Sketch and Guess

1. Synopsis

In this unit, students will create a complex multiplayer drawing game, Sketch and Guess, in which a player can draw pictures and guess others' pictures, similar to the game of Pictionary. In the first lesson, students will develop a simple drawing game with the Canvas component. In the second lesson, students will add the ability to "send" drawings to other app users--so that they can watch the user draw in real time. In the third lesson, students will add the capability to guess the drawing. In the fourth lesson, students will add answer checking capability to the game. Some students may also add various features (such as multiple colors and changing line width) with the help of the "Student Guide: Challenge", making the app more user-friendly and fun. Students will apply the computational thinking practice of being incremental and iterative, since they will modify and add complexity to the app over four lessons. They will increase their knowledge of the use of conditionals, as they will employ nested if statements that use boolean operators. As always, testing and debugging will be necessary to make sure the apps work correctly.

2. Learning Objectives

After completing this unit, students will be able to:

1. Make a multiplayer drawing app that uses CloudDB.
2. Use CT concepts such as sequences, events, conditionals, parallelism, naming, operators, and data manipulation in creating an app.
3. Use the boolean "not" and "and" operators and nested if statements correctly.
4. Demonstrate understanding of how to use CloudDB to pass multiple pieces of information between devices;.
5. Work collaboratively to code and test a working multiplayer app.

3. Mapping with the CSTA Standards

This table show the alignment of this unit with the intended learning outcomes to the CSTA CS Standards. The entries in the tables indicate the expected relevance of the unit to each outcome:

2-NI-04	Model the role of protocols in transmitting data across networks and the Internet. [C] NI: Network Communication & Organization [P] Abstraction (4.4)	unplugged activity to model CloudDB
2-DA-07	Represent data using multiple encoding schemes. [C] DA: Storage [P] Abstraction (all)	Tag/value pairs in CloudDB
2-AP-10	Use flowcharts and/or pseudocode to address complex problems as algorithms. [C] AP: Algorithms [P] Abstraction (4.4, 4.1)	student guides provide simple flowcharts for students to complete
2-AP-11	Create clearly named variables that represent different data types and perform operations on their values. [C] AP: Variables [P] Creating (5.1, 5.2)	Boolean and number variables are used, as well as tag/value pairs in database.
2-AP-12	Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals. [C] AP: Control [P] Creating (5.1, 5.2)	nested if statements are used
2-AP-13	Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.	The unit is broken into 4 parts - students do not design themselves but

	[C] AP: Modularity [P] Computational Problems (3.2)	incrementally build.
2-AP-17	Incorporate existing code, media, and libraries into original programs, and give attribution. [C] AP: Program Development [P] Abstraction (4.2), Creating (5.2), Communicating (7.3)	Students use a template with some provided UI and code blocks.
2-AP-18	Systematically test and refine programs using a range of test cases. [C] AP: Program Development [P] Testing (6.1)	Testing happens within each lesson
2-IC-22	Collaborate with many contributors through strategies such as crowdsourcing or surveys when creating a computational artifact. [C] IC: Social Interactions [P] Collaborating (2.4), Creating (5.2)	Collaborative drawing

4. Learning Prerequisites

Students should have a command of the App Inventor development environment, and be familiar with CloudDB as a means of storing and sharing data in the cloud.

5. Lesson Plan

This unit consists for five 45 minutes lessons.

Lesson 1

Time	Activity
10 min	**Introduction to Drawing with Canvas Component** 1. Demonstrate a simple drawing app (SketchandGuess_checkpoint1.aia). 2. Introduce drawing on a Canvas and how lines are drawn based on user input.
30 min	**Coding of Simple Drawing App** 1. Ask students to pair up and work using the Pair Programming model to build a drawing app with the help of Student Guide: Lesson 1. 2. Test and debug the app.
5 min	**Wrap-up** 1. Review Canvas and drawing features learned in this lesson. 2. Ask students: a. "Have you played any similar games before?" (Pictionary) b. "How can you make this app work like Pictionary?"

Lesson 2

Time	Activity
5 min	**Review of Drawing App** 1. Review the drawing app. 2. Ask students how they can play this game with another person on a single device. 3. Ask students, "How can you play this game with partners who are using different devices?" 4. Teacher demonstrates the CloudDB version of the drawing app (SketchandGuess_checkpoint2.aia) to show Sketch and Guess between two devices.
10 min	**Demonstrating Drawing Using CloudDB** Run unplugged activity with students acting as CloudDB, and Sketcher and Guesser.
25 min	**Adding CloudDB Component to the App** 1. Explain that whenever a user draws something on their device, they will also save the x,y coordinates for the start and end of the drag event to CloudDB. Other users will take that information and use it to draw the same line on their device. 2. Ask students to work using the Pair Programming model, following Student Guide: Lesson 2, to add CloudDB to their apps. 3. Ask students to try the app with their partner. Each student will download the apk to their own tablet and can play the app together. One student draws and their partner can see the drawing happen on their tablet too.
5 min	**Wrap-up** Ask students how they might make this into a Sketch and Guess game, where one person draws and other players guess.

Lesson 3

Time	Activity
10 min	**Review and Introduction to Lesson** 1. Ask the whole class what else can be improved, and how they might make this into a Sketch and Guess game, where one

	person draws and other players guess. 2. The teacher demonstrates and explains the new function of the game: a "Sketcher" draws a randomly displayed word from a list of simple words included in the template. Other players may watch the drawing and guess what is being drawn. (SketchandGuess_checkpoint3.aia) 3. Show students how to add a Spinner component, and demonstrate how it works.
10 min	**New Conditional elements** 1. Explain to students they will need to do more complex conditionals in this app. 2. Review how nested if-blocks work. 3. Explain boolean operators (**not** and **and**). 4. Explain boolean variables.
20 min	**Coding the Sketch and Guess App** Following the "Student Guide: Lesson 3", students add code to their apps to make a partially working "Sketch and Guess" game.
5 min	**Wrap-up** 1. Ask students, "What is still missing from the app?" 2. Explain to students that they will implement guess checking in lesson 4.

