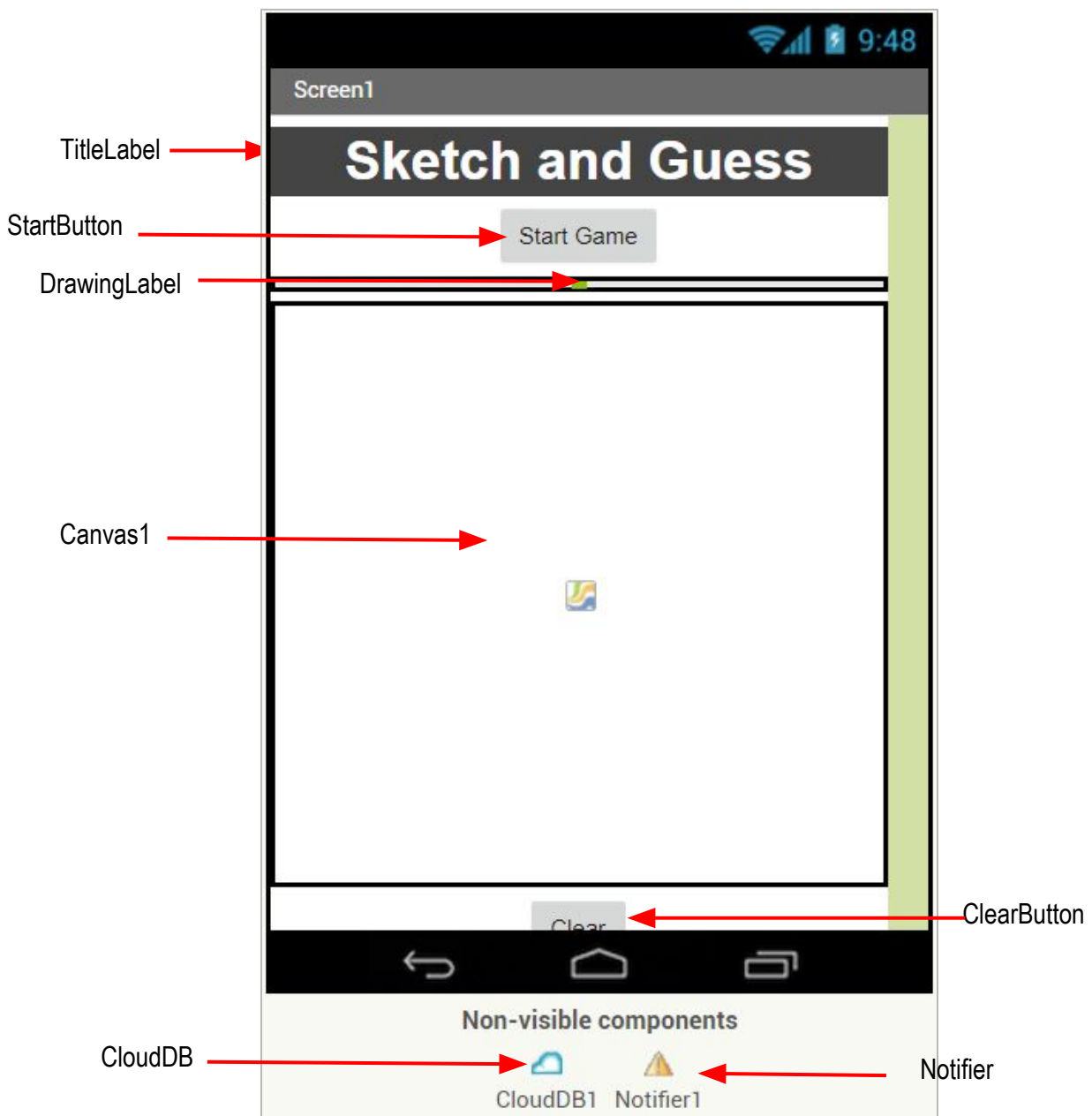# SKETCH & GUESS

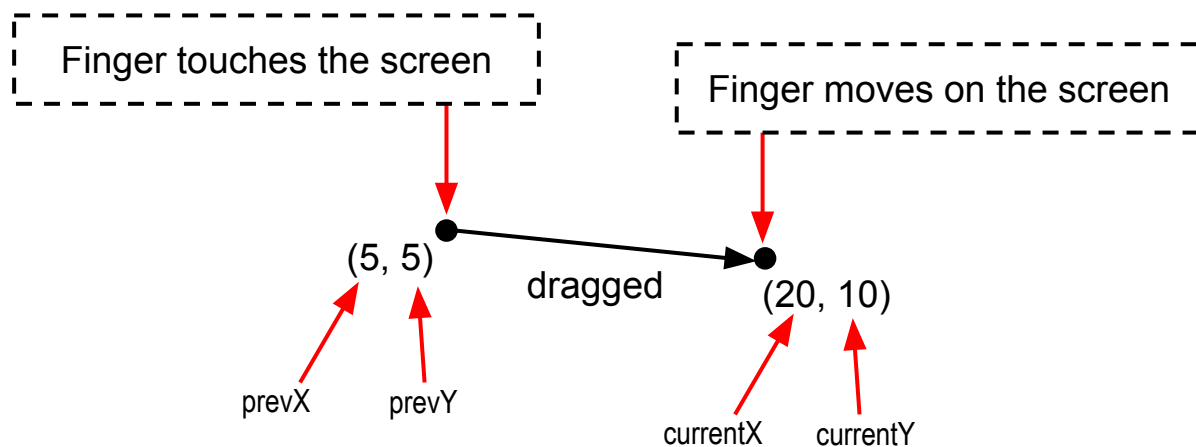In this lesson, you will make a simple drawing app.

START HERE

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.
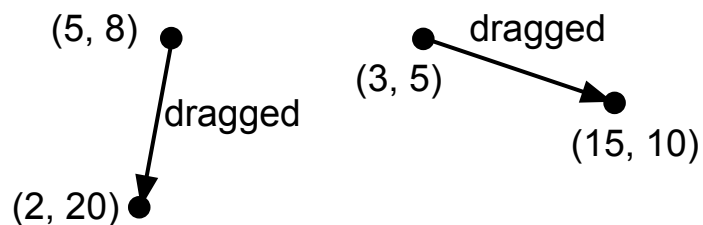
HOW TO DRAW A LINE IN THE APP

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)                           (20, 10)
                         dragged

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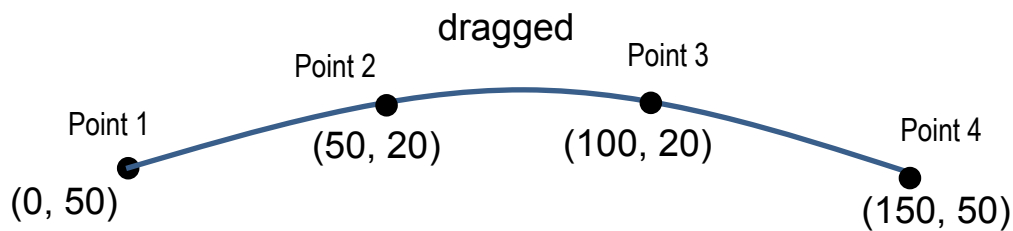
