

Activity 1.1.2 Digital Doodle

Introduction

Let's create an app! In this app, you'll be able to take a photograph with the press of a button and then draw on it.

You will use MIT App Inventor to create **event handlers**. An event handler executes a set of computer program instructions when a certain event happens. Examples of events are the user pressing a button or a text message being received. You could make an app that texts someone automatically when you get to school—it's just a matter of creating an event handler.



When _____ event _____,
do _____ event handler _____.

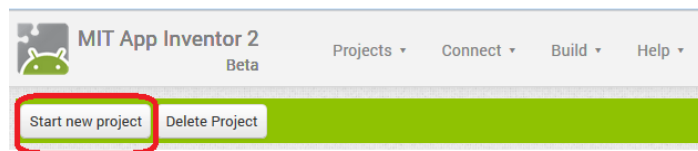
Materials

- Computer with browser
- Android device with AI Companion
- Google ID
- 1.1.2 sourceFiles.zip

Procedure

1. Form pairs as directed by your teacher. Practice a professional greeting that includes shaking hands and saying something enthusiastic while smiling and making eye contact.
2. Open MIT App Inventor by following these steps.
 - a. App Inventor **authenticates** you by asking Google to do so. **Sign in** to a Google account at <https://google.com>. Before typing a username or password in a browser, always check the schema (HTTPS) and domain name (google.com).
 - b. Navigate to the App Inventor website <http://ai2.appinventor.mit.edu> and **allow** Google to provide your user name to App Inventor. **Continue** past the splash screen(s).

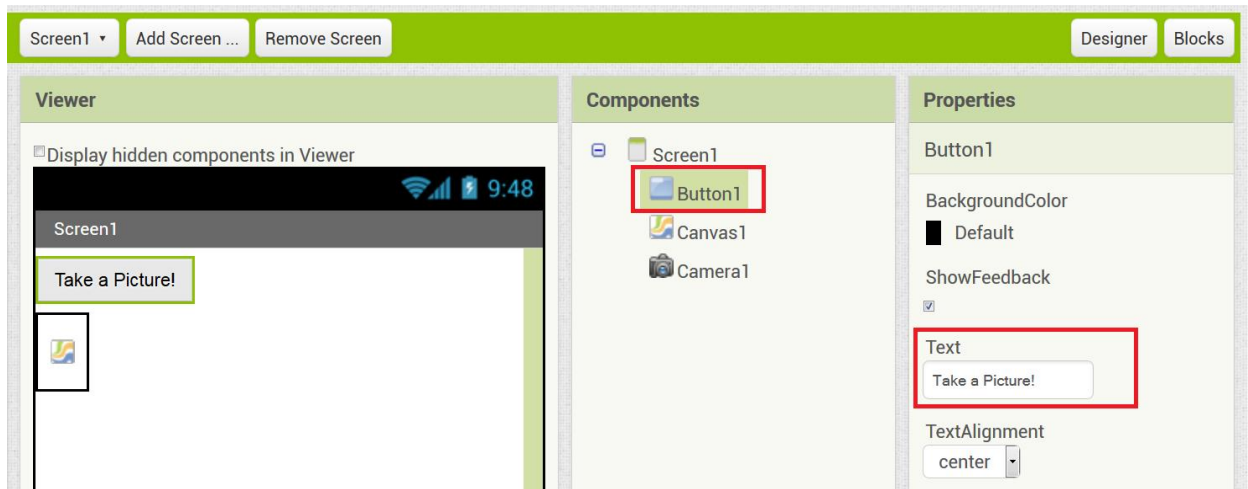
3. You will be creating Digital Doodle in one of two ways, as directed by your teacher. One way is to start with a new App Inventor project; the other way uses some starter code provided with this activity.
 - a. If you will be starting with a new App Inventor project, skip ahead to step 4.
 - b. If you will be using starter code provided with this activity, continue here.
 - i. Download the source files for this activity from the activity's resources.
 - ii. Unzip the file. To unzip a file on a typical computer, **right-click** the file and select **7zip > Extract files**. Save the extracted files in a new folder you create for this project.
 - iii. In App Inventor, select **Projects > Import project (.AIA) from my computer**.
 - iv. Jump ahead to step 8, "Test your program by..."
4. Select **Start New Project**. Name the project as directed by your teacher and create a project folder for your work as directed by your teacher.



