Activity 1.1.3 Count App

Introduction

How do computers work?

A computer—whether a laptop, a smartphone, or a computer under the hood of a car—follows one instruction at a time. Each instruction tells a **processor** to do one simple task such as "add two numbers together." Programs are lists of those instructions, one instruction after another.

Programs also define new instructions that are more **abstract**, such as "count the number of people in a picture."

How are programs created?

A software developer uses a computer language to write a program, listing instructions for the computer to follow. The program might respond to **input** such as the keyboard, mouse, or touch screen. The program might produce **output** such as sound, graphics, or motor movement.

Materials

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

Procedure

1. At the retail store, Jennifer sometimes has to take inventory, counting how many of each item are on the shelves. She has to write down or type in the number in stock of each item, but she needs her hands free to climb ladders for the high shelves and to be able to move things around to count them.

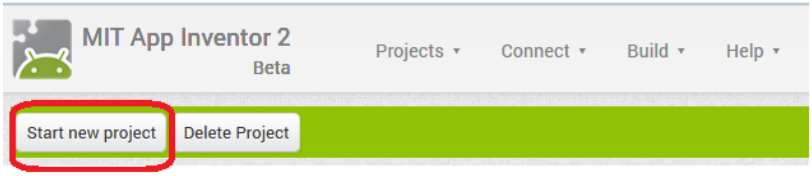
   Paul sells tickets for some of the school's sporting events. He also is on the athletics committee that sets the various prices for the tickets. The committee is discussing prices for some of the games that have free attendance. Paul would like to be able to quickly count attendees at the free games so that he can bring this information back to the committee to inform the discussion.

2. You will be creating your own tally counter. To create the tally counter, you will learn how to **decompose** a problem into smaller steps. Problem decomposition is a central concept in learning to program.

Overview
1. Person tallying wants to increment a counter so they can count people or things without keeping track of the total so far. 2. Person tallying wants to count several tallies at once so they can compare the counts of different items being tallied. 3. Person tallying wants to control the tally counter by voice so that they can have both hands free. 4. Person tallying wants the tally to accumulate automatically so they don't have to pay close attention to the device's screen. 5. Person tallying wants text message or email alerts when various counters reach certain thresholds so they can respond with action.

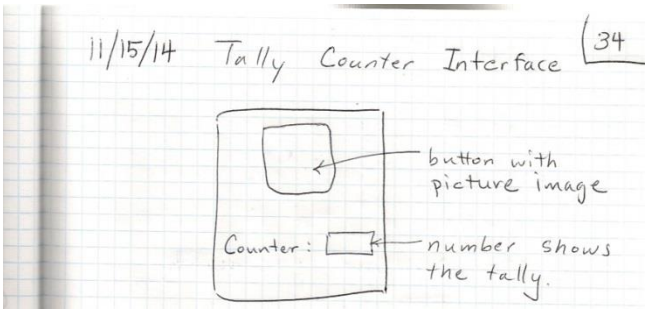
Task List
• When the user presses a button, add 1 to a counter. • Make the interface attractive with an image for the button, some color, and a label for the counter. • Make a sound when the user presses a button so they know it has been pressed. • When the user presses a reset button, make the counter 0 (zero). • A checkbox can enable or disable whether the tally button works.

3. 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. Alternatively, as directed by your teacher, upload countMeIn.aia and skip to Step 7.



4. In this step, you will create the interface in the Designer view. In the next step, you will add functionality in the Blocks view.

   One of the students sketched the **interface** she wanted. Her sketch is shown below. Create the interface with the following steps as described in the video on the basic tally app.



   a. Drag the components (a button and two labels) onto the interface from the **User Interface** drawer. Place the two labels inside a horizontal layout component from the **Layout** drawer. Rename the components to say what type of component they are and why they are there.

Component	Name
Button	`Button_Tally`
Label	`Label_CounterDescription`
Label	`Label_CounterNumber`

   b. Set some of the initial values of the components' **properties** using the Properties panel in the Designer view. The following properties will initialize the text of the labels and to initialize the size of the button.

Component Name	Property	Value
`Label_CounterDescription`	Text	`Counter #1:`
`Label_CounterNumber`	Text	`0`
`Button_Tally`	Width	`200`
`Button_Tally`	Height	`200`

   c. Make the button display an image as follows.

i. Upload an image file using the Media panel. The image countMeIn.JPG is on the shared (S:\) drive, or you may use an image of your choice.

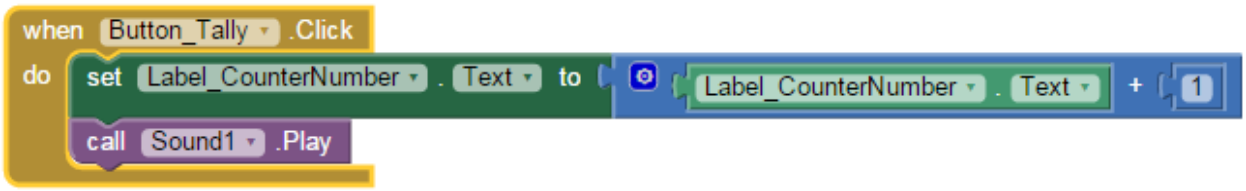
ii. Select the button in the Components panel and then in the Properties panel, set the button's image property to use that file.

d. Make a sound component to play a clicking sound when the button is pressed.
   i. Drag a sound component from the Media drawer into the interface Viewer. The component will appear at the bottom of the viewer as a non-visible component.

   ii. Upload a short sound file using the Media panel. The sound click.MP3 is provided in the shared (S:\) drive, but you can use your own short sound if you wish.

   iii. Select the sound component in the Components panel and then in the Properties panel, set the source to your sound file.

5. In the Blocks view, create an event handler for when the button is clicked. The handler should play the sound and increment the tally when the button is clicked.



6. Test your app by connecting to the AI2 Companion with the following steps.

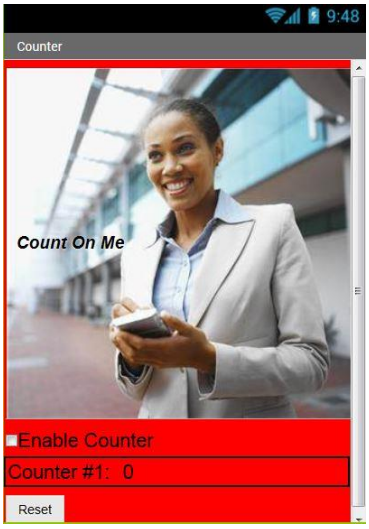
   a. On the Android device, launch the AI2 Companion.

   b. In App Inventor in the browser, select **Connect** > **AI2 Companion**.

   c. Use the QR code or six-letter passcode to connect.

7. Create solutions for the following challenges. (REQUIRED)

   a. Add a reset button. When the user presses the reset button, make the counter go back to zero.



b. The user wants to be able to carry the Android device around as they're tallying, but doesn't want to accidentally tally while holding the tablet when walking. Add an "enabled" checkbox so that the user can enable or disable whether the tally button works. You will use the `if-then` block as shown here.