Lesson 4

Time	Activity
5 min	**Introduction to Lesson** 1. Ask students what is missing from the existing Sketch and Guess app. 2. Teacher demonstrates the app with guess checking (SketchAndGuess_checkpoint4.aia) 3. Explain to students that they will implement guess checking in this lesson by adding code to provide feedback to users when they guess.
35 min	**Coding New Feature** 1. Following "Student Guide: Lesson 4", students add guess checking to their apps. 2. Student groups test the app for the new functionality. 3. Any student pairs who complete Lesson 4 coding may attempt the "Student Guide: Challenge" where they can add color

	buttons and a slider to change the line width in the app.
5 min	**Wrap-up** 1. Check in with students to see where they are with the app. 2. Explain that students have one more lesson to complete the app, try the Challenge, or add a different feature themselves.

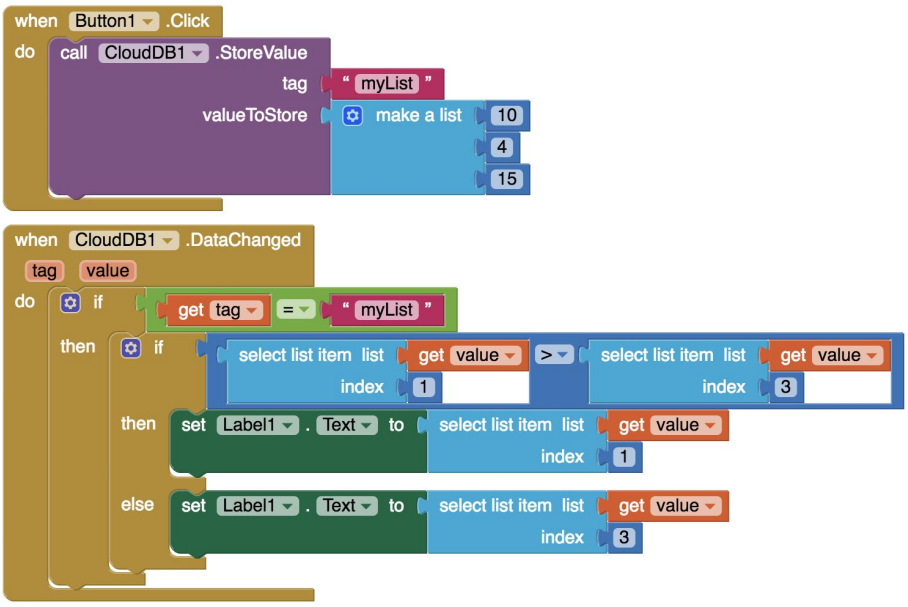
Lesson 5

Time	Activity
5 min	**Introduction to Lesson** 1. Ask students if there are other features they would like to add to the Sketch and Guess app. 2. Students can either finish the app if they have not completed it, or they can try the Challenge, where they can add color buttons and a slider to change the line width in the app.
25 min	**Coding** 1. Students who have not completed Parts 1-4 may work to complete the standard app. 2. Students may try the Challenge. 3. Students may consider adding their own new features, with teacher approval.
15 min	**Wrap-up** 1. Review the use of CloudDB in this unit. 2. Review nested if statements , boolean variables and operators. 3. Ask students to reflect on the game, and ask for volunteers to share any new features added. 4. Ask students to answer multiple choice questions and learning attitudes survey.

6. Assessment

Multiple Choice Questions

1. If the following code is used, and the user presses Button1, what is displayed in Label1?



A. myList

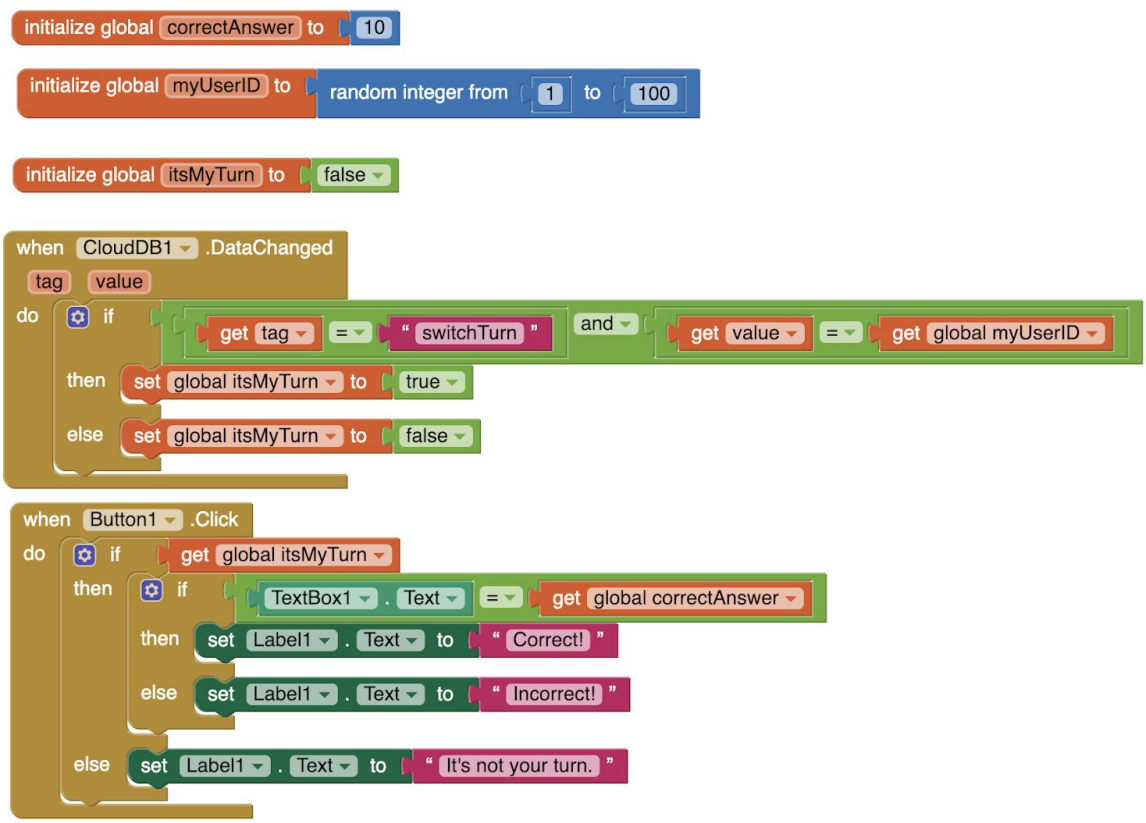
B. 10

C. 4

D. 15

Answer: D

2. For the next question, use the following code blocks.



A player starts playing the app, and their randomly generated userID is 243. They play the game and get a DataChanged event, with a tag "Score" and a value "243". The player enters the number 5 into TextBox1 and then presses Button1. What appears in Label1?

