SKETCH & GUESS: PART 3

SEQUENCE OF EVENTS

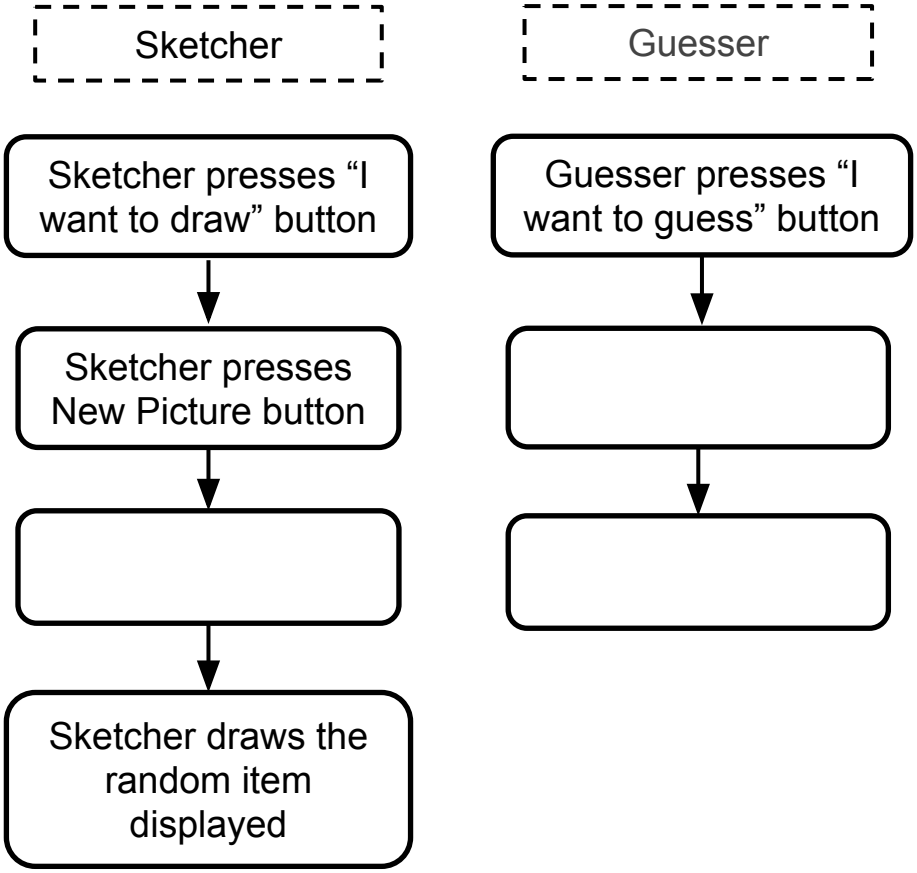
In Lessons 1 and 2, you built a Sketching app. To make it more fun, you will code it so you can draw something on your device and your partner can see the drawing on their device.

1 Review with your partner the diagrams below. Check that you understand the sequence of steps on the drawing part of the Sketch and Guess App. Fill in the empty spaces with A, B, or C.

A. Guess what it is

B. Show a random item to draw

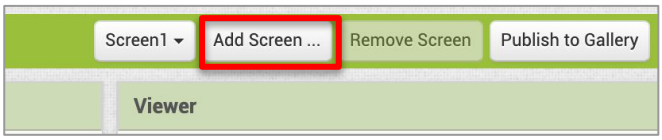
C. See what the Sketcher draws



ADD GUESSERSCREEN

To add the ability for the Guesser to see what is being drawn, you need to add the GuesserScreen to the app.

2 Open your SketchAndGuess project. Add a new screen and name it **GuesserScreen**.



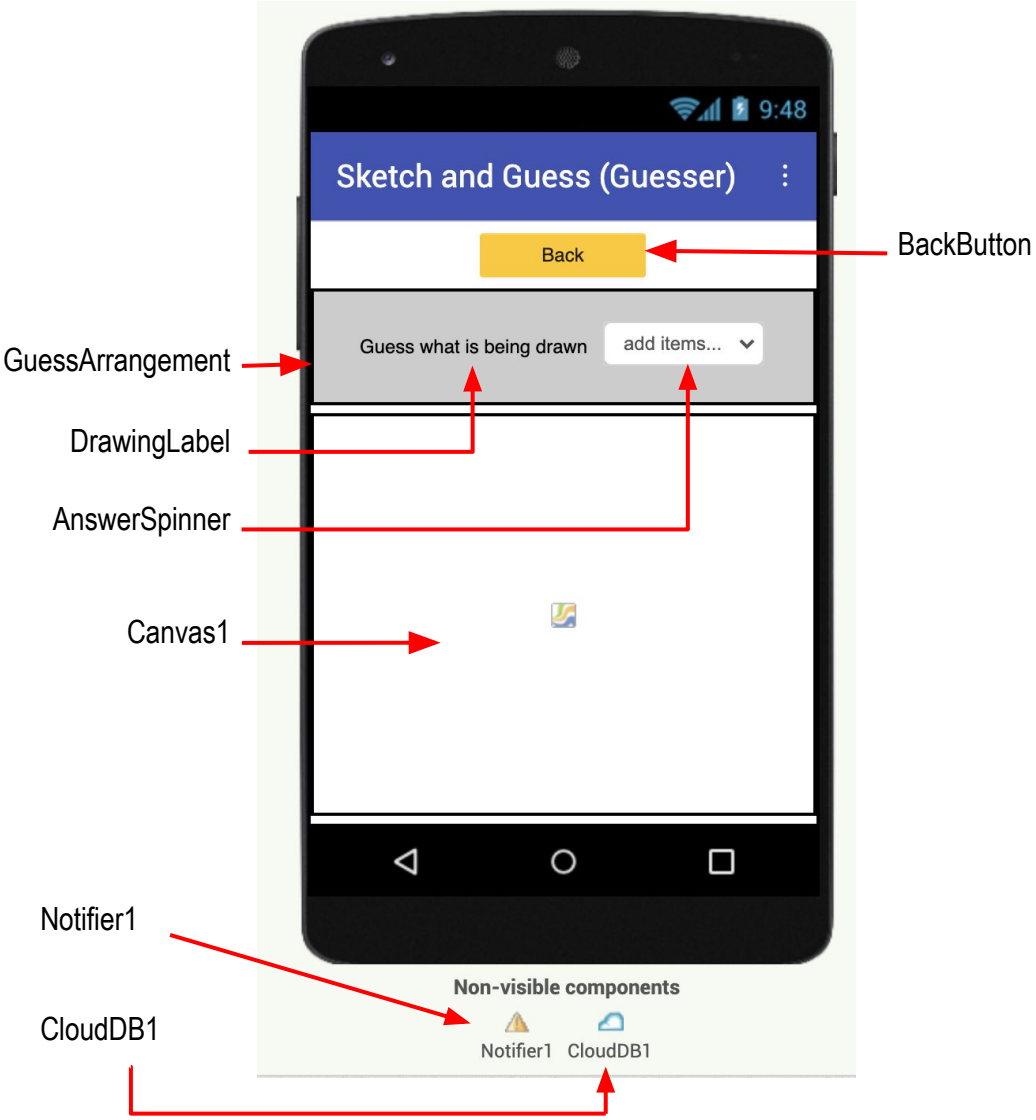
3 Set the GuesserScreen property *AlignHorizontal* to "**Center: 3**". Set the *Title* property to **"Sketch and Guess (Guesser)"**.

