

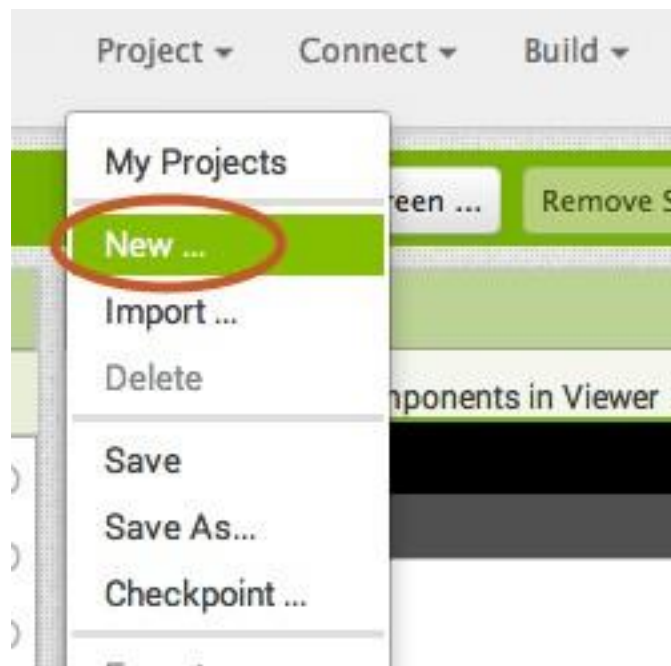
---

## DigitalDoodle: Drawing App

This tutorial will show you how to draw a line on the screen as the user drags a finger around.

### Start a New Project

From the My Projects page, click New Project. If you have another project open, go to My Projects menu and choose New Project.



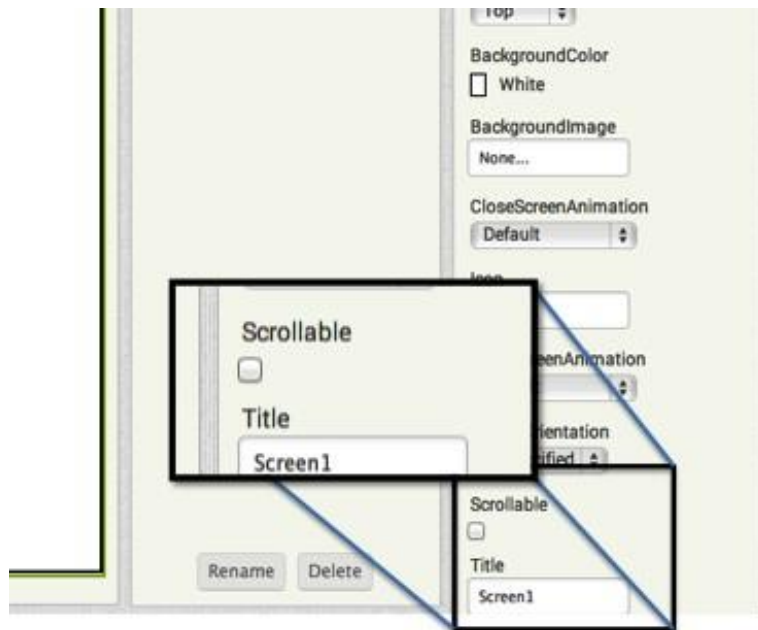
### Name the Project

Call this project DigitalDoodle, or create your own name for this drawing app.



