces of text—in programming, we call them *text strings*. To make text strings, you use a Built-in Text block.

1. Click the empty Text block from the Built-in blocks.

2. Drag three empty Text blocks into the three empty Notifier1 slots.





You can set a text string by clicking between the quote marks.

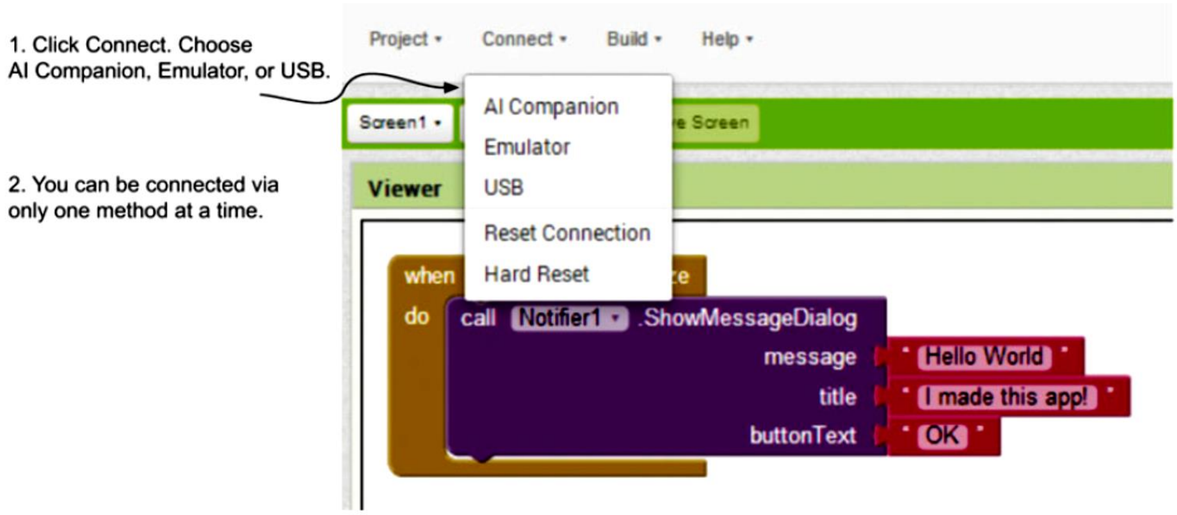
Change the three text strings to what is shown here.

TIP:

Before testing the app you should download the “MIT AI2 Companion” from Google Play.

Testing the app

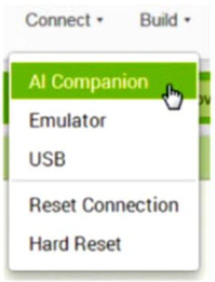
Connect your smartphone or tablet from the Connect menu at the top of your screen.



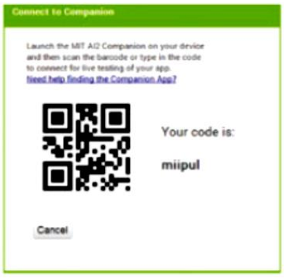
1. Click Connect. Choose AI Companion, Emulator, or USB.

2. You can be connected via only one method at a time.

- Connect your smartphone on the Wi-Fi network.
- Choose AI Companion. A code will appear on your computer screen. You can enter or scan this code into your phone by running the AI Companion phone app.



1. Choose AI Companion from the Connect menu.



Connect to Companion

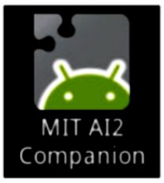
Launch the MIT AI2 Companion on your device and then scan the barcode or type in the code to connect for live testing of your app.
Need help finding the Companion App?

Your code is:

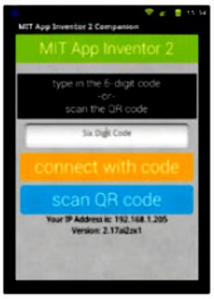
miipul

Cancel

2. A unique app code appears in both QR and text form.



3. On your phone, start the MIT AI2 Companion app.



4. Either type the app code and click Connect with Code or click Scan the QR Code and point your phone at the QR code on your computer screen.