4 Add the following components to GuesserScreen.

Drawer	Component	Component Name	Properties
User Interface	Button	BackButton	*Width*: **33%**
			BackgroundColor: (choose a color)
			Text: "**Back**"
Layout	Horizontal-Arrangement	GuessArrange-ment	*Width*: **"Fill parent"**
			AlignHorizontal: **"Center: 3"**
User Interface	Label	DrawingLabel	*Text*: "**Guess what is being drawn**"
User Interface	Spinner	AnswerSpinner	*Prompt*: "**Guess the Drawing**"
Drawing and Animation	Canvas	Canvas1	*Width*: **"Fill parent"**
			Height: **"Fill parent"**
Storage	CloudDB	CloudDB1	
User Interface	Notifier	Notifier1	

ADD GUESSERSCREEN (continued)

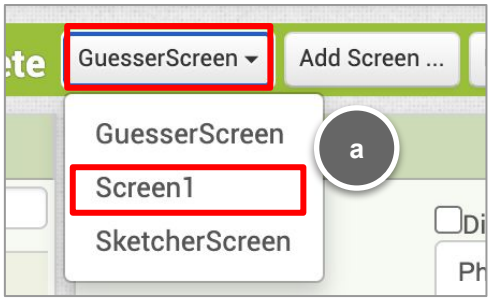
Your GuesserScreen should look something like this:.



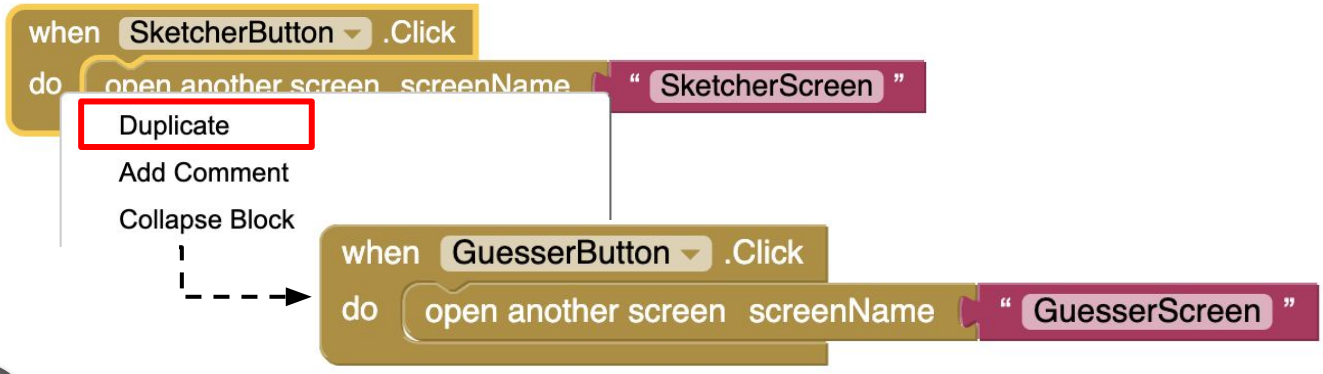
OPEN AND CLOSE GUESSERSCREEN

Just like with **SketcherScreen**, you need to add the code that allows the user to get to the **GuesserScreen** by pressing the **GuesserButton** on **Screen1**, and to return to **Screen1** from **GuesserScreen** by pressing the **BackButton**.

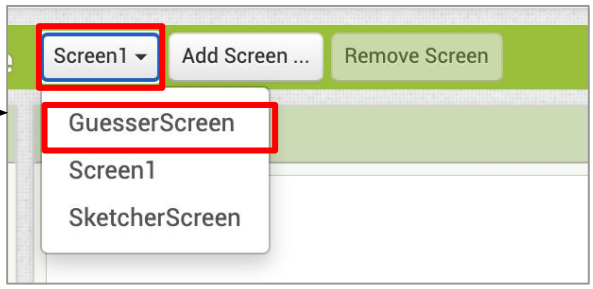
5 Return to Screen1 and open the Blocks Editor.



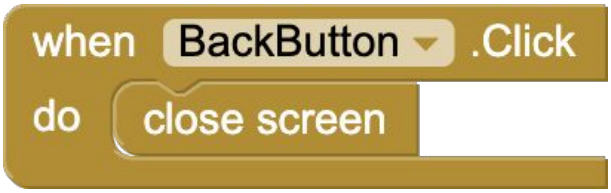
6 Right-click and duplicate the **SketcherButton.Click** block. Change **SketcherButton** to **GuesserButton**, and change the text block to "**GuesserScreen**".



7 Switch to **GuesserScreen**.



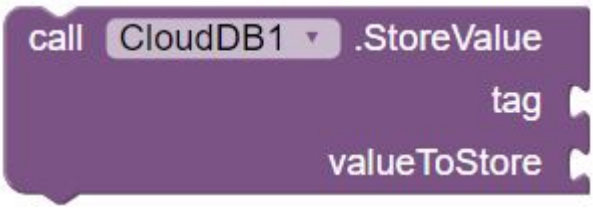
8 Code **BackButton.Click** to close the screen.