A. Correct!

B. Incorrect!

C. It's not your turn.

D. switchTurn

Answer: C

3. The player continues playing the game, and they get a DataChanged event, with the tag "switchTurn" and the value "243". The player enters the number 10 into TextBox1 and then

presses Button 1. What appears in Label1?

A. Correct!
B. Incorrect!
C. It's not your turn.
D. switchTurn

Answer: A

Survey of learning attitudes

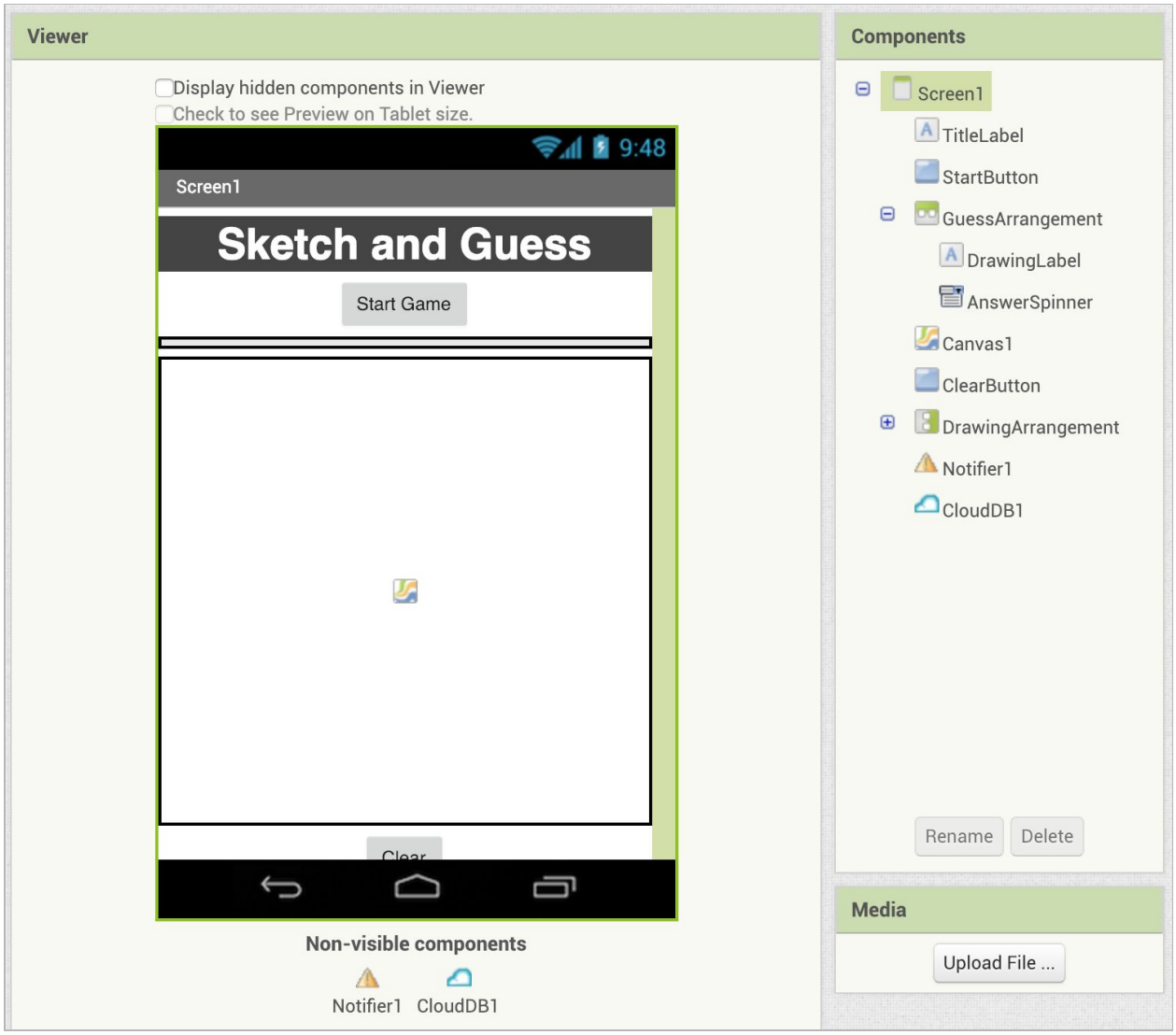
In order to evaluate students' attitude, perception, and understanding towards coding, students are required to finish a 5-point scale survey below by putting a "✓" in the appropriate box.

After completion of this unit, I think…	**Disagree**	**Somewhat disagree**	**Neutral**	**Somewhat agree**	**Agree**
Learning how to make apps makes me want to learn more about coding.					
I feel more connected to the technology around me when I make apps.					

I am excited to share this app with friends and family.					