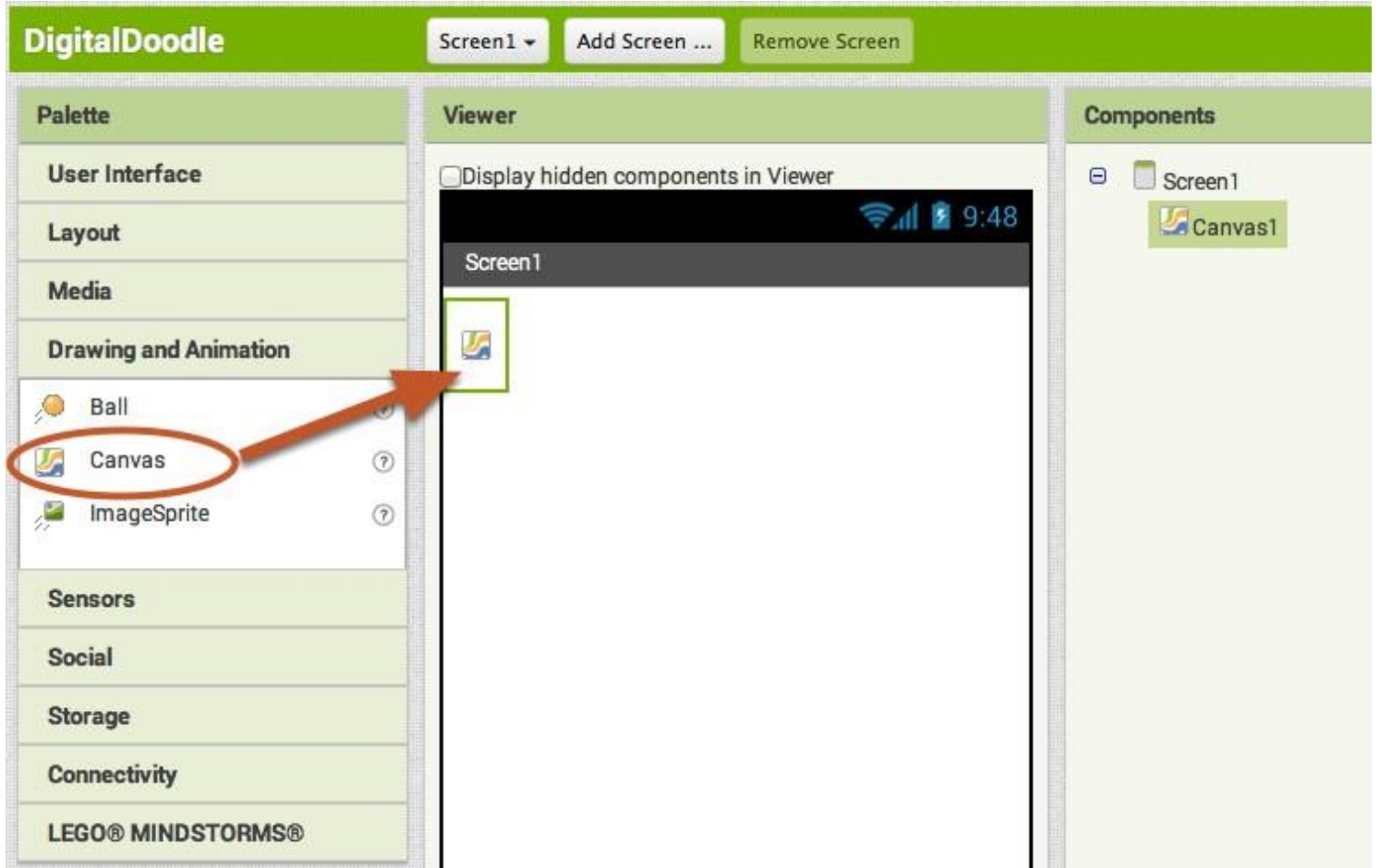
## Set the Screen so that it does not scroll

The default setting for App Inventor is that the screen of your app will be "scrollable", which means that the user interface can go beyond the limit of the screen and the user can scroll down by swiping their finger (like scrolling on a web page). When you are using a Canvas, you have to **turn off the "Scrollable" setting** (UNCHECK THE BOX) so that the screen does not scroll. This will allow you to make the Canvas to fill up the whole screen.



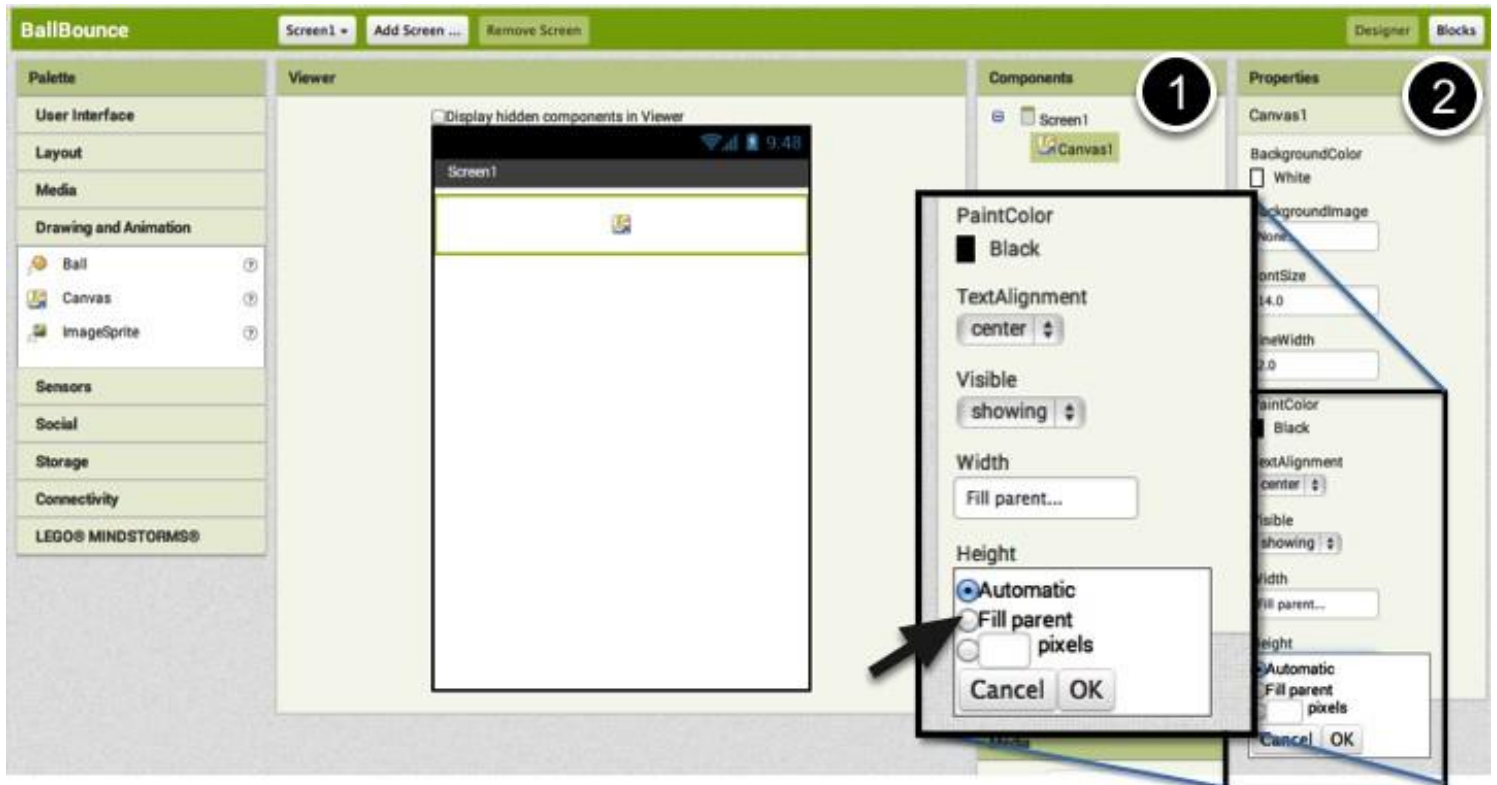
## Add a Canvas

From the Drawing and Animation drawer, drag out a Canvas component.



## Change the Height and Width of the Canvas to Fill Parent

Make sure the Canvas component is selected (#1) so that its properties show up in the Properties Pane (#2). Down at the bottom, set the Height property to "Fill Parent". Do the same with the Width property.



## That's all for the Designer! Go over to the Blocks.

Believe it or not, for now this app only needs a Canvas. Go into the Blocks Editor to program the app.



## Get a Canvas.Dragged event block

In the Canvas1 drawer, pull out the **when Canvas1.Dragged** event.

The screenshot shows the DigitalDoodle interface. The top bar is green and contains the text "DigitalDoodle" on the left, and "Screen1", "Add Screen ...", and "Remove Screen" on the right. Below the top bar is a "Blocks" panel on the left and a "Viewer" panel on the right. The "Blocks" panel is divided into "Built-in" and "Screen1" sections. The "Built-in" section lists categories: Control (orange), Logic (green), Math (blue), Text (red), Lists (light blue), Colors (grey), Variables (orange), and Procedures (purple). The "Screen1" section contains a "Canvas1" block, which is circled in red. The "Viewer" panel shows three event blocks: "when Canvas1.Dragged", "when Canvas1.Flung", and "when Canvas1.TouchDown". The "when Canvas1.Dragged" block is circled in red and contains the following parameters: startX, startY, prevX, prevY, currentX, currentY, and draggedSprite. The "when Canvas1.Flung" block contains parameters: x, y, speed, heading, xvel, yvel, and flungSprite. The "when Canvas1.TouchDown" block contains parameters: x and y.

## Get a Canvas.DrawLine call block

In the Canvas1 drawer, pull out the **when Canvas1.DrawLine** method block..

The image shows a programming environment with two main panels: 'Blocks' on the left and 'Viewer' on the right.

**Blocks Panel:**

- Built-in
  - Control
  - Logic
  - Math
  - Text
  - Lists
  - Colors
  - Variables
  - Procedures
- Screen1
- Canvas1** (highlighted with an orange oval)
- Any component

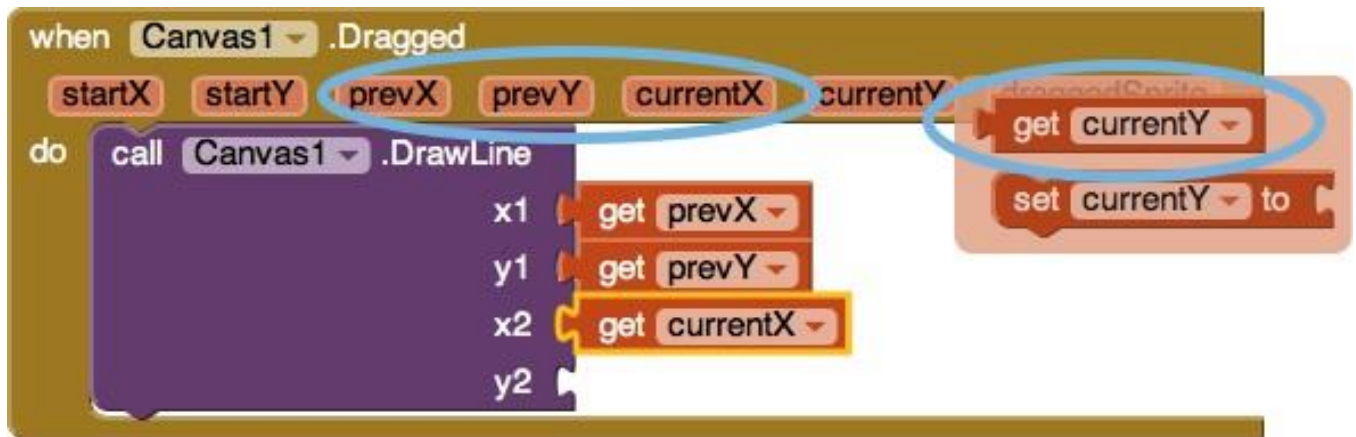
**Viewer Panel:**

The Viewer shows a sequence of code blocks:

- when Canvas1 .Touched
  - x y touchedSprite
  - do
- call Canvas1 .Clear
- call Canvas1 .DrawCircle
  - x
  - y
  - r
- call Canvas1 .DrawLine** (circled in orange)
  - x1
  - y1
  - x2
  - y2
- call Canvas1 .DrawPoint

## Draw Line block.

The Canvas Dragged event will happen over and over again very rapidly while the user drags a finger on the screen. Each time that Dragged event block is called, it will draw a small line between the previous location (prevX, prevY) of the finger to the new location (currentX, currentY). Mouse over the parameters of the Canvas1.Dragged block to pull out the get blocks that you need. (Mouse over them, don't click on them.)





## Test it out!

Go to your connected device and drag your finger around the screen. Do you see a line?