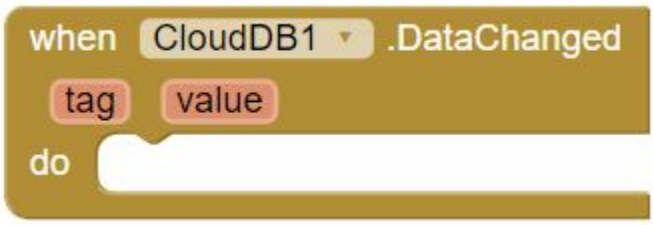


USING CLOUDDB TO DRAW ACROSS DEVICES

In this lesson, you will use the **CloudDB** component that you used in the Two-button Game.

Using the **CloudDB.StoreValue** and **CloudDB.DataChanged** blocks, two devices communicate with each other through CloudDB, based on the tag.

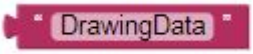




A **tag** is a name you give to data. It works like a variable. Each **tag** has a **value**, just like variables have values.

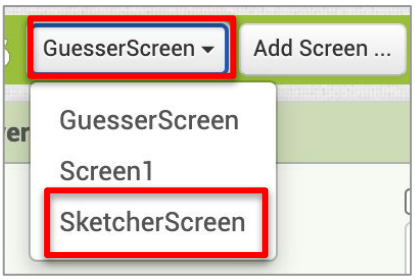
You store a value by its **tag**, and you can see what has changed for each tag.

The name of the tag to store what is being drawn is **DrawingData**.

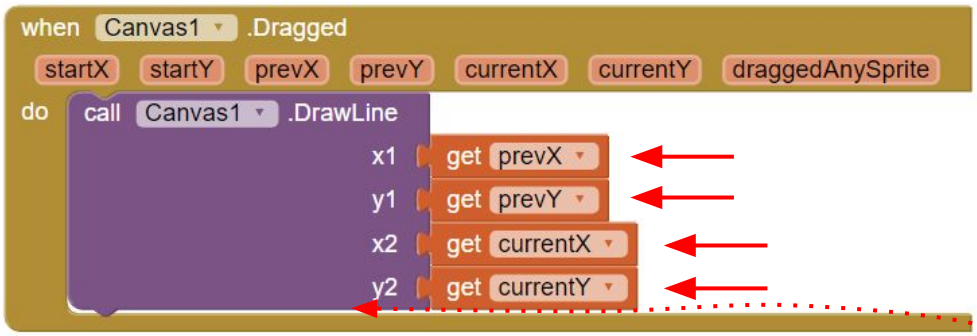
Tags	Meaning	Sketcher (You)	Guesser (Partner)
	The start point and end point for drawing	Store the coordinates of drawing	Get the coordinates of drawing

USING CLOUDDB TO DRAW ACROSS DEVICES

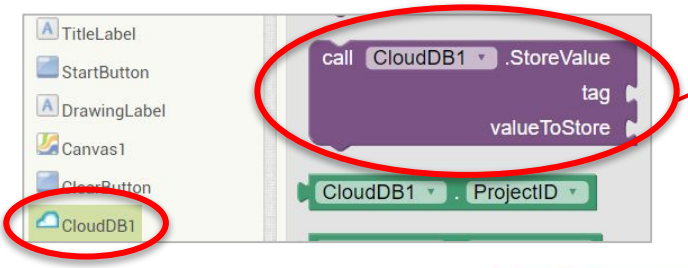
9 Switch to **SketcherScreen**, and open the Blocks Editor. The Sketcher needs to store the drawing information to CloudDB as they draw.

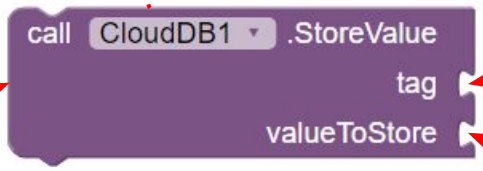


There are four coordinates that are used to draw when you drag on the screen. You will package those 4 coordinates in a list and store them in a single tag in CloudDB rather than using 4 separate tags.