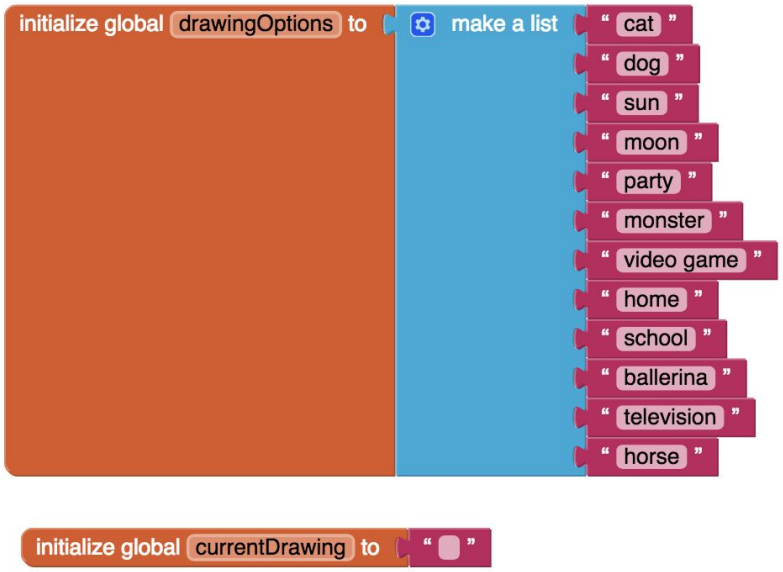
7. Screen Design and Code

Designer

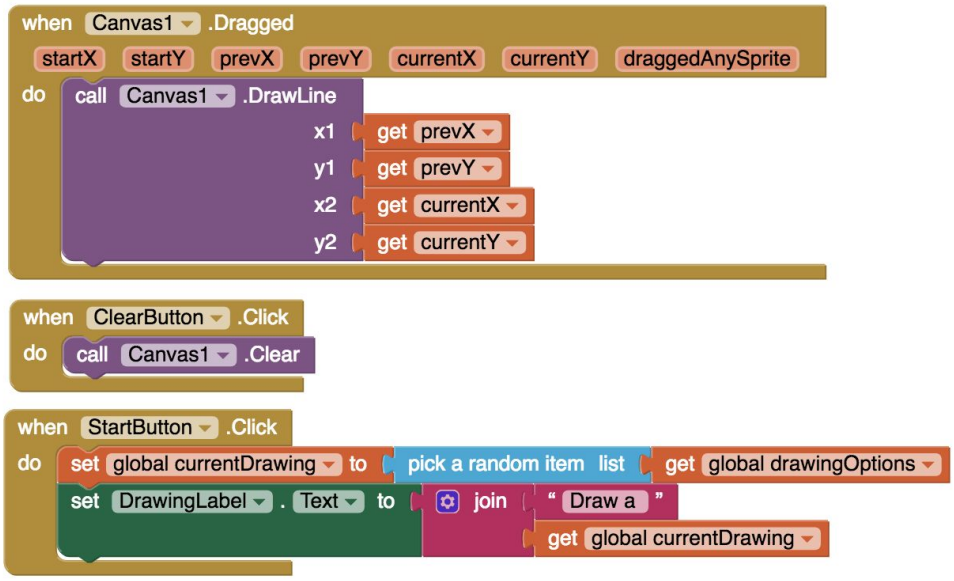


Blocks

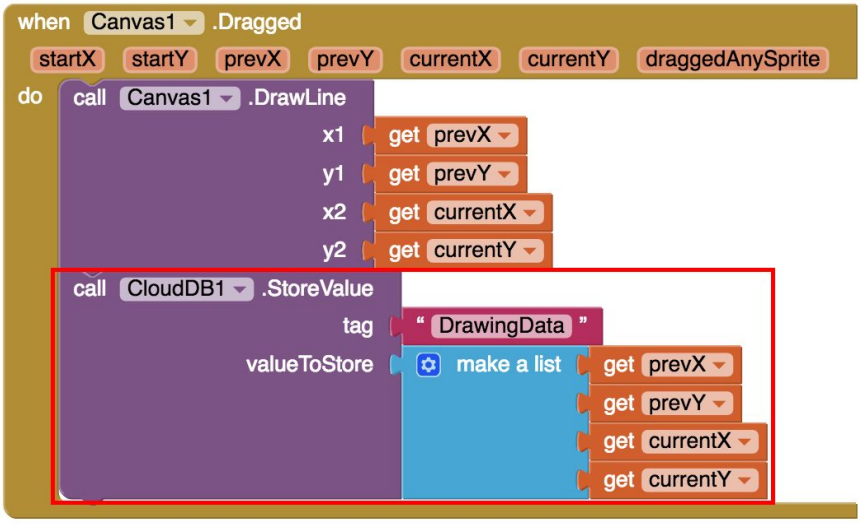
Template

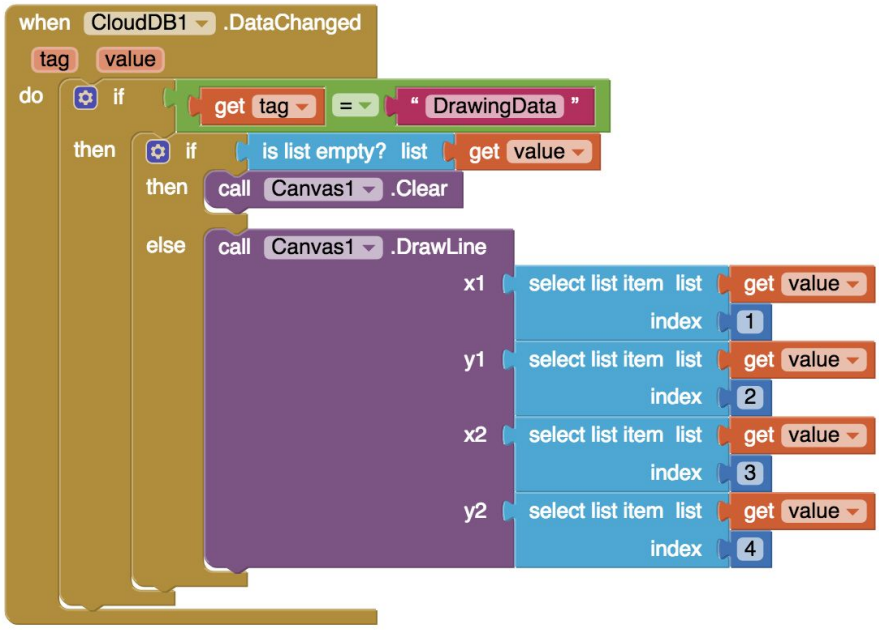


Checkpoint 1 (added to above blocks)



