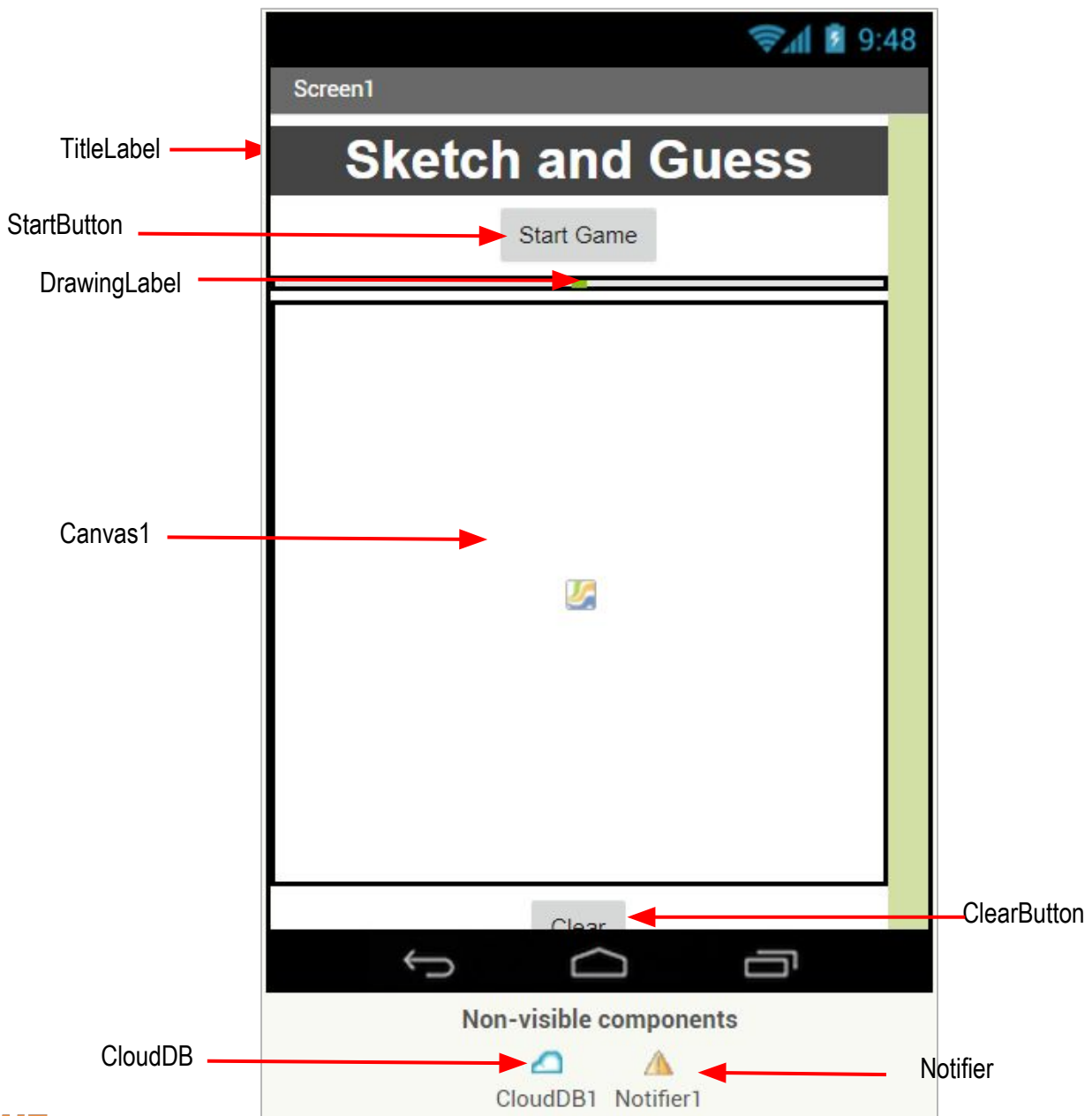# SKETCH & GUESS: PART 1

START HERE

In this lesson, you will make a simple drawing app.