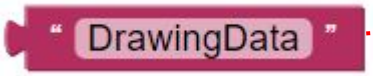


10 Pull out **CloudDB1.StoreValue** from the CloudDB1 drawer.

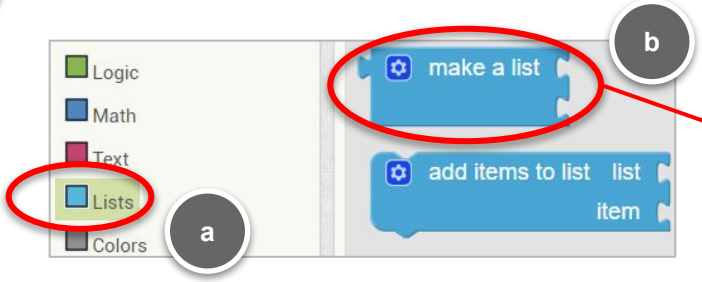


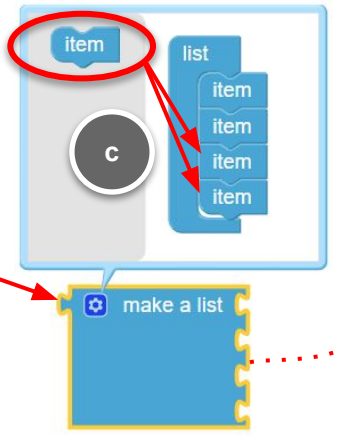


11 The tag will be “DrawingData”.



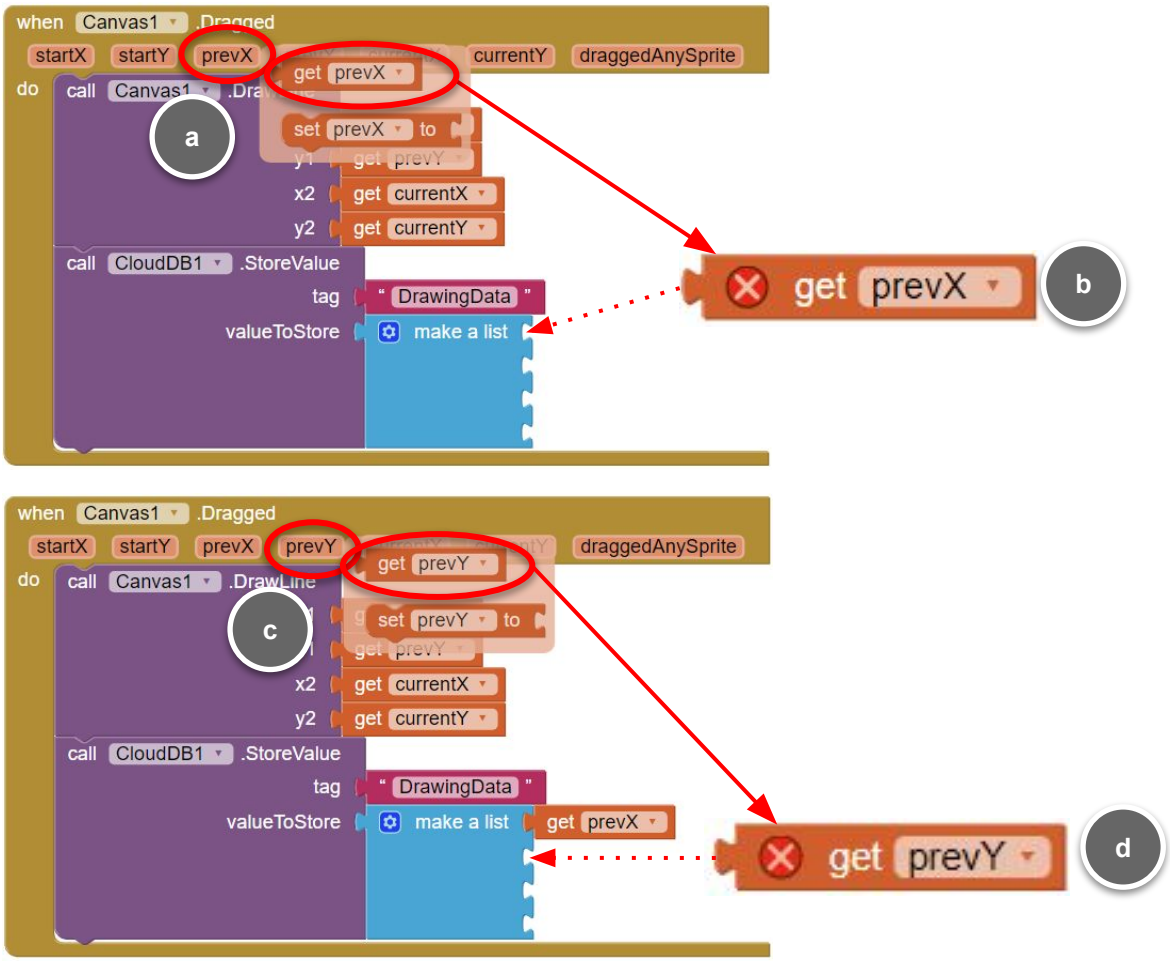
12 Drag out a **make a list** block for the **valueToStore.**



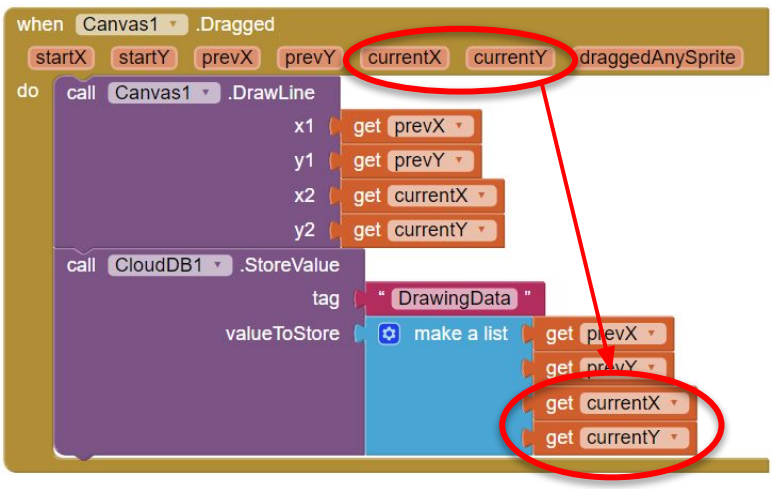


STORE THE COORDINATES IN CLOUDDB

13 Mouse over the **prevX** and **prevY** blocks from **Canvas1.Dragged** and snap those blocks to the **make a list** block.



14 Do the same for the remaining blocks: **currentX** and **currentY**.

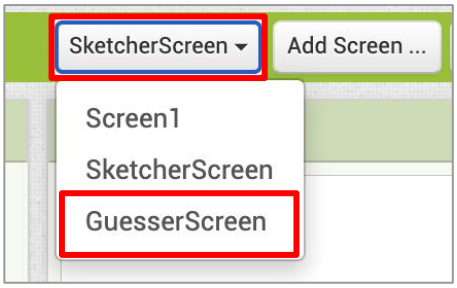


GET THE DRAWING COORDINATES FROM CLOUDDB FOR THE GUESSER

You just coded blocks to store drawing coordinates on the CloudDB server using the **CloudDB.StoreValue** block.

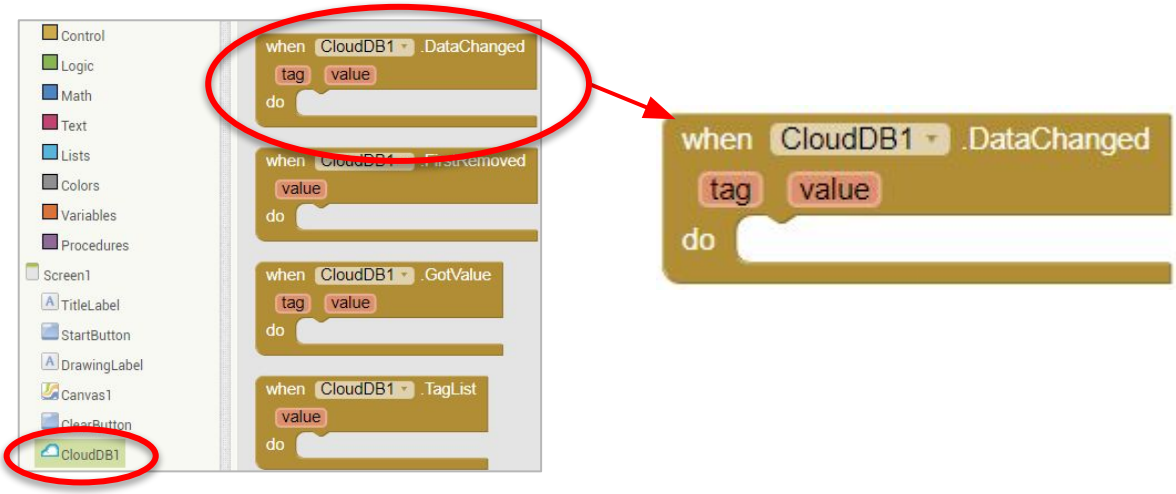
The Guesser now needs to see what the Sketcher drew on their device.