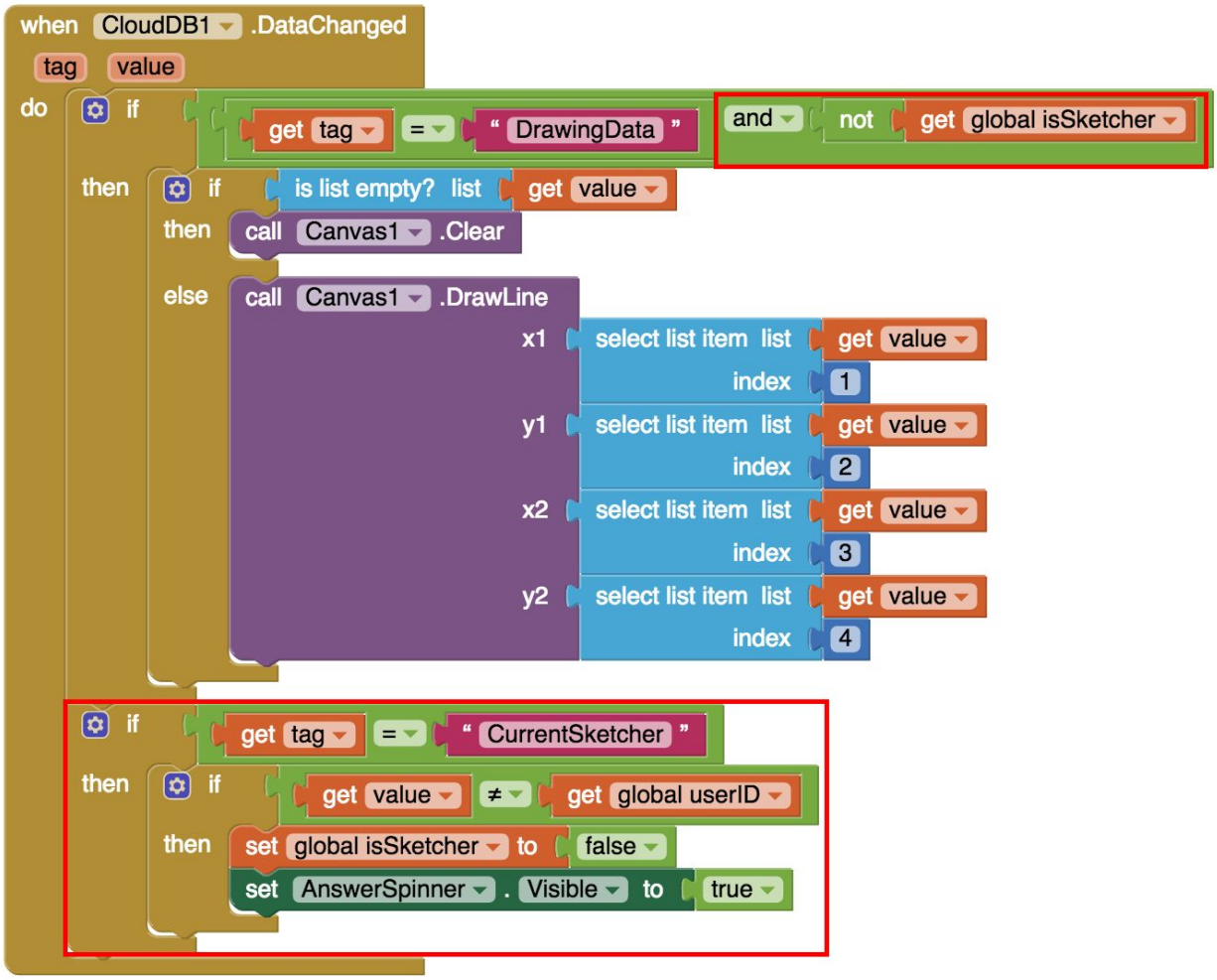
Checkpoint 2 (added to above blocks - blocks outlined in red are added to existing events)



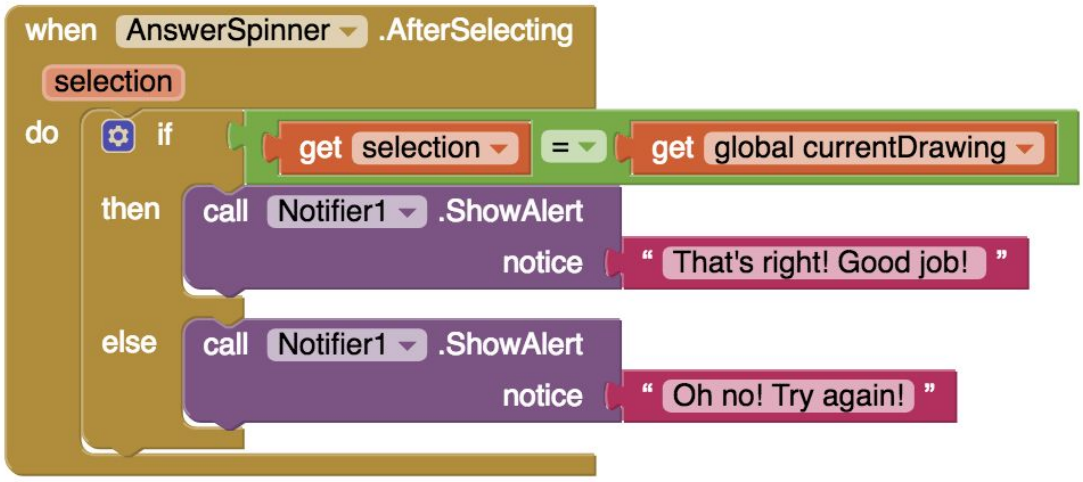


Checkpoint 3 (added to above blocks - blocks outlined in red are added to existing events)

```
initialize global userID to random integer from 0 to 999999

initialize global isSketcher to false

when Screen1 .Initialize
do set AnswerSpinner . Elements to get global drawingOptions

when StartButton .Click
do set global currentDrawing to pick a random item list get global drawingOptions
   set DrawingLabel . Text to join " Draw a " get global currentDrawing
   set global isSketcher to true
   call CloudDB1 .StoreValue
      tag " CurrentSketcher "
      valueToStore get global userID
   set AnswerSpinner . Visible to false

when Canvas1 .Dragged
   startX startY prevX prevY currentX currentY draggedAnySprite
do if get global isSketcher
   then call Canvas1 .DrawLine
      x1 get prevX
      y1 get prevY
      x2 get currentX
      y2 get currentY
      call CloudDB1 .StoreValue
         tag " DrawingData "
         valueToStore make a list get prevX
                                  get prevY
                                  get currentX
                                  get currentY

when ClearButton .Click
do if get global isSketcher
   then call CloudDB1 .StoreValue
      tag " DrawingData "
      valueToStore create empty list
      call Canvas1 .Clear
```