(Source: Book “Hello App Inventor!” Beer P., Simmons C., Publications *Manning Shelter Island*)

Worksheet 2 (part C)

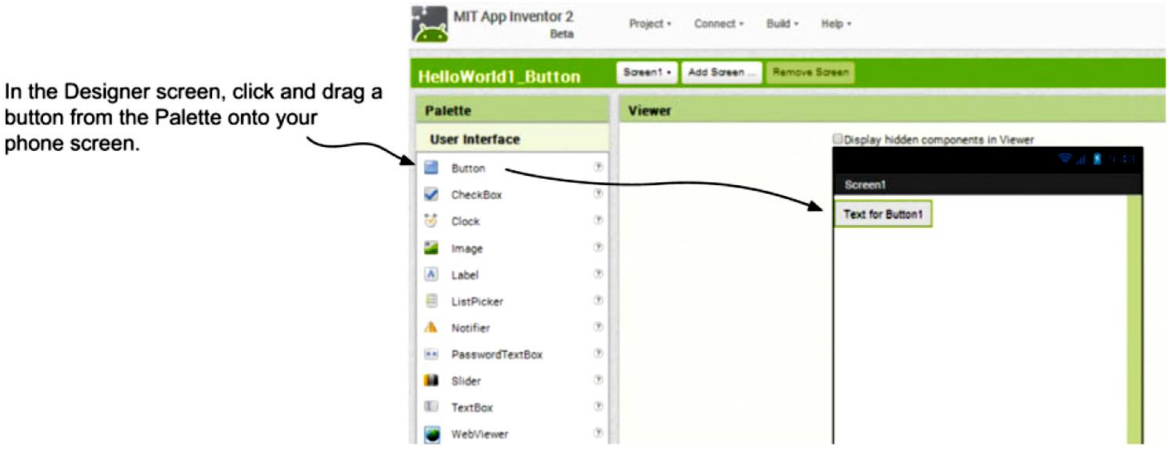
Adding functionality: Hello World! App, version 2

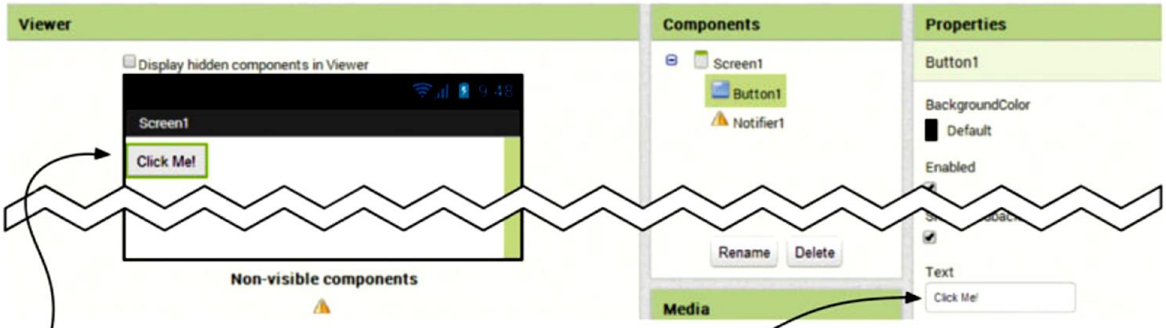
Activity: You will add a button to a new copy of your app.
Time: 15 Minutes

The problem in the Hello World! App is that the `Screen1.Initialize` event handler you use to trigger the app only happens once: when the app first runs.

It would be better to have an event like a *button press* that the user can touch to trigger the event handler whenever they want to see the message.

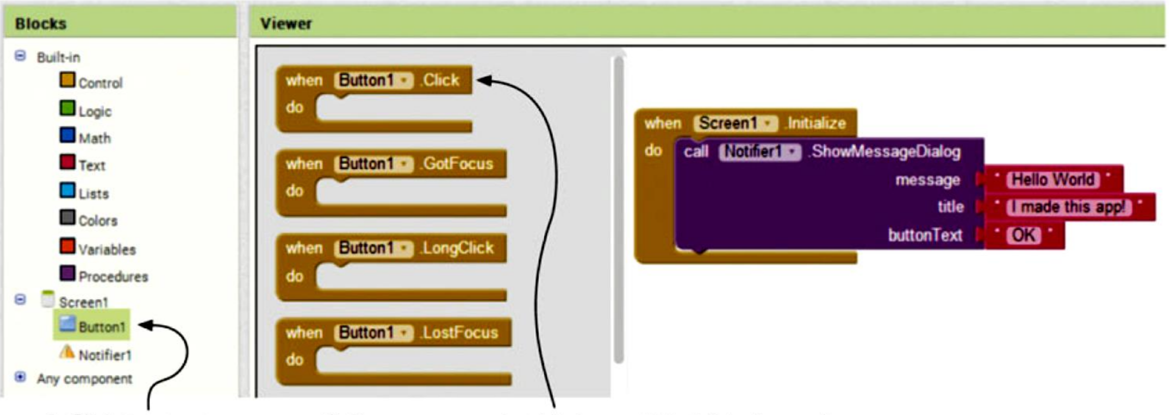
Follow your facilitator and complete the tasks below (put a mark if completed).
Don't hesitate to ask if there is something you are not sure of.