15 Switch to the **GuesserScreen**, and make sure you are in the **Blocks Editor**.



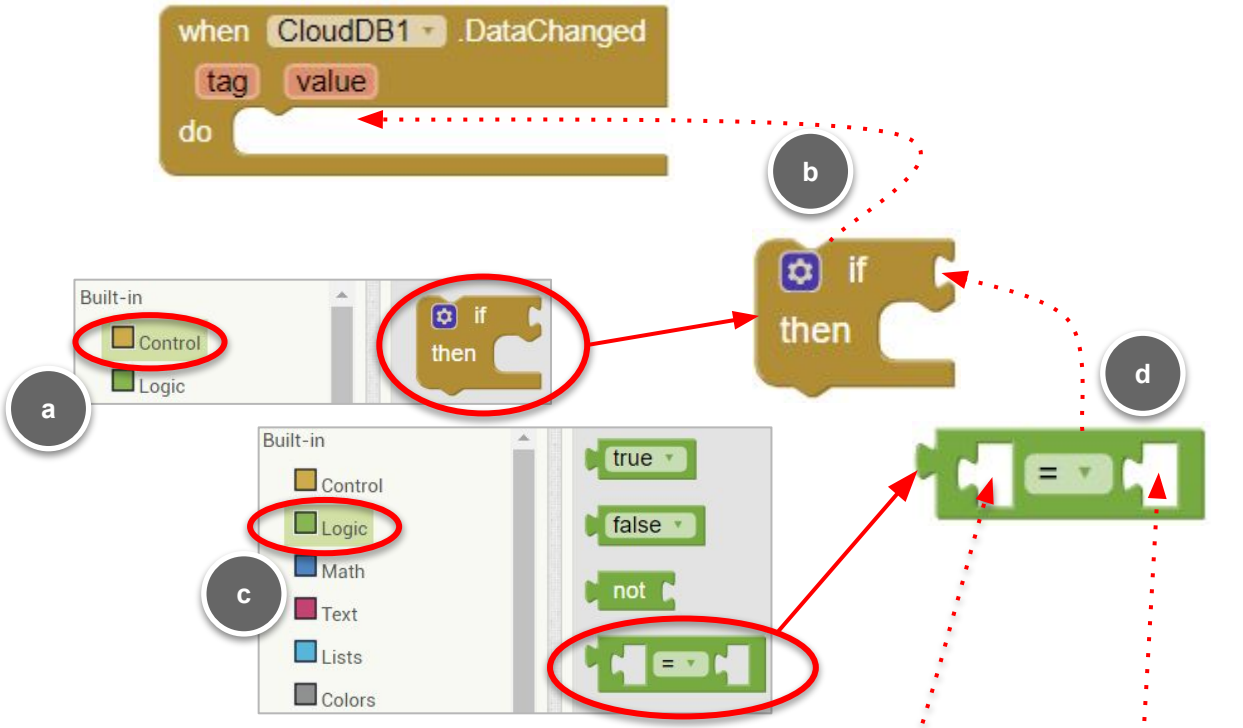
When the Sketcher draws, the drawing data is stored in CloudDB. CloudDB then needs to update all the other users with that drawing data, so the drawing can appear on their devices. This can be done with the **DataChanged** event.

16 Pull out the **CloudDB1.DataChanged** block from the **CloudDB1** drawer.

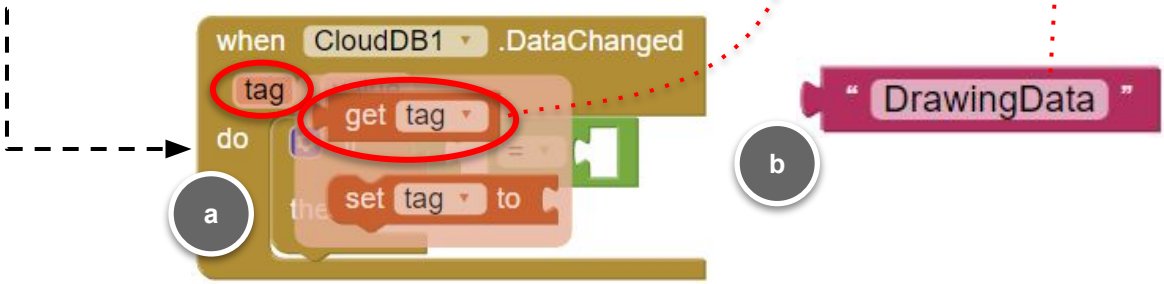


GET THE DRAWING COORDINATES FROM CLOUDDB

17 Many tags can be saved to CloudDB, so you need to check and make sure you've got the right tag, "DrawingData".



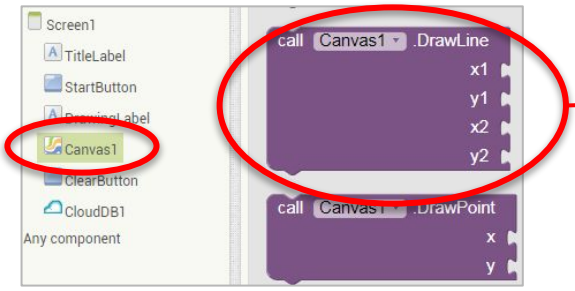
18 Test if the tag = "DrawingData".

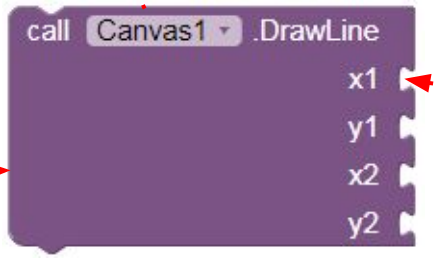


GET THE DRAWING COORDINATES FROM CLOUDDB

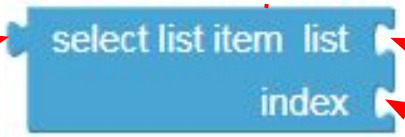
Remember that the drawing coordinates were stored as a list.

