Two-Button Game App

1. Synopsis

In this unit, students will create a simple two-button game that introduces students to MIT App Inventor's CloudDB functionality. This app will review use of App Inventor, and give students an opportunity to work more with the Clock component. Students will work in pairs on this unit. In the first lesson, students will create the two-player game for use on one tablet. In the second lesson, students learn how to share data via CloudDB and will add CloudDB to the app so that students can play against each other from their own tablets. Students will also apply their computational thinking practices of being incremental and iterative, and testing and debugging, using a step-by-step approach to development of their app.

2. Learning Objectives

After completing this unit, students will be able to:

1. Build a simple app that uses CloudDB.
2. Add multiple components and use Arrangement components to organize a complex user interface involving several components for an app.
3. Use the Clock component to add a timer to a game app.
4. Demonstrate understanding of CloudDB and storing and retrieving data from the cloud.
5. Work collaboratively to build and test a working app.

3. Mapping with the Computational Thinking Framework

These tables show the alignment of this unit with the intended learning outcomes for the computational thinking framework. The entries in the table indicate the expected relevance of the unit to each outcome:

✓✓✓ : High relevance
✓✓ : Some relevance
✓ : Low relevance

Computational Thinking Concepts

Unit 8: Two-Button Game		
1. Sequences	✓	Because a timer is involved, the order of events is important.
2. Events	✓✓✓	Three events are used: CloudDB.DataChanged, Button.Click, and Clock.Timer.
3. Repetition	✓✓	The Clock Timer event is repeated every time the timer is triggered.
4. Conditionals	✓	A conditional is used to test for tags in CloudDB.DataChanged.
5. Parallelism	✓✓	A clock allows events to happen simultaneously, as does CloudDB.
6. Naming	✓	Descriptive naming of components is necessary.
7. Operators	✓	Math operators are required to make the app function correctly.
8. Manipulation of data and elementary data structures	✓✓✓	Students use CloudDB in the second lesson.

Computational Thinking Practices

Unit 8: Two-Button Game		
1. Reusing and remixing		
2. Being incremental and iterative	✓✓	Students will build the app incrementally over the four class periods.
3. Abstracting and modularizing		
4. Testing and debugging	✓✓	Testing and debugging will be done using multiple devices and apks.
5. Algorithmic thinking		

Computational Thinking Perspectives

Unit 8: Two-Button Game		
1. Expressing		
2. Connecting	✓	Students connect with their real life by making a simple game.
3. Questioning	✓✓	Students will face challenges learning a new component. Adding CloudDB adds a new level of questioning of how to expand across apps rather than being limited to one device.
5. Computational identity	✓✓	The introduction of CloudDB provides them with a whole new set of skills to boost their individual computational identity.
6. Digital empowerment	✓✓	Students feel empowered by building an app that works on multiple devices.

4. Mapping with the CSTA Standards

This table shows the alignment of this unit with the intended learning outcomes to the CSTA CS Standards. The entries in the table 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)	CloudDB unplugged activity demonstrates data transmission.
2-DA-07	Represent data using multiple encoding schemes. [C] DA: Storage [P] Abstraction (all)	Data is stored in CloudDB.
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)	Students use compound conditionals in CloudDB1.DataChanged event.
2-AP-18	Systematically test and refine programs using a range of test cases. [C] AP: Program Development [P] Testing (6.1)	Testing is done with two devices using apk files.

5. Learning Prerequisites

Students should have completed the previous units of this App Inventor curriculum.

6. Lesson Plan

This unit consists of four 45 minute lessons.

Lesson 1

Time	Activity
10 min	**Introduction to Unit** 1. Demonstrate the Two-button Game App (TwoButtonGame.aia). 2. Remind students about the use of Layout components to have more control over arrangement of other UI components. 3. Explain that in part 1, students will make a simple app where two players compete by clicking a button on the same device.
30 min	**Coding** 1. Students work in pairs to complete *Student Guide: Part 1*, which is a basic two button game played on a single device. Alternatively, they can follow the Youtube video for Part 1: (https://youtu.be/xhbYVOh9SMA). 2. Once student pairs complete part 1 and show the teacher a working app, they can start on *Student Guide: Part 2*, which adds a timer to the game. The Youtube video for Part 2 is here: (https://youtu.be/IRqdrigBlXw).
5 min	**Wrap-up** 1. Review Layout components. 2. Ask students how they could add a timer to the game.