If you want to draw a line in the app, you need to use **prevX**, **prevY** and **currentX**, **currentY**. Below please work with your partner to fill in the blanks.

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

prevX		
prevY		
currentX		
currentY		

MIT
APP INVENTOR

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			

TEMPLATE CODE

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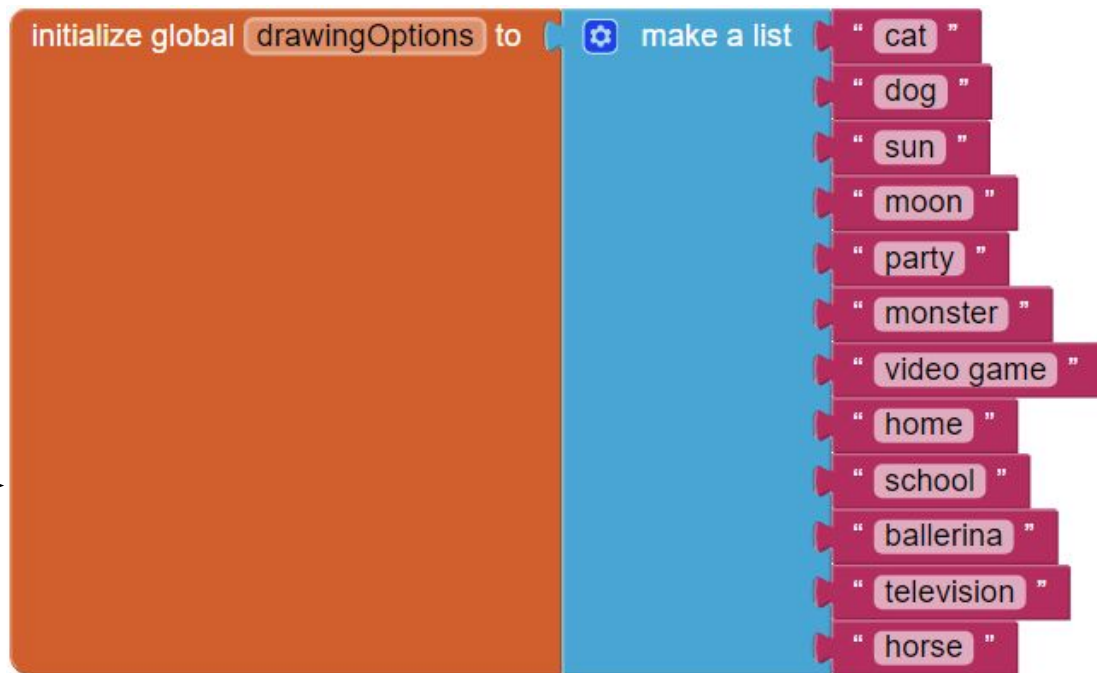
❏ Also note there is a **currentDrawing** variable, which will be used to save a random item from the **drawingOptions** list.

When adding many new features to a game, it is always a good idea to break it down into parts. Get one part working before moving on.

THE APP CHALLENGE

Part 1: Make a simple drawing app

- ❏ When the **StartButton** is clicked, a random item is chosen from the **drawingOptions** list, and displayed in **DrawingLabel**, to tell the user what to draw.

- ❏ When the user drags their finger on the **Canvas**, a black line is drawn where their finger drags.

- ❏ When the user clicks the **ClearButton**, the **Canvas** clears.

Part 2: Send drawing data to CloudDB so you can see drawings across devices

- ❏ When the user drags their finger across the screen, in addition to drawing, the drawing information (**prevX, prevY, currentX, currentY**) is stored in **CloudDB**. HINT: Store in a list, and it can be stored using a single tag.

- ❏ When new drawing data is received from **CloudDB**, take the drawing data and use that to draw on the **Canvas**.

- ❏ When a user clicks the **ClearButton**, communicate that action to **CloudDB** as well.

- ❏ When information about the clearing of the Canvas is received from **CloudDB**, clear the **Canvas**.

THE APP CHALLENGE (continued)

Part 3: Set one user as the Sketcher, and others users as Guessers for the game

❏ Whoever presses the **StartButton** first will be the Sketcher.
 ❏ Set variable **isSketcher** to true to keep track that the current user is the Sketcher.
 ❏ Store the **userID** of this player using the tag "CurrentSketcher" in **CloudDB**.
 ○ **HINT:** When the app starts, generate a userID for the player using **random number from 1 to 99999** (this ensures each player has a unique ID.)
 ❏ Make the **AnswerSpinner** invisible, since the Sketcher does not need to guess what is being drawn.

❏ Retrieve updated information from **CloudDB** to set up a Guesser:
 ❏ Check for "CurrentSketcher" tag and that the value is not this user's **userID**.
 ○ If that is the case, this user is a Guesser, so
 ❏ set variable **isSketcher** to false.
 ❏ Make the **AnswerSpinner** visible.
 ❏ Display the message "Guess the drawing in **DrawingLabel**.

❏ When drawing on and clearing the **Canvas**, only allow the Sketcher to perform those actions.

❏ When receiving updated drawing and clearing information from **CloudDB**, only perform the drawing and clearing actions if the user is a Guesser.

**Part 4: Check for correct guesses for the Guesser**

❏ When the Sketcher starts drawing, store the **currentDrawing** information in **CloudDB**.

❏ When a Guesser receives **currentDrawing** information from **CloudDB**, store that in a variable for later answer checking.

❏ When a Guesser makes a guess by selecting from the **AnswerSpinner**, notify them whether they are correct or incorrect.

THE APP CHALLENGE (continued)

Final Challenge

❏ Add color buttons so the Sketcher can change the color of the pen.
NOTE: These are included in the template, in the **DrawingArrangement**.

❏ Use the Slider component to allow the Sketcher to change the size of the drawing pen.

❏ Store the color and pen line width in CloudDB so the drawing on the Guessers' devices appear in the same color and line width as the Sketcher's.

Below is the final layout of the app.
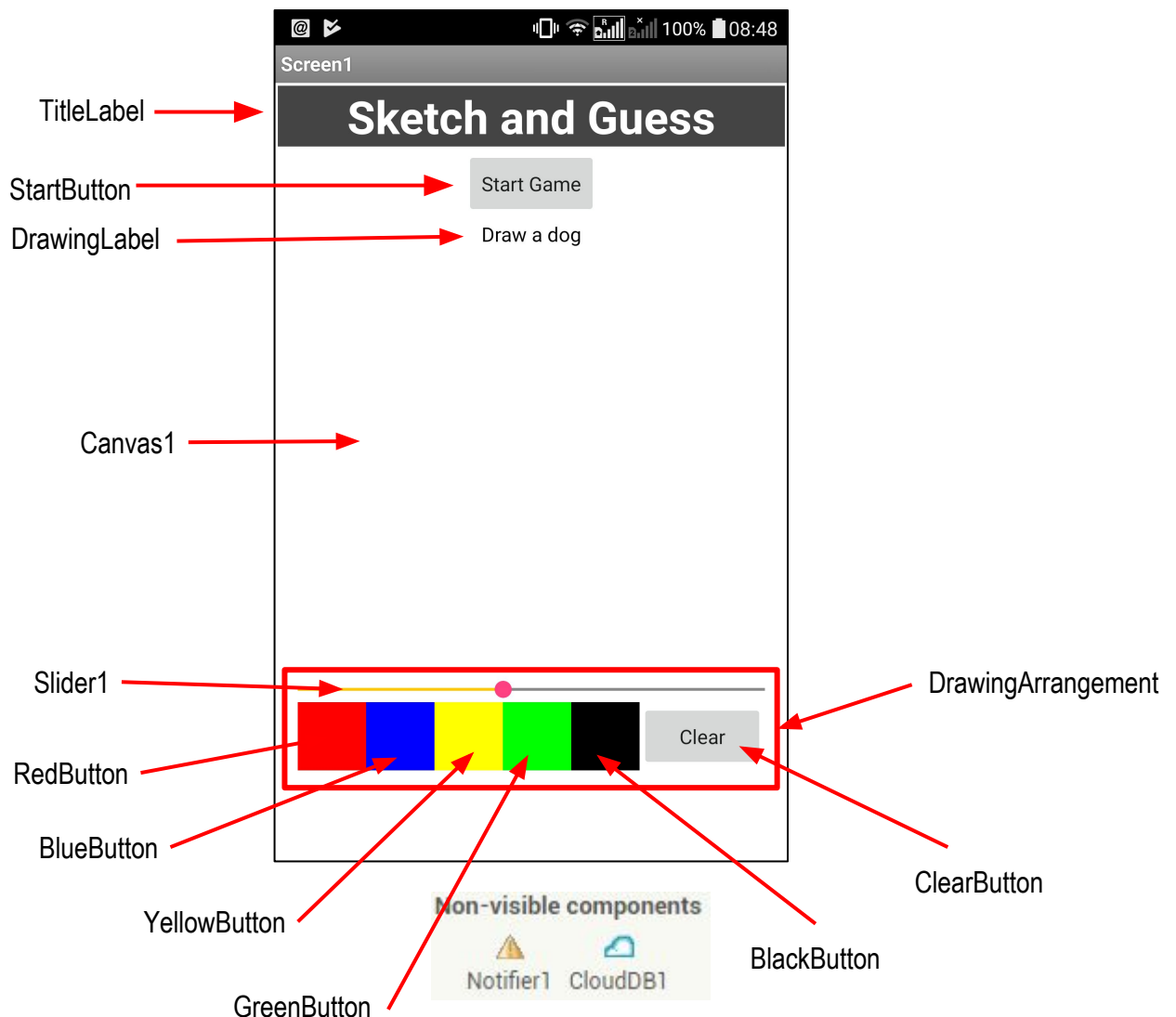
Choose Ways to Extend Your App

Here are a few features you could add if you want to expand your app

Add TextToSpeech to speak what is to be drawn

Add notification to all users if someone guesses correctly

Keep score! Each player can keep track of their correct guesses!
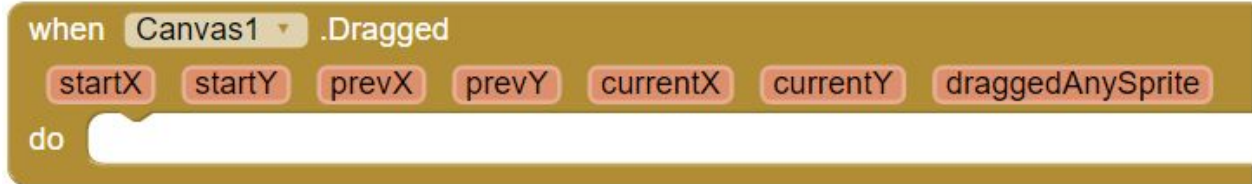
What other ideas do you have?

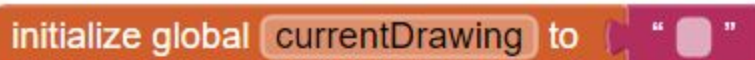# COMPUTATIONAL THINKING CONCEPTS

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