1 Open the SketchAndGuess_template project.

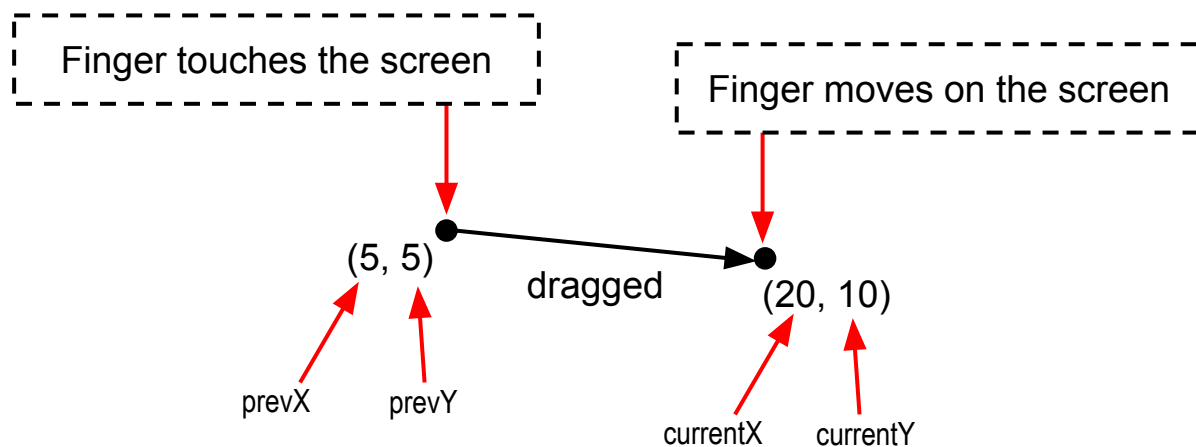**2** With your partner, look at the user interface components that are included in the template. Try to figure out what all the components do.

TitleLabel

StartButton

DrawingLabel

Canvas1
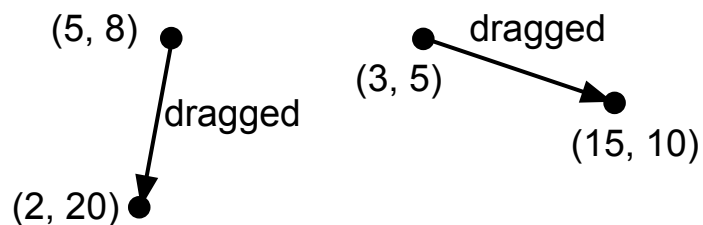
ClearButton

CloudDB

Notifier

HOW TO DRAW A LINE IN THE APP

To draw something on the Canvas, you need to use the **when Canvas.Dragged** block and the **call Canvas1.DrawLine** blocks.

The diagram below shows how to use coordinates to draw a line in the app. A line is drawn by joining two points. Using **Canvas1.DrawLine**, you need to specify the position of the start point (x1, y1) and the position of the end point (x2, y2).

```
┌─────────────────────────────┐
│  Finger touches the screen  │
└─────────────────────────────┘
                                    ┌─────────────────────────────┐
                                    │  Finger moves on the screen │
                                    └─────────────────────────────┘

          (5, 5)  ●━━━━━━━━━━━━━━━━●
                      dragged        (20, 10)

      prevX    prevY          currentX   currentY
```
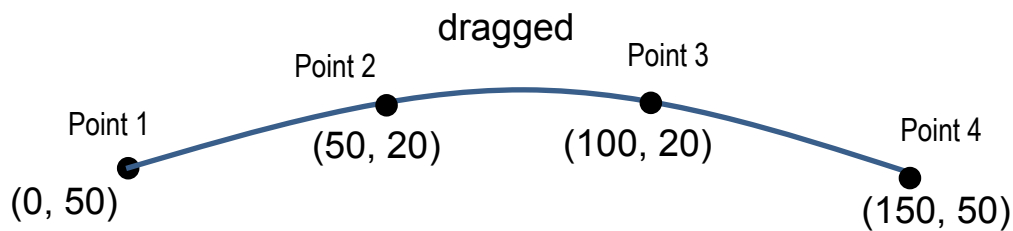
If you want to draw a line in the app, you need to use **prevX**, **prevY** and **currentX**, **currentY**. Below please work with your partner to fill in the blanks.

```
  (5, 8) ●                    ● dragged
          \                  (3, 5) \
           \ dragged                 \
            \                         ● (15, 10)
  (2, 20) ●
```

prevX		
prevY		
currentX		
currentY		

HOW TO DRAW A CURVE IN THE APP

A line is formed by connecting many dots. Each of the dots is represented by its coordinate (x,y).