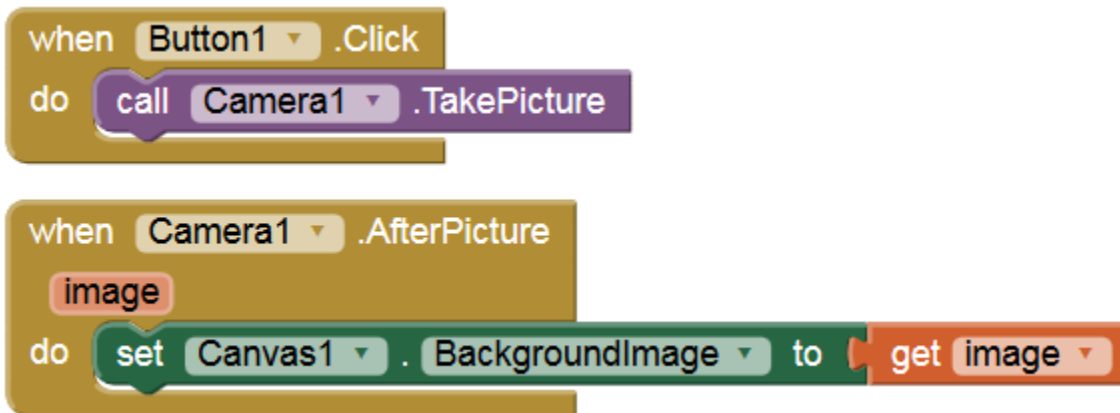
5. Drag three components into your app from the drawers listed here. The camera is a nonvisual component, which means it will appear at the bottom of the UI designer window when you drag it to the viewer.

Component	Designer Drawer	Purpose
Button	User Interface	Respond to being clicked
Canvas	Drawing and Animation	Provide a touch-sensitive surface where you can draw and animate
Camera	Media	Access the device's camera

6. Select the button in your Viewer or Components list. Change the button's Text property to "Take a Picture!"

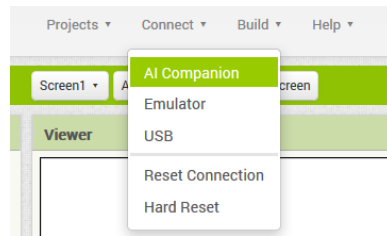


- Switch to the **Blocks view** by clicking the Blocks button in the upper-right corner. Create this script. The following table shows you where to find each block.

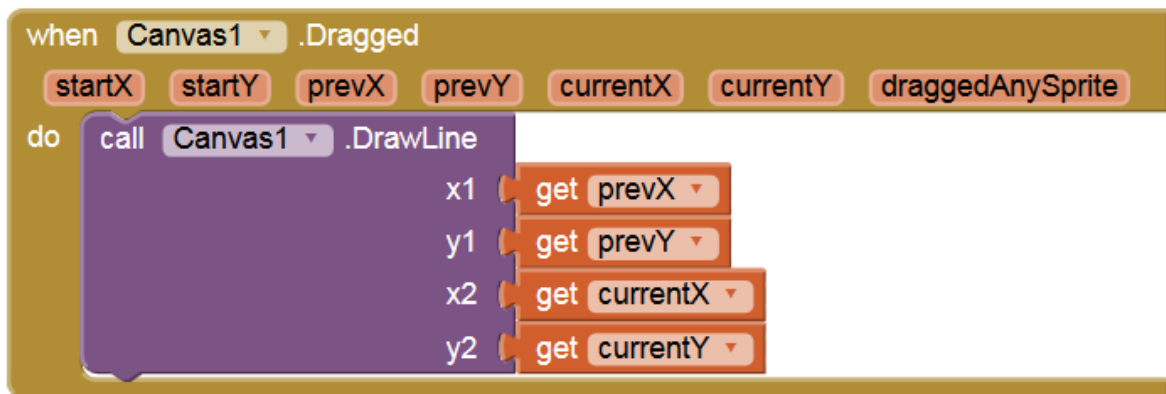


Block	Blocks Drawer	Purpose
	Camera drawer	Lets the user take a picture
	Camera drawer	Event handler! This script will execute after the user takes a picture.
	Button drawer	Event handler! The script inside this block will be executed when the user releases a button.
	Canvas drawer	Sets the background image of the canvas.
	mouse over on the camera's "After picture" block	Refers to the picture just taken. Only available locally, within the "After picture" event handler.