TASK	DONE?
1. Saving a new copy of Hello World! You call the new project `HelloWorld1_Button` so that when you look at the list of projects, you know what makes this version different from any others. Giving items sensible names will help you later when things get more complicated and you've created lots of different projects. 1. Menu Project → Save As… 2. Call your new project `HelloWorld1_button` 3. Click OK.	
2. Adding a Button component  In the Designer screen, click and drag a button from the Palette onto your phone screen. MIT App Inventor 2 Beta Project ▾ Connect ▾ Build ▾ Help ▾ HelloWorld1_Button Screen1 ▾ Add Screen ... Remove Screen Palette User Interface Button CheckBox Clock Image Label ListPicker Notifier PasswordTextBox Slider TextBox WebViewer Viewer □Display hidden components in Viewer Screen1 Text for Button1 You need to tell the user what to do with the button. Change the button to say `Click Me!`	



You can change the text that appears on the button by changing its `Text` property here.

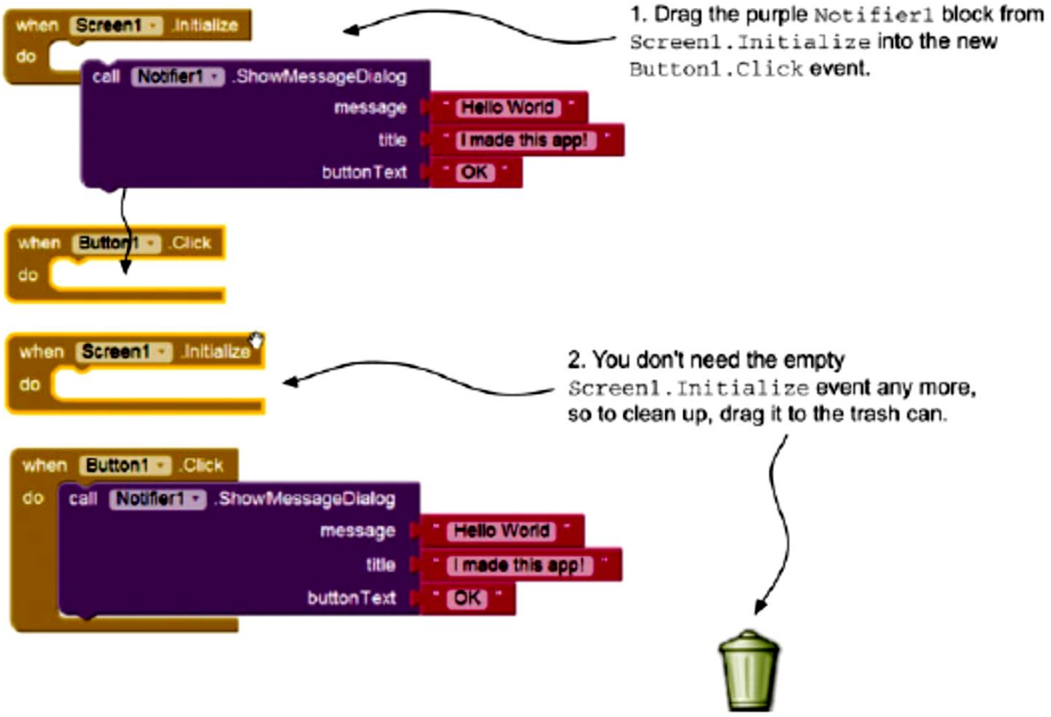
3. Programming the blocks



1. Click `Button1`.

2. Drag a `Button1.Click` event block into the work area under your `Screen1.Initialize` code block.

Now drag the **Notifier** block's code from `Screen1.Initialize` into `Button1.Click`:



1. Drag the purple `Notifier1` block from `Screen1.Initialize` into the new `Button1.Click` event.

2. You don't need the empty `Screen1.Initialize` event any more, so to clean up, drag it to the trash can.

4. Testing your app Connect your smartphone or tablet by choosing from the Connect menu “AI Companion” at the top of your screen.	
Taking it further **1.** Add a new button and notifier to your app. Make the notifier say “Goodbye World” if the user clicks the second button. **2.** Try changing the color properties of **Screen1** and **Button1**. Also change the **Font** and **Size** properties of `Button1`.	
Don’t forget to Sign out from AppInventor once you’re finished with your work.	

(Source: Book “Hello App Inventor!” Beer P., Simmons C., Publications *Manning Shelter Island*)