dragged

Point 2

Point 3

Point 1

(50, 20)   (100, 20)

Point 4

(0, 50)

(150, 50)

If you want to draw a curve in the app, you need to draw many lines, each with its own **prevX**, **prevY** and **currentX**, **currentY**. Below please work with your partner to fill in the blanks.

	segment 1 (Point 1-Point 2)	segment 2 (Point 2-Point 3)	segment 3 (Point 3-Point 4)
prevX			
prevY			
currentX			
currentY			

MIT
APP INVENTOR

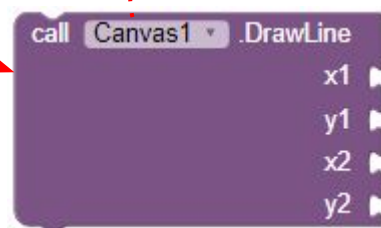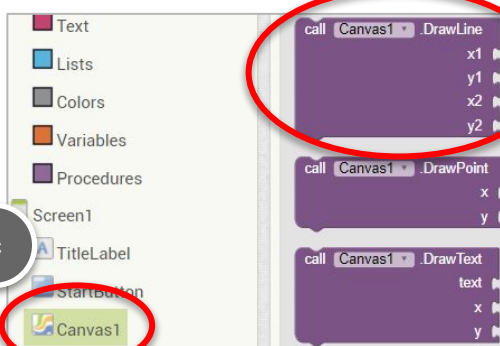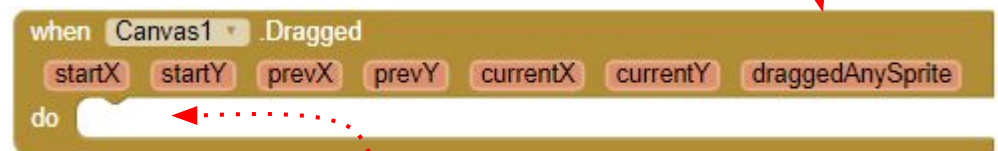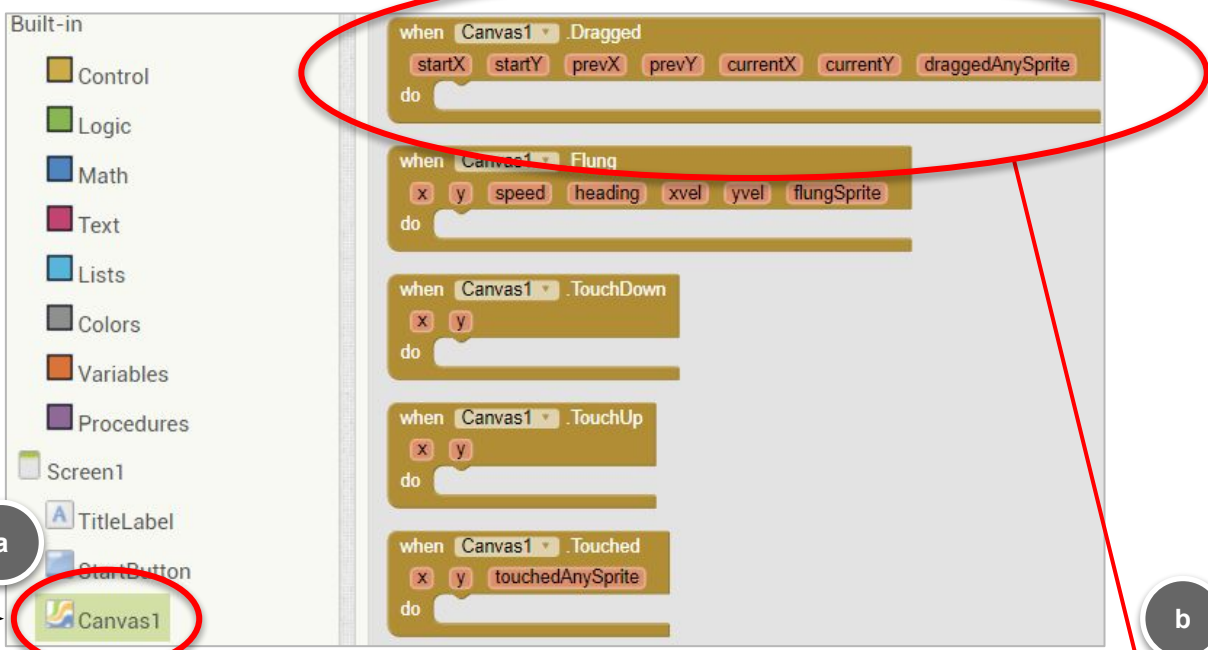## DRAW WHEN USER DRAGS ON THE CANVAS