8. Test your program by connecting to the device with the following steps:
 - a. In the App Inventor browser window, select **Connect > AI Companion**.



- b. On the Android device, launch the **AI2 Companion** app. The browser should display a six-character code.
 - c. Enter the six-character code in the Android device's AI2 Companion app and select **Connect with code**. Alternatively, use the QR code.
9. To be able to draw on your picture, create the following event handler:



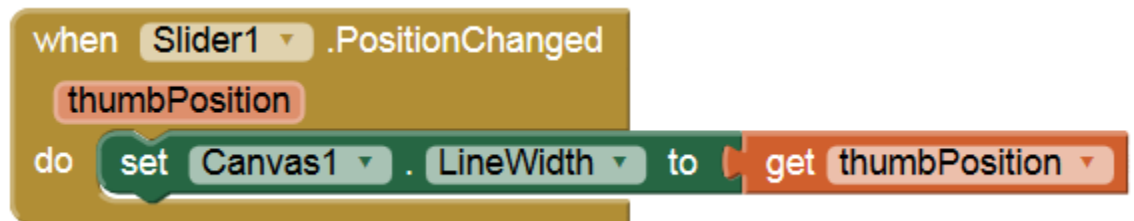
10. Add one or more of these features to your app. Implementing each feature requires some change in the Designer view and some change in the Blocks view. Some guidance is provided below for the Designer View for each feature, followed by a picture of the blocks needed. You will switch between Designer and Blocks as you go through these steps.

- a. Add a slider to adjust the pen size that draws as you drag your finger on the touch screen.

The following image shows a Slider component alongside a Label component. The two components have been placed inside a Horizontal Arrangement component.

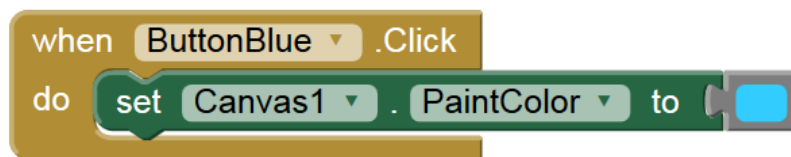


Component	Designer Drawer	Purpose
Label	User Interface	Display text
Slider	User Interface	Change a value as the user slides a control back and forth
Horizontal Arrangement	Layout	Center a component or place components side by side



b. Add buttons to change the color of the pen.

Here we've add three buttons, placed them inside a Horizontal Arrangement, and set the Shape and BackgroundColor property of each button. In the Components list of the Designer view, we've also renamed each button to a name like "ButtonBlue." That helps the programmer keep track of the components.

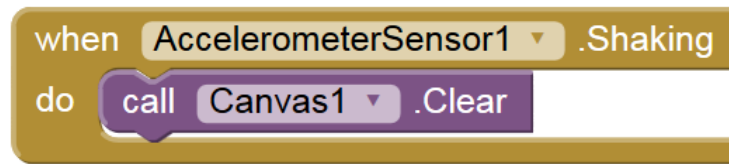


c. Clear the canvas when you shake the tablet.

Add an accelerometer sensor component in the Designer view.

Component	Designer Drawer	Purpose
Accelerometer	Sensors	Detect when the tablet is shaken. Also

Sensor		detects when the tablet is tilted forward or left/right.
--------	--	--



11. Save your work in App Inventor. Provide evidence of your work to your teacher, as directed. Some methods for sharing your work with your teacher:
 - a. Show your App Inventor screen.
 - b. Demonstrate the app on a tablet using AI Companion.
 - c. Download the App Inventor program by selecting **Projects > Export selected project (.AIA) to my computer**.
 - d. Download the Android app by selecting **Build > App (save .APK to my computer)**.

Conclusion Questions

1. What is an event handler? Describe how you create an interface with event handlers in App Inventor.
2. Think of some software you have used. This could be an application on a mobile device or any other computer. Describe an event that the software responds to, and describe what the event handler of that software does.