Lesson 2

Time	Activity
10 min	**Introduction to Clock Timer** 1. Explain that in part 2, students will add a Clock component so there is a countdown timer in the app. 2. Demonstrate the Clock component. a. Properties for setting how often Timer fires. b. Clock.Timer block event
30 min	**Coding** 1. Students work in pairs to complete *Student Guide: Part 2*, which includes a 10 second timer for playing the app. Alternatively, they can follow the Youtube video for Part 2: (https://youtu.be/IRqdrigBlXw). 2. Once student pairs complete part 2 and show the teacher a working app, they can start on *Student Guide: Part 3*, which incorporates CloudDB to allow for play across two devices. The Youtube video for Part 3 is here (https://youtu.be/Jm2lMLCYVbs).
5 min	**Wrap-up** 1. Review Clock component. 2. Ask how students think this could work over multiple devices.

Lesson 3

Time	Activity
15 min	**Introduction to CloudDB** 1. Demonstrate the CloudDB version of the Two-Button Game to show how CloudDB works to update scores on multiple tablets. Have 2 students compete in a game. 2. Explain that we're going to use a new component called CloudDB, which is like TinyDB, but it stores data in the cloud, so people can share data between devices. Explain the similarities and differences between variables, TinyDB, and CloudDB. 3. Do the unplugged activity to explain how CloudDB stores and retrieves data in an app.
25 min	**Coding** Students follow Student Guide: Part 3, which incorporates CloudDB to allow for play across two devices.
5 min	**Wrap-up** Review CloudDB component. Ask students how they could use CloudDB in other apps they might build.

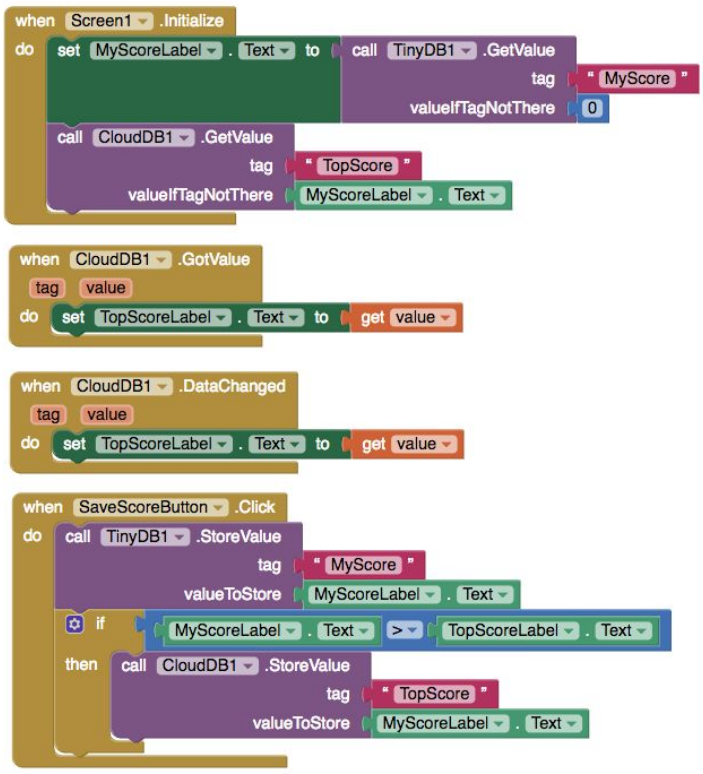
Lesson 4

Time	Activity
5 min	**Introduction to Lesson** Students may either continue with *Student Guide: Part 3* or may continue on to the Challenge to enhance their apps.
30 min	**Coding** Students continue working on the Two Button Game. If they complete a working multi-device app, they can try some of the challenges.
10 min	**Wrap-up** 1. Review CloudDB component. 2. Ask students to share a new feature they have added to their game. 3. Ask students how they could use CloudDB in other apps they might build.

7. Assessment

Multiple-choice questions

Please answer Questions 1-2 with the following code blocks.



You are Player 1, and your neighbor is Player 2. Both of you are running the same app, but on different tablets. The app code is shown below.

1. Player 1 opens the app for the first time. No one else has played it. What is displayed in MyScoreLabel and TopScoreLabel?
   (A) Nothing
   (B) 0 and 0
   (C) 1 and 2
   (D) MyScore and TopScore

(Answer: B)

2. Player 1 scores 25 and presses the SaveScoreButton. Then Player 2 opens the app for the first time. What is displayed in MyScoreLabel and TopScoreLabel on Player 2's tablet?