Checkpoint 4: (added to above blocks - blocks outlined in red are added to existing events)



Challenge: (added to above blocks - blocks outlined in red are added to existing events)

```
when Slider1 .PositionChanged
  thumbPosition
do  set Canvas1 . LineWidth to  get thumbPosition
```

```
when CloudDB1 .DataChanged
  tag  value
do  if  get tag = " CurrentDrawing "
    then  set global currentDrawing to  get value
    if  get tag = " DrawingData "  and  not  get global isSketcher
    then  if  is list empty?  list  get value
          then  call Canvas1 .Clear
          else  set Canvas1 . PaintColor to  select list item  list  get value
                                                                 index  5
                set Canvas1 . LineWidth to  select list item  list  get value
                                                                 index  6
                call Canvas1 .DrawLine
                                x1  select list item  list  get value
                                                         index  1
                                y1  select list item  list  get value
                                                         index  2
                                x2  select list item  list  get value
                                                         index  3
                                y2  select list item  list  get value
                                                         index  4
    if  get tag = " CurrentSketcher "
    then  if  get value ≠  get global userID
          then  set global isSketcher to  false
                set Spinner1 . Visible to  true
                set DrawingLabel . Text to  " Guess the drawing. "
                set DrawingArrangement . Visible to  false
```

Appendix 1
Unit 9 Teacher's Guide: Lesson 1

Learning Objectives

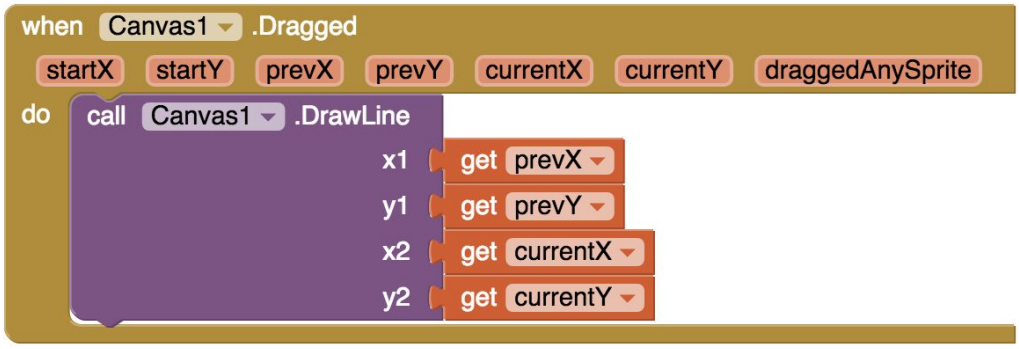
At the end of this lesson, students should be able to:

1. Draw using the Canvas component in App Inventor.

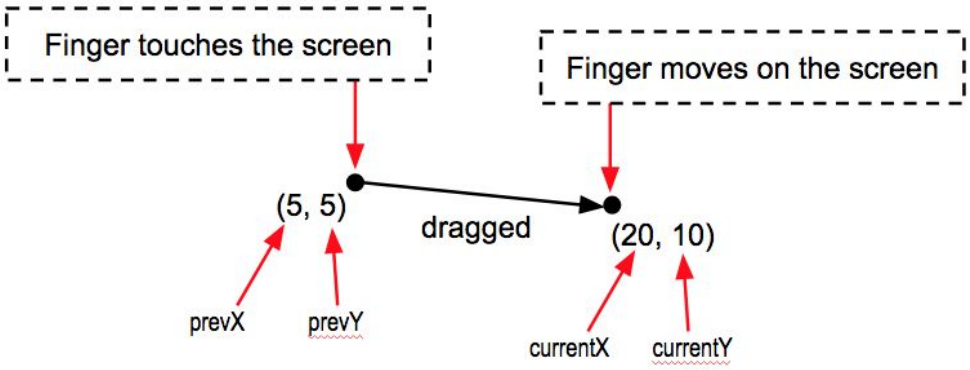
Lesson Outline

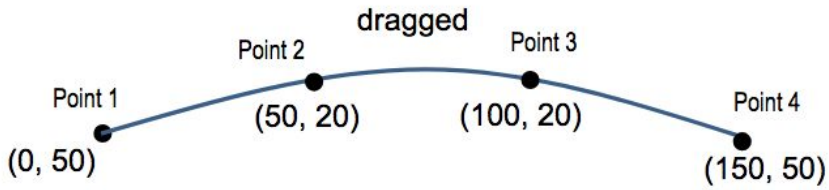
Introduction to Drawing with Canvas Component (10 minutes)

1. Demonstrate a simple drawing app (SketchandGuess_checkpoint1.aia).
2. Introduce drawing on a Canvas
    a. Show a Canvas component added in the Designer
    b. Demonstrate the Canvas.Dragged event block and Canvas.DrawLine blocks.



    c. Explain how a series of lines create a single line.





3. Ask students to fill out answers on pages 2 and 3 in Student Guide Part 1.

Coding of Simple Drawing App (30 minutes)

1. Ask students to pair up and work using the Pair Programming model to build a drawing app with the help of Student Guide: Lesson 1. The result should be a simple drawing app that allows the user to draw with a black line on the screen.
2. Make sure students test and debug the app using the MIT AI2 Companion.

Wrap-up (5 minutes)

1. Review Canvas and drawing features learned in this lesson.
2. Ask students:

a. “Have you played any similar games before?” (Pictionary)
b. “How can you make this app work like Pictionary?”
    - Hopefully students will relate the use of CloudDB to pass information between devices.

Appendix 2
Unit 9 Teacher's Guide: Lesson 2

Learning Objectives

At the end of this lesson, students should be able to:

1. Use CloudDB to pass drawing data between mobile devices.
2. Demonstrate understanding of nested if statements as a way to test for multiple conditionals.
3. Test an App Inventor app through the Build menu, to produce an apk that is installed on a mobile device.
4. Work collaboratively to build and test a multiplayer game.