19 You need to select each item in the list to get the four values to pass to **Canvas1.DrawLine.**

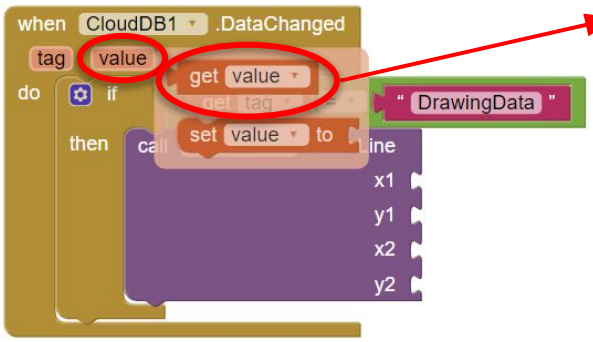




20 Drag out a **select list item** block and snap to **x1.**



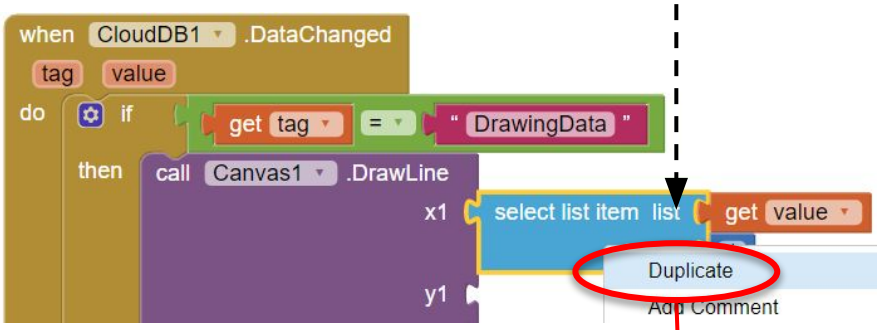
21 Mouse over value and snap the **get value** block to the **list** in the **select list item** block.



22 And drag a Math **0** block, and change to **1** for the **index.**

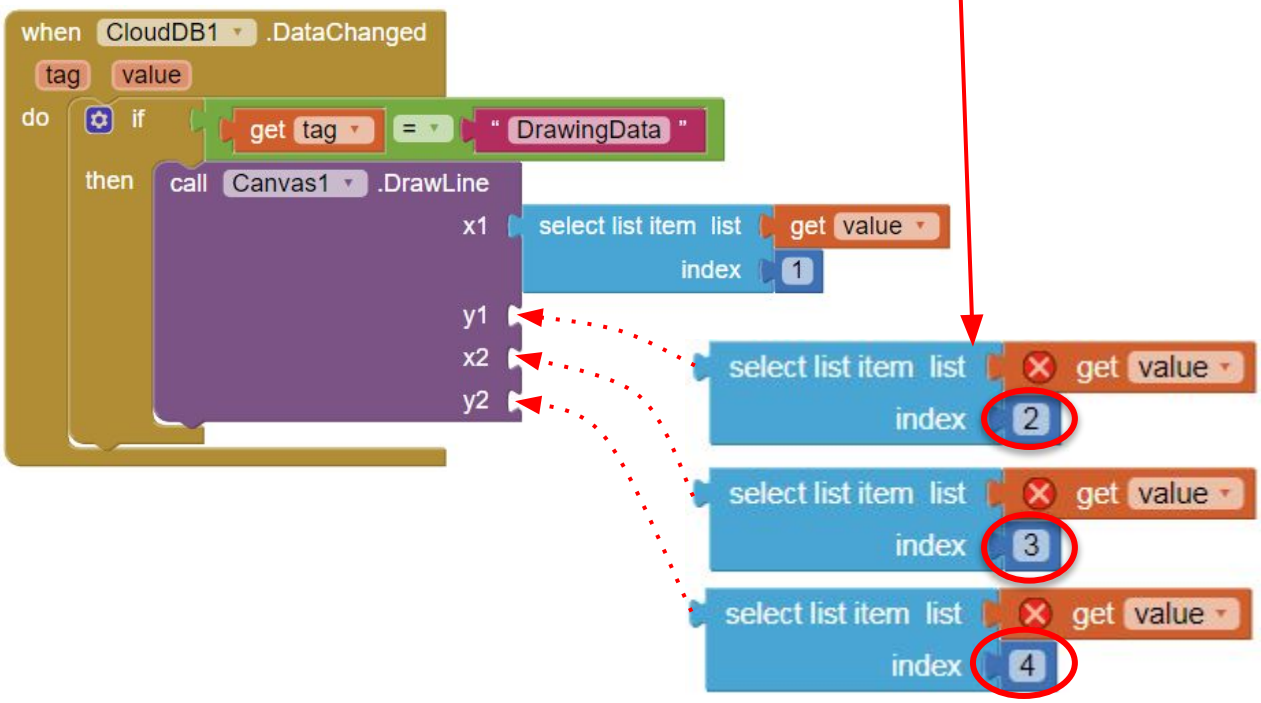
GET THE DRAWING COORDINATES FROM CLOUDDB

23 Duplicate the **select list item** block three times.



x3

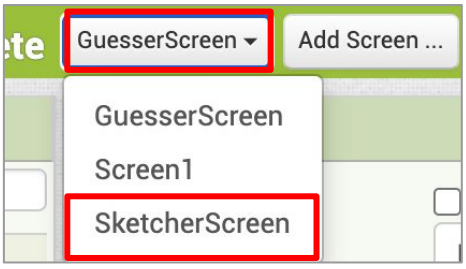
24 Change the index appropriately for each, then snap into the **y1**, **x2**, and **y2** slots in **Canvas1.DrawLine**.



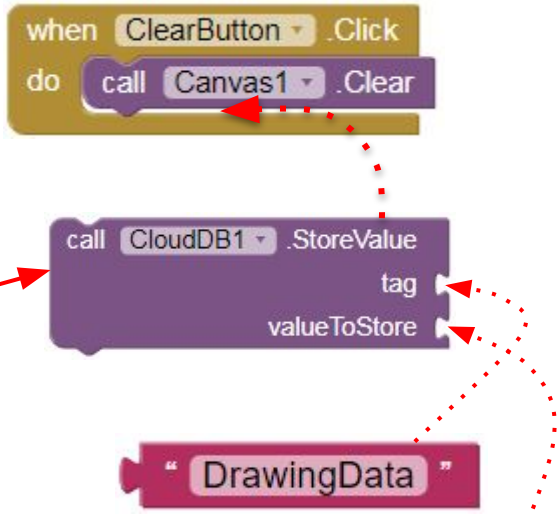
CLEAR THE CANVAS

When the Sketcher clears the screen, they need to send a message to the other devices to clear their screens too.