   (A) Nothing
   (B) 0 and 0
   (C) 0 and 25
   (D) 25 and 25
   (E) MyScore and TopScore

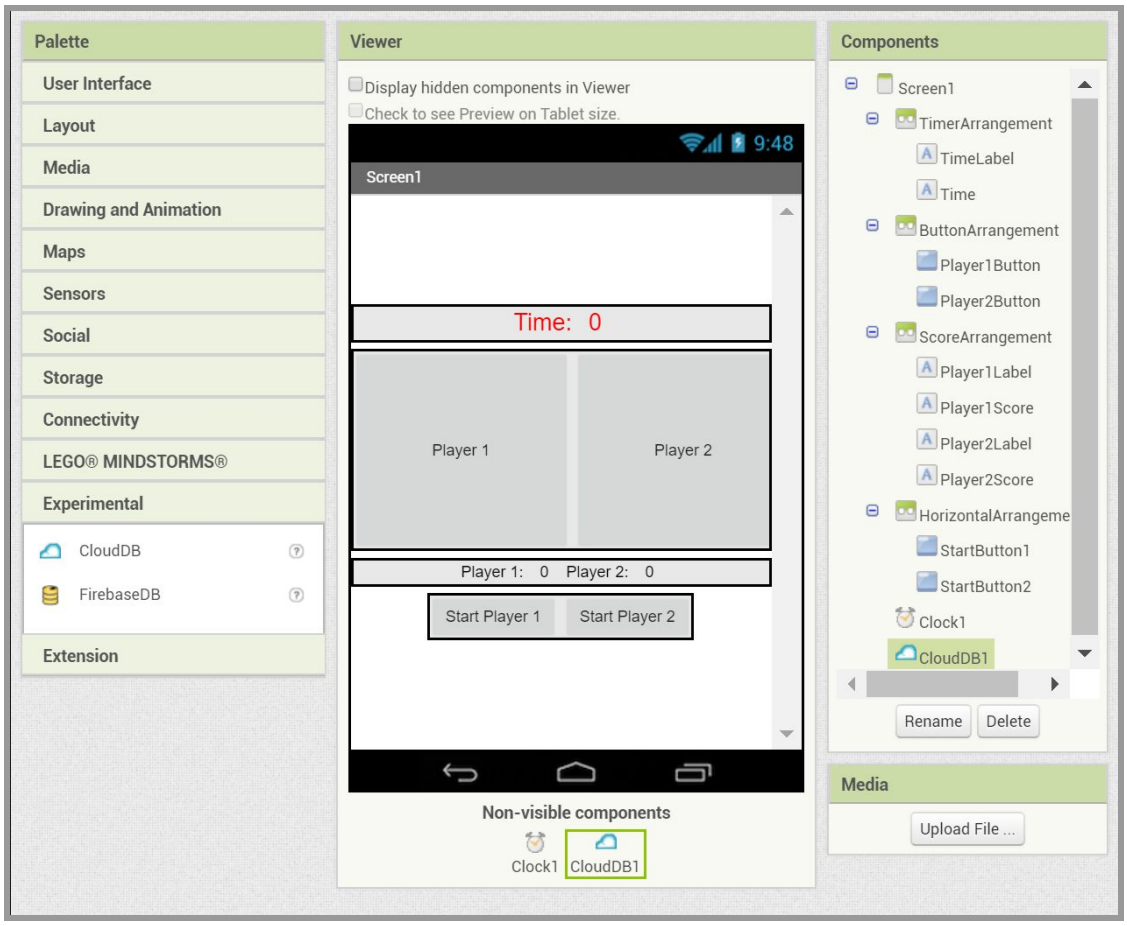
(Answer: C)