3 Switch to the Blocks editor from the Designer. ─ ─ ─ ─ ─ ─ ➤
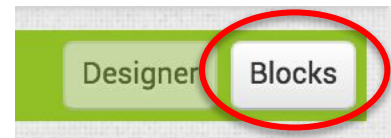
Notice that there are some code blocks already provided for you. Some blocks won't be used until Lessons 2 and 3. Do not worry! The student guides will let you know when and how to use them!
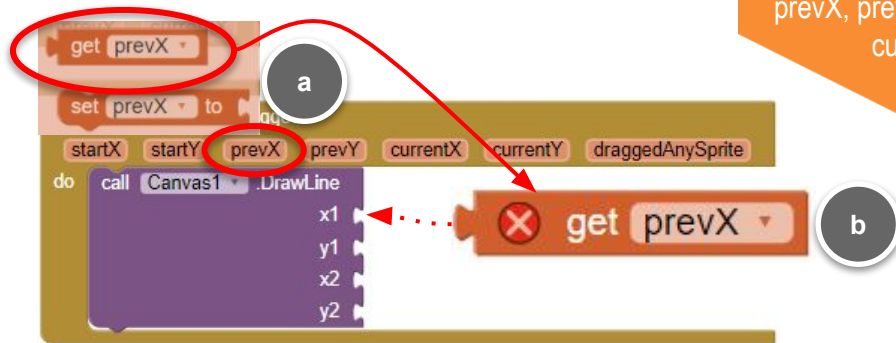
4 From the Blocks menu, pull out **Canvas1.Dragged** and **Canvas1.DrawLine** blocks.
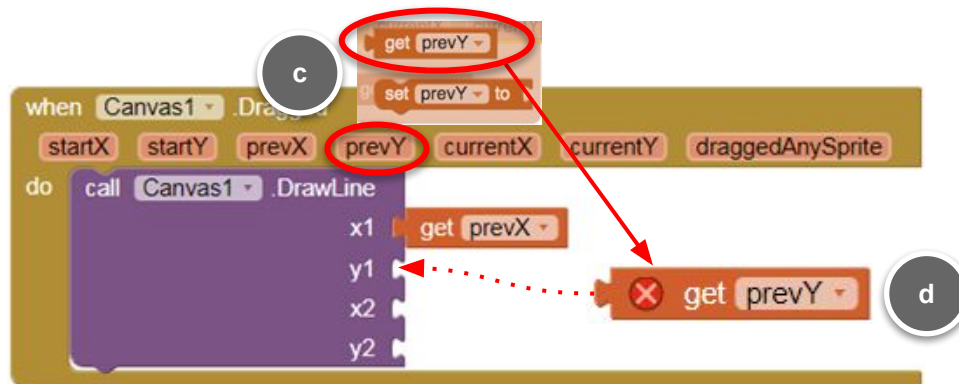
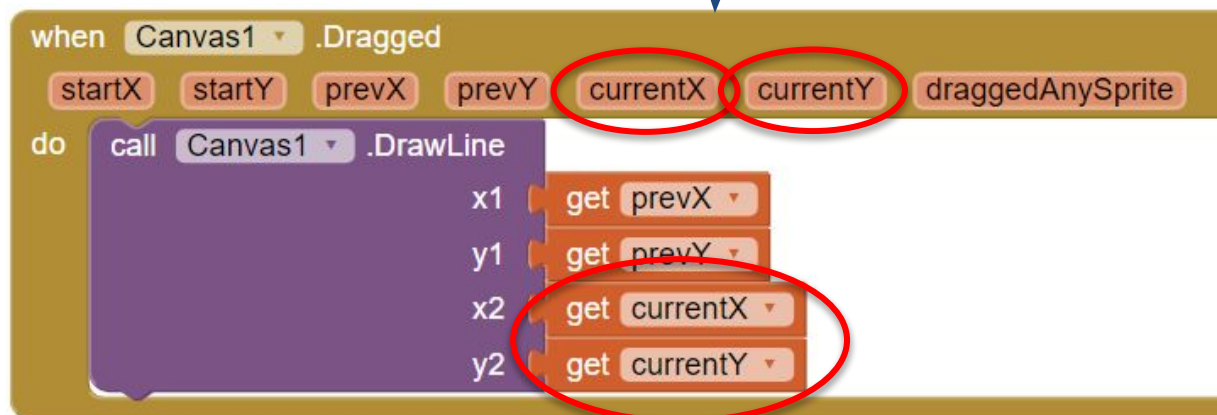## SPECIFY THE START AND END POINTS

5 Set the start point.

startX and startY mark where the user first starts drawing. You will continue to keep track of X,Y as the user draws, so use prevX, prevY to currentX, currentY.
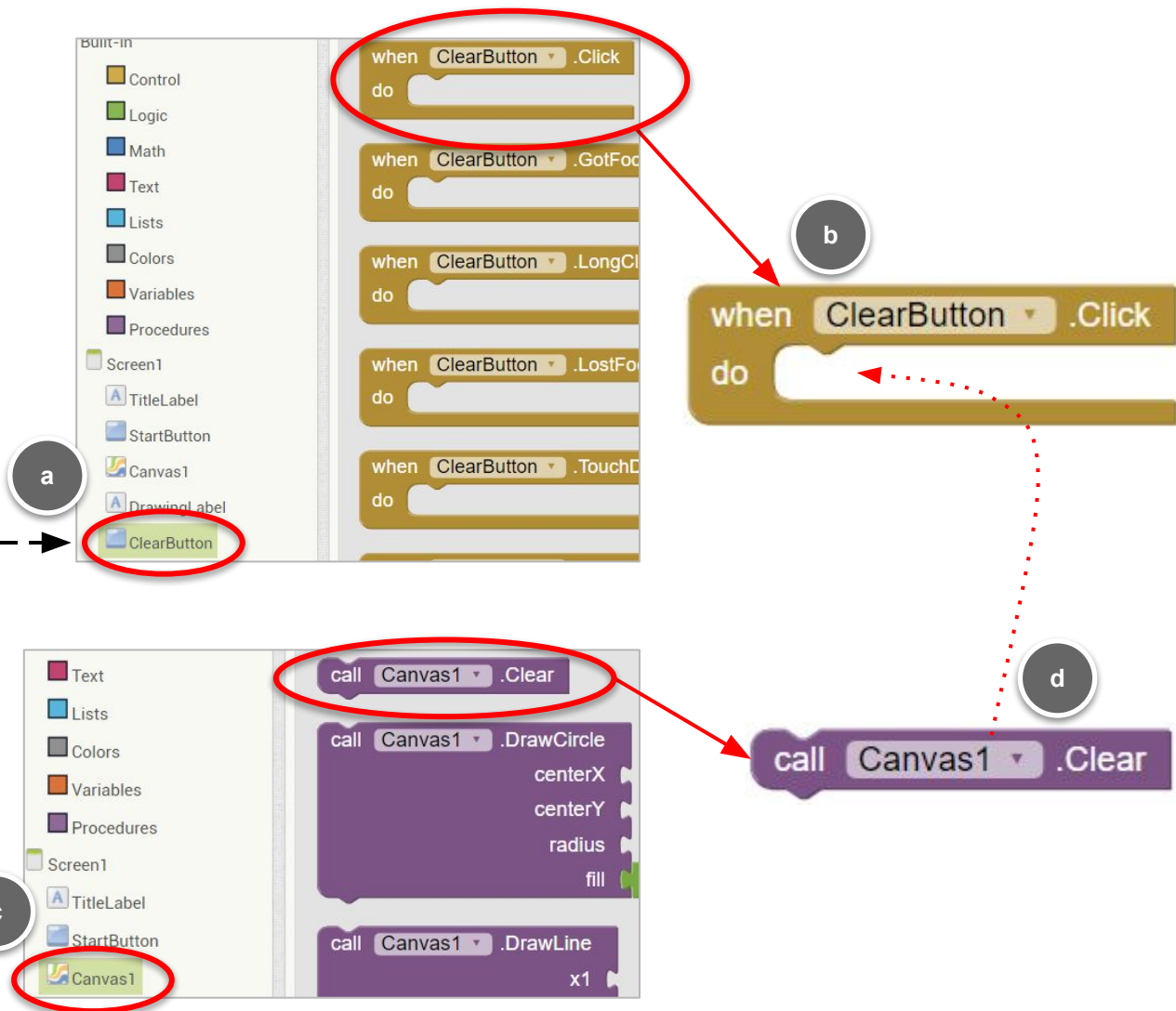
Hover your mouse over the prevX and pull out the **get prevX**

6 Continue to set the end point: **x2** and **y2** to **currentX** and **currentY** respectively.
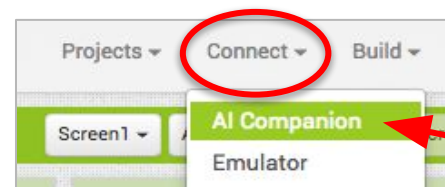
CLEAR THE CANVAS

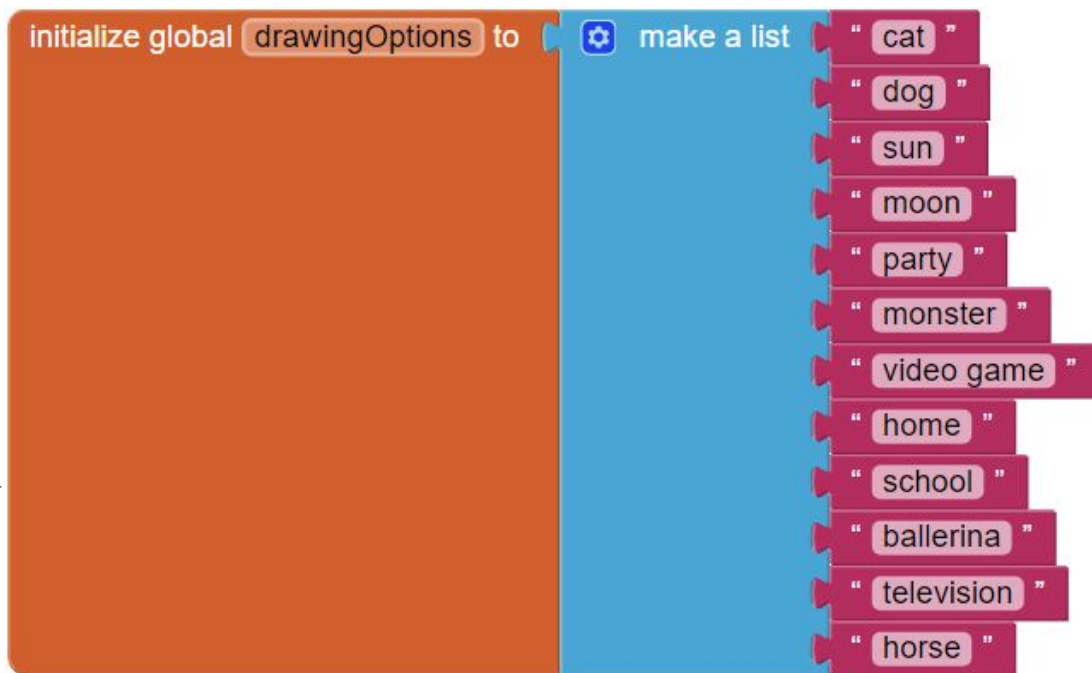**7**   Pull out **ClearButton.Click** block and add a **Canvas1.Clear** block to it.

8   Test your app using MIT AI2 Companion.

TELL THE SKETCHER WHAT TO DRAW

9 There is a list variable in the template, calling **drawingOptions**, that has a list of possible objects to draw. Modify the items as you wish.
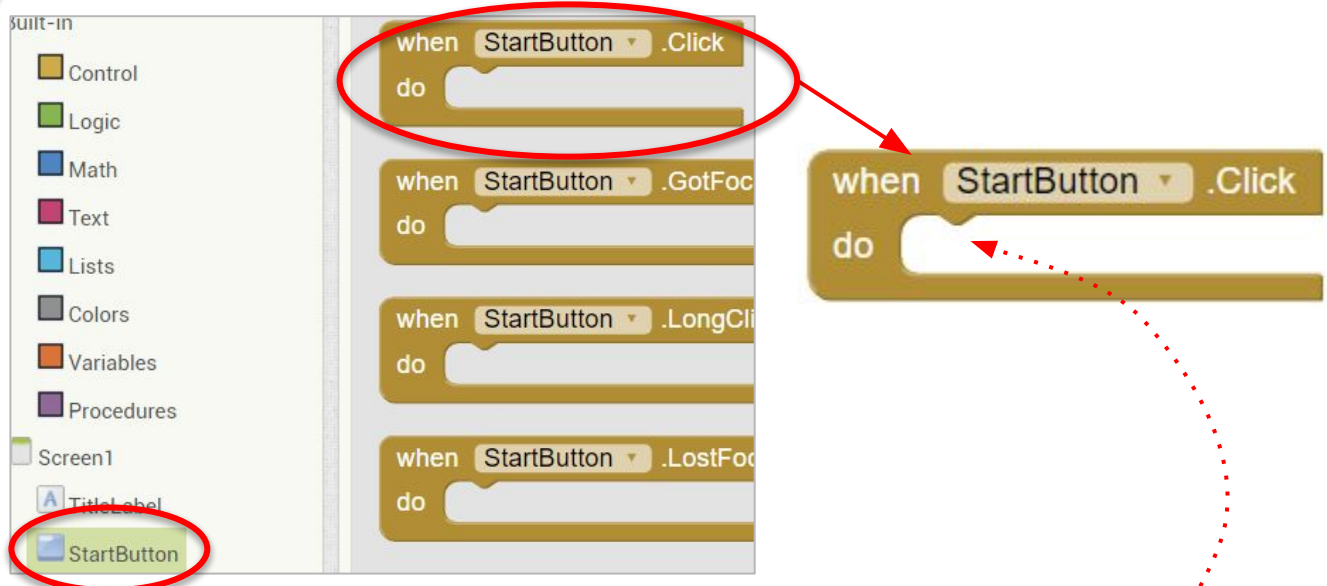

10 Also note there is a **currentDrawing** variable, which will be used to save a random item from the **drawingOptions** list.

PICK A RANDOM ITEM WHEN STARTBUTTON IS CLICKED

11   Pull out a **StartButton.Click** block.

12   Pick a random item from the drawingOptions list, then save it to the variable, currentDrawing.
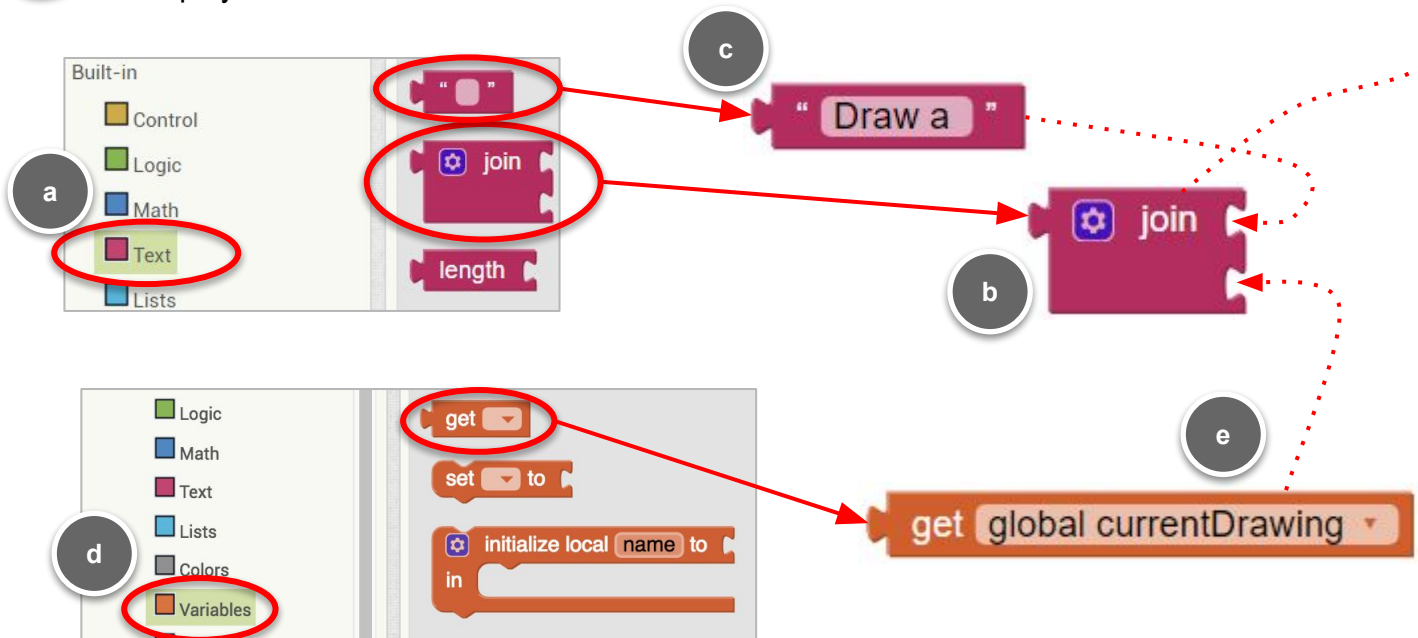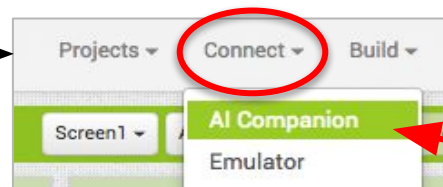
DISPLAY RANDOM ITEM TO DRAW

13 Pull out a **set DrawingLabel.Text** block.

```
when  StartButton .Click
do    set global currentDrawing to    pick a random item  list    get global drawingOptions
```

Screen1
- TitleLabel
- StartButton
- Canvas1
- DrawingLabel

```
DrawingLabel . Text
set DrawingLabel . Text to
DrawingLabel . TextColor
```

```
set DrawingLabel . Text to
```

14 Add a **join** block from the Text drawer to display what to draw in the Label.

Built-in
- Control
- Logic
- Math
- Text
- Lists

a

```
" ☐ "
join
length
```

c

```
" Draw a "
```

b

```
join
```

- Logic
- Math
- Text
- Lists
- Colors
- Variables

d

```
get ▾
set ▾ to
initialize local name to
in
```

e

```
get global currentDrawing
```

15 Test using the MIT AI2 Companion.

Projects ▾   Connect ▾   Build ▾

Screen1 ▾

AI Companion
Emulator

COMPUTATIONAL THINKING CONCEPTS
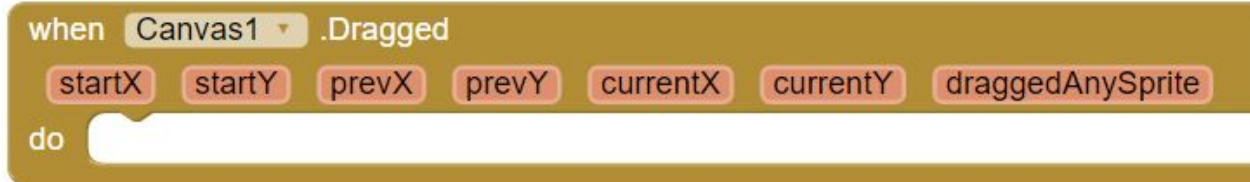
Sketch And Guess Part 1
1. Sequences
2. Events
3. Naming/Variables
4. Manipulation of data and elementary data structures