25 Switch back to the **SketcherScreen**.

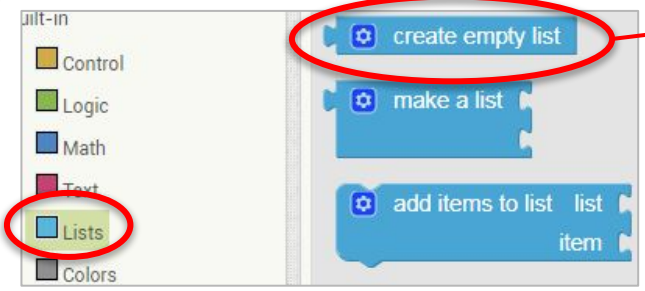


26 To signal the screen should be cleared, store an empty list using the "DrawingData" tag in CloudDB in the **ClearButton.Click** event.

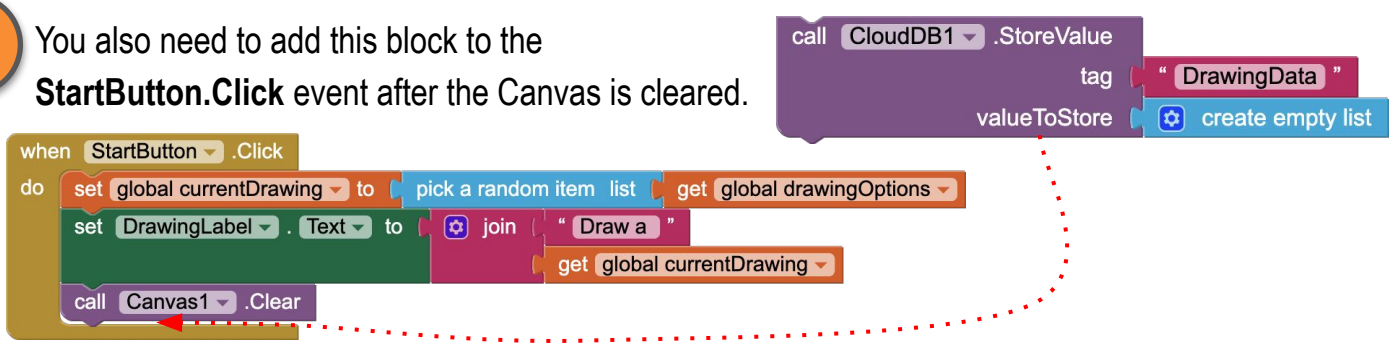


27 Drag a text block with the text "DrawingData" and snap in as the **tag**.

28 And snap in **create empty list** as the **valueToStore.**



29 You also need to add this block to the **StartButton.Click** event after the Canvas is cleared.

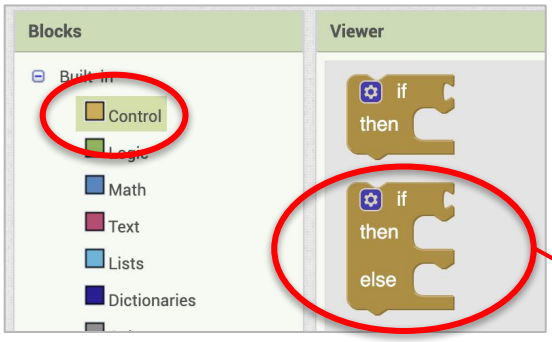


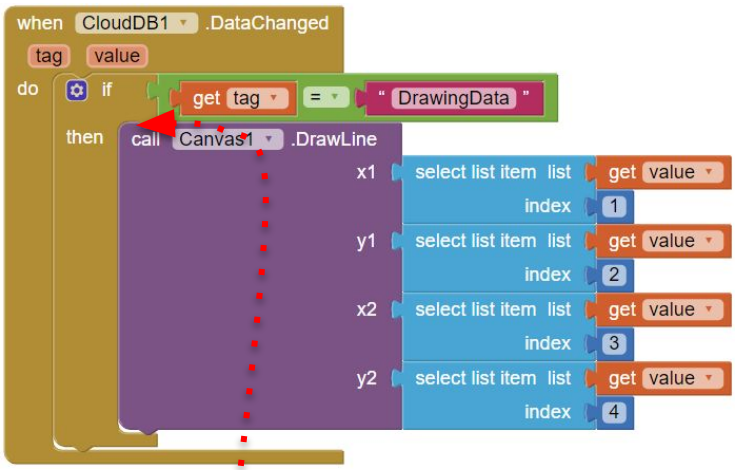
CLEAR THE CANVAS

Just as with drawing lines, you need to add code to the **CloudDB1.DataChanged** event to check if you need to clear the Canvas on the Guesser's device.

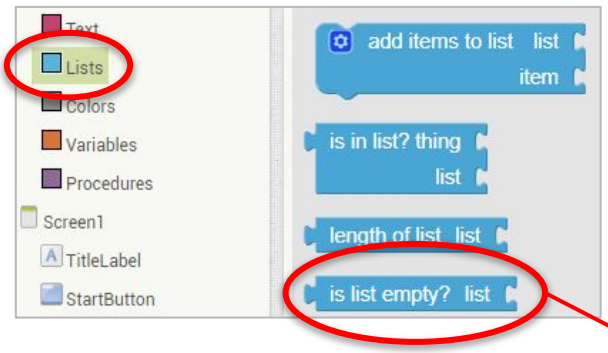
30 Switch back the **GuesserScreen** and find the **CloudDB1.DataChanged** event.

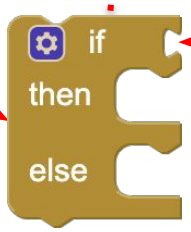
31 Drag out an **if-then-else** block.
Snap it inside the **if tag = "DrawingData"** block.





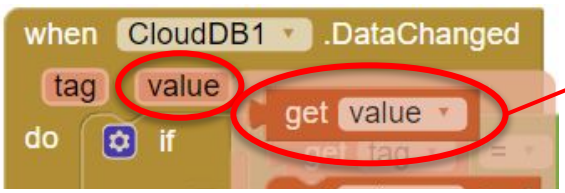
32 Snap in a **is list empty?** block from the Lists drawer.