Survey of learning attitudes

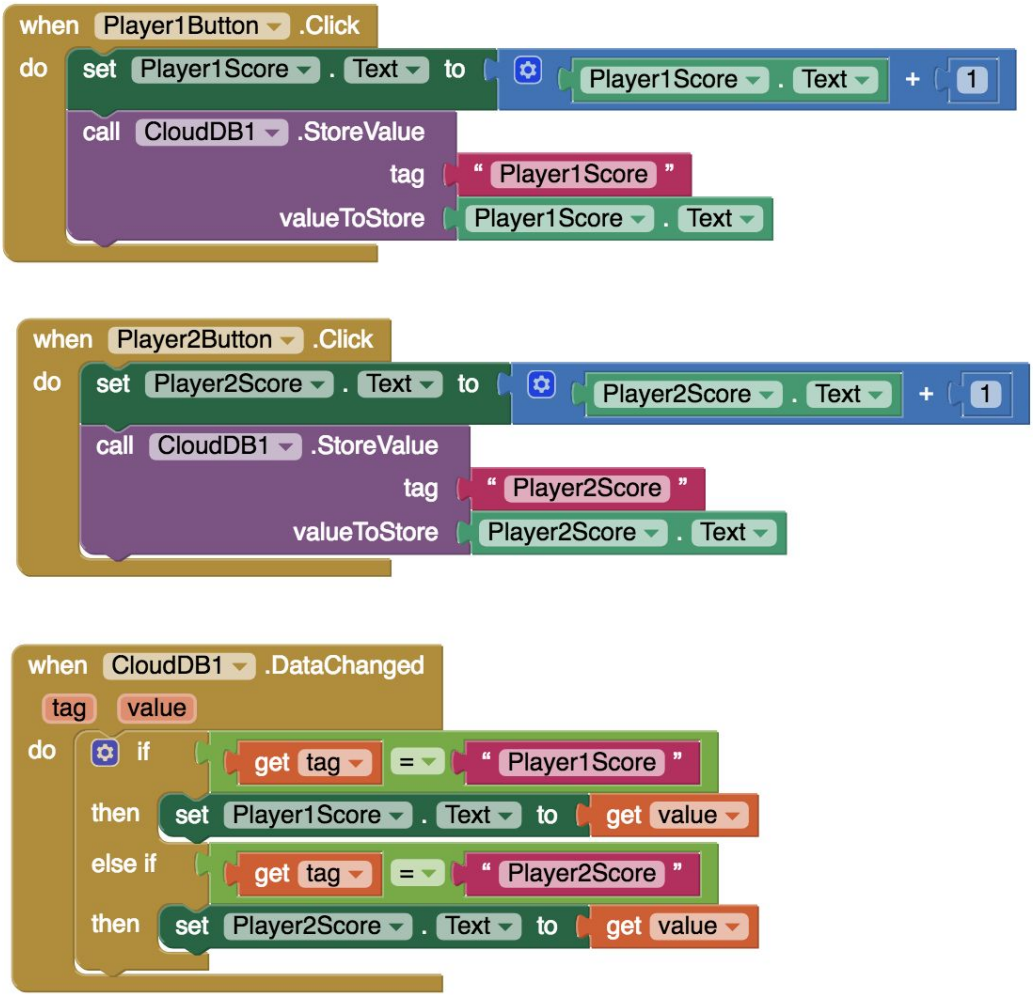
In order to evaluate students' attitude, perception, and understanding towards coding, students are required to finish a 5-point scale survey below by putting a "✓" in the appropriate box.

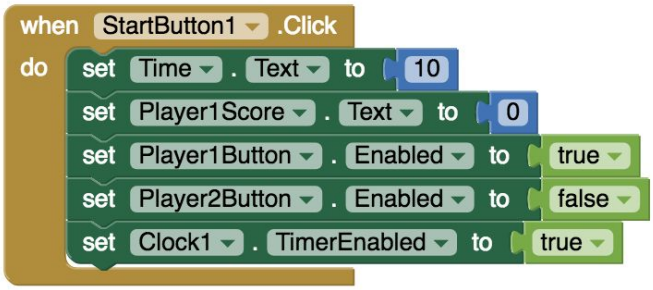
After completion of this unit, I think…	**Disagree**	**Somewhat disagree**	**Neutral**	**Somewhat agree**	**Agree**
Learning how to make apps makes me want to learn more about coding.					
I feel more connected to the technology around me when I make apps.					
I am excited to share this app with friends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

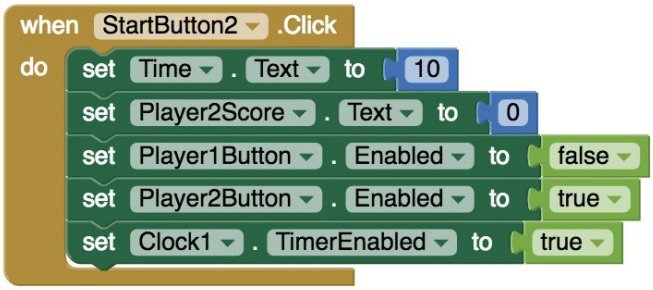
8. Screen Design and Code

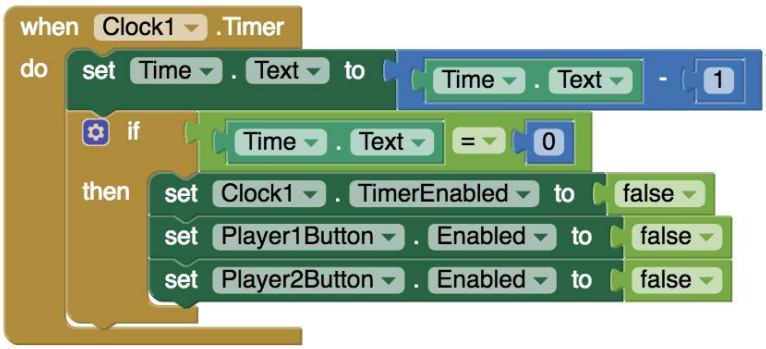


Blocks









Appendix 1
Teacher's Guide: Lesson 1

Learning Objectives