Lesson Outline

Review of Drawing App (5 minutes)

1. Review the drawing app by asking students how they can play this game with another person on a single device. (One person draws and the other person watches and tries to guess what the first person has drawn).
2. Ask students, "How can you play this game with other people on different devices?"
3. Teacher demonstrates the CloudDB version of the drawing app (SketchandGuess_checkpoint2.aia) to show Sketch and Guess between two devices

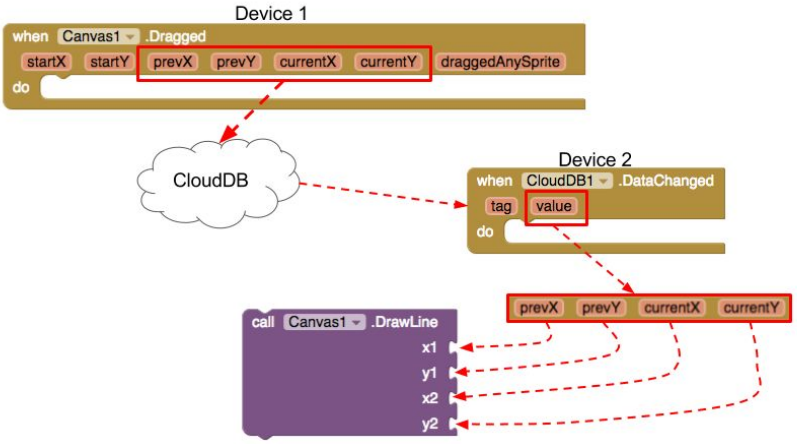
Demonstration of Drawing Using CloudDB (10 minutes)

This unplugged activity will help students to strengthen their understanding of how CloudDB works, and how it can be used to send drawing information to other users.

Use the *CloudDB Unplugged Activity Storing Drawing Data* document to run the activity. One student, the Sketcher, will draw a polygon shape on a grid. With each line they draw, they will write the coordinates of the starting and ending point of the line segment, and pass it to a messenger who will deliver it to CloudDB. CloudDB will then hand it back and ask them to deliver it to the Guesser. The Guesser will take the numbers, and use the information to draw the same line segment on their paper. In the end, they should both have the same figure on their papers.

Adding CloudDB Component to the App (25 minutes)

1. Explain that whenever a user draws something on their device, they will also save the x,y coordinates for the start and end of the drag event to CloudDB. Other users will take that information and use it to draw the same line on their device.



   Because there are 4 numbers (prevx,prevy,currentx,currenty) to be sent, the data will be put in a list, to be saved under a single tag

2. Ask students to work using the Pair Programming model, following Student Guide: Lesson 2, to add CloudDB to their apps. The goal of for this lesson is that if one person

draws something on their app, it will appear on the other person's device.

3. Explain to students that in order to test their app, they will have to download an apk to each partner's device.
    a. Show students how to create an apk with the QR code option. From the Build menu, select "App (Provide QR code for apk)".
    b. When the QR code appears (it might take a few minutes), both partners install the apk on their devices.
    c. Students should test the app. One student will draw and the other can see what is being drawn on their device.

Wrap-up (5 minutes)

Ask students how they might make this into a game, where one person draws and other players guess. What is needed in the app to make the current app into a game? Students might have some suggestions for deciding who is the Sketcher and who is the Guesser. In Part 3, there is one solution for making this into a game, much like Pictionary.

Appendix 3
Unit 9 Teacher's Guide: Lesson 3

Learning Objectives

At the end of this lesson, students should be able to:

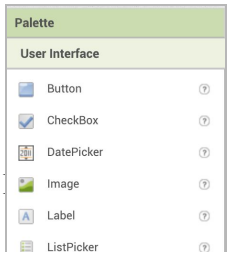
1. Demonstrate understanding of how to use the boolean operators, "not" and "and", in addition to nested if blocks, to create more complex conditionals.
2. Use the Spinner component to allow for app users to make a choice.
3. Use CloudDB to send drawing information between devices in order to make a working Sketch and Guess app.

Lesson Outline

Review and Introduction to Lesson (10 minutes)

1. Review where the app stands now. Users can draw and others can see what they are drawing on their devices.
2. Ask the whole class what can be improved, and how they might make this into a Sketch and Guess game, where one person draws and other players guess.
3. The teacher demonstrates and explains the new function of the game: a "Sketcher" draws a randomly displayed word from a list of simple words included in the template. Other players may watch the drawing and guess what is being drawn. (SketchandGuess_checkpoint3.aia)
4. Show students how to add a Spinner component, and demonstrate how it works.

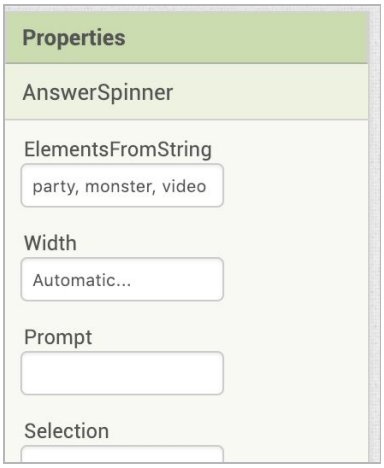
   It appears in the User Interface drawer.



It is similar to ListView and ListPicker components

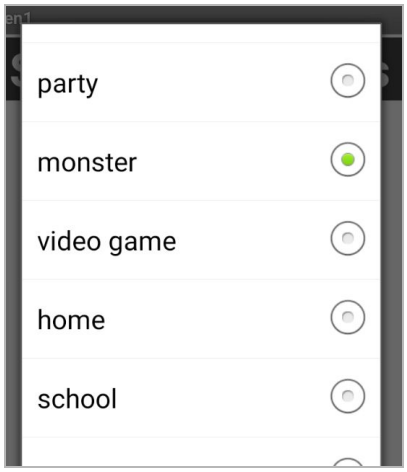
You can set the elements of the Spinner in the

Properties panel in the Designer.

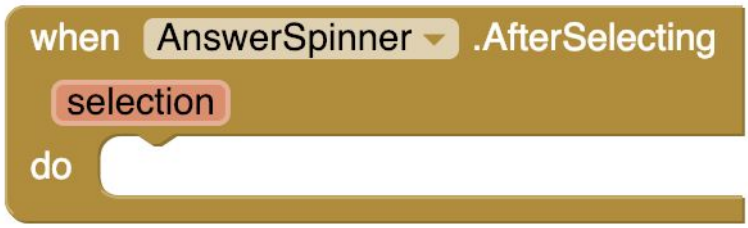


Or you can set it programmatically.

It appears a a list with radio buttons as choices.



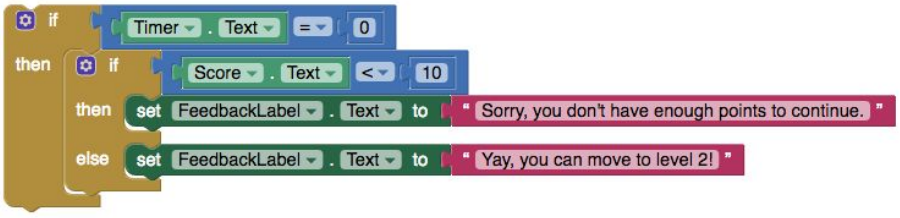
After the user makes their choice, there is an AfterChoosing event. App inventors can test

for the 

which selection user chose.

New Conditional elements (10 minutes)

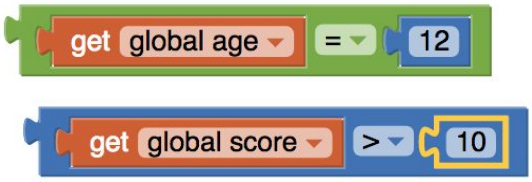
1. Explain to students they will need to do more complex conditionals in this app. They have already use if-then, if-then-else, and if-then-else-if.
2. They have also used nested if blocks. Review how nested if-blocks work.



This conditional first tests if the Timer.Text = 0. If that is true, then it continues to the inner if-then-else statement. If Score.Text <10, it sets FeedbackLabel.Text to "Sorry, you don't have enough points to continue." Else (if Score>=10), FeedbackLabel.Text is set to "Yay, you can move to level 2!". If Timer.Text does not = 0, then nothing happens.

3. Explain boolean operators (**not** and **and**).

   Introduce to students to boolean operators (and, or, and not). In this unit, students will use **and** but it is worth showing them the different operators.



If you want to check if two things are both true, use AND. In this case, we have a game keeping track of time and score.

If you want to check if either of two things is true, use OR. Here we are checking for either an only child, or someone with more than 4 siblings.

And then you can combine AND and OR, depending on what you are testing.

Ask students to read and complete the ***Advanced Conditionals Worksheet.***

4. Explain boolean variables.

   Variables can be defined as boolean, meaning that their value can be either true or false. If that is the case, they be tested whether their value is true or false in a conditional statement. For example, in this app, students will use a variable *isSketcher* to determine which player is currently drawing. For the Sketcher, the variable will be set to true, and for the Guesser, it will be set to false.

To test if the current player is the Sketcher, a conditional will be used. Note that you can either test if *isSketcher* = true, or you can just use *isSketcher* as the condition itself in the if statement. Because the boolean variable itself can be true or false, it can become part of the if conditional statement.

is equivalent to

Coding the Sketch and Guess App (20 minutes)

Following the "Student Guide: Lesson 3", students add code to their apps to make a partially working "Sketch and Guess" game. Remind students that they are using the Pair Programming model, and to switch "driver" and "navigator" roles every 10-15 minutes.

Wrap-up (5 minutes)

1. Ask students, "What is still missing from the app?" Hopefully students will offer that the

Guesser currently has no way of finding out if their guess is correct or not.

2. Explain to students that they will implement guess checking in lesson 4.

Appendix 4
Unit 9 Teacher's Guide: Lesson 4

Learning Objectives

At the end of this lesson, students should be able to:

1. Incorporate a conditional statement to notify a user if their guess is correct or not.

Lesson Outline

Introduction to Lesson (5 minutes)

1. Ask students what is missing from the existing Sketch and Guess app. Students should acknowledge that the Guesser cannot tell if their guess is correct or not.
2. Teacher demonstrates the app with guess checking (SketchAndGuess_checkpoint4.aia)
3. Explain to students that they will implement guess checking in this lesson by adding code to provide feedback to users when they guess. This should be familiar to students, as they have used if statements and the Notifier component in previous game apps to notify the user whether they have won or not.

Coding New Feature (35 minutes)

1. Following *Student Guide: Part 4*, students add guess checking to their apps.
2. Student groups test the app for the new functionality.

Any student pairs who complete Lesson 4 coding may attempt the *Student Guide: Challenge* where they can add color buttons and a slider to change the line width in the app.

Wrap-up (5 minutes)

1. Check in with students to see where they are with the app. Consider how much time will be needed by student groups to complete the app.
2. Explain that students have one more lesson to complete the app, try the Challenge, or add a different feature themselves.

Appendix 5
Unit 9 Teacher's Guide: Lesson 5

Learning Objectives

At the end of this lesson, students should be able to

1. Collaboratively test and debug an app until it works correctly and as expected.
2. Add a new feature to the standard app, such as color, line width, or images.

Lesson Outline

Introduction to Lesson (5 minutes)

1. Check in again with student groups to see where they are with the app..
2. Students can either finish the app if they have not completed it, they can try the *Student Guide: Challenge*, or they can add their own new feature. There are some suggestions at the end of the Challenge.

Coding (25 minutes)

1. Any students still working on Parts 1-4 may continue working to complete the standard app.
2. Students can choose to try the Challenge, adding color buttons and a slider to change the paintbrush size.
3. Students can also choose to extend the app by adding their own feature.

Wrap-up (15 minutes)

1. Review the use of CloudDB in this unit.
2. Review nested if statements, boolean variables and operators.
3. Ask students to reflect on the app, and ask for volunteers to share any new features added.
4. Ask students to answer multiple choice questions and learning attitudes survey.