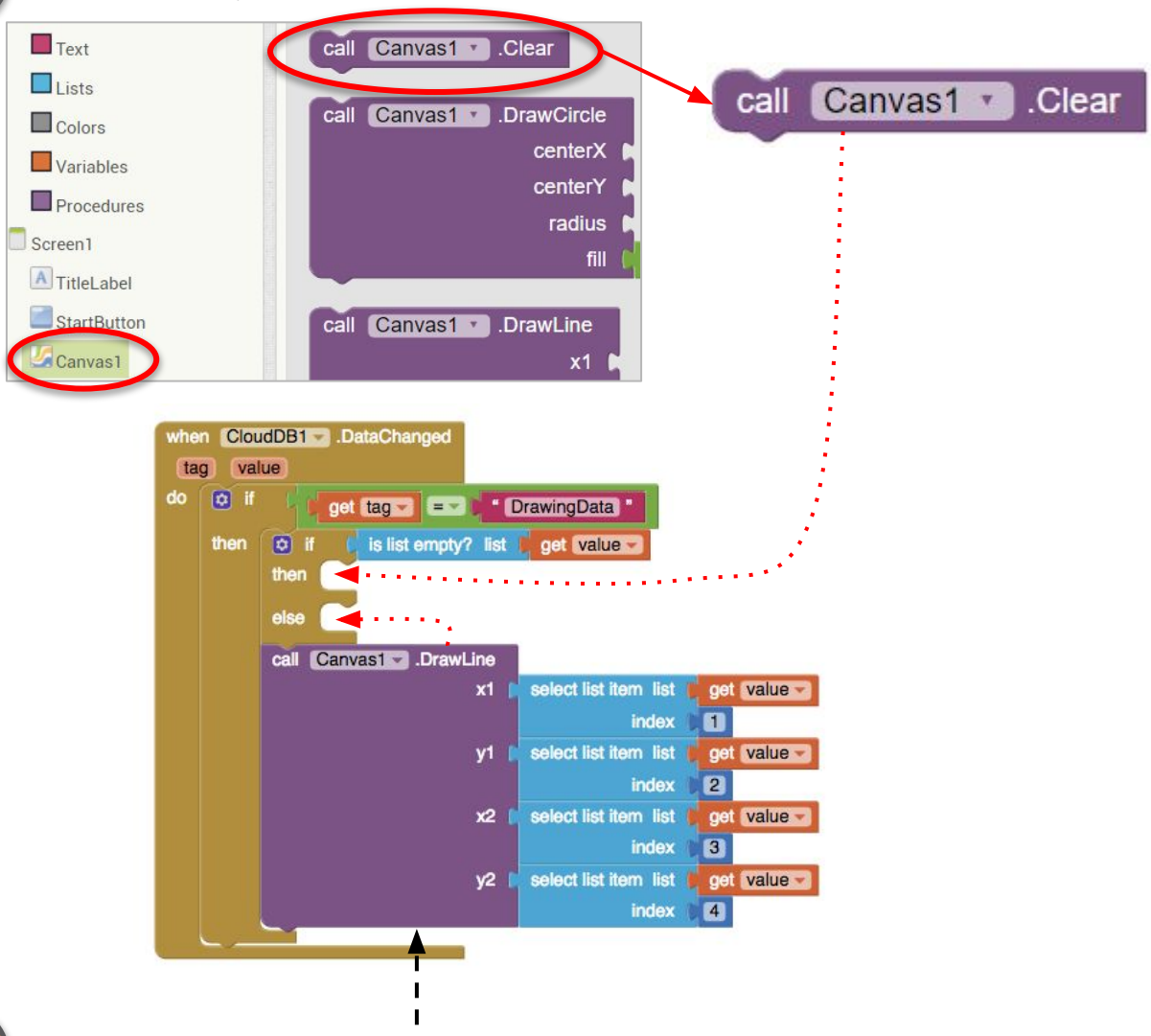
An **if** block inside another **if** block is called a **nested if.**

33 The **value** from CloudDB will be the list to check.



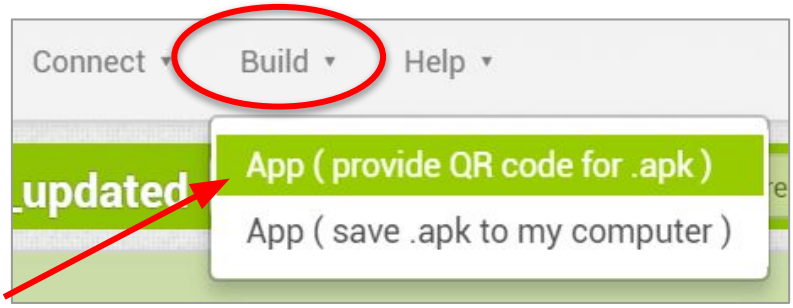
CLEAR THE CANVAS

34 If the list is empty, clear the Canvas.



35 Lastly, move the **Canvas1.DrawLine** block into the **else** part of the **if-then-else** block.

36 Test with your partner. Build the apk using the QR code option, scan the QR code and download and install the apk on your individual devices. One person click the "I want to draw" button and the other click the "I want to Guess" button. Can one person draw and the other see it being drawn on their device?

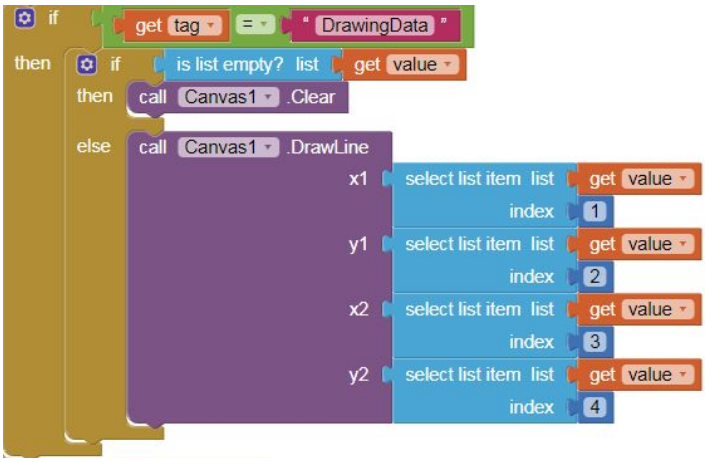


COMPUTATIONAL THINKING CONCEPTS

Sketch And Guess Part 3

1.Events

2. Conditionals



3. Naming / variables

4. Manipulation of data and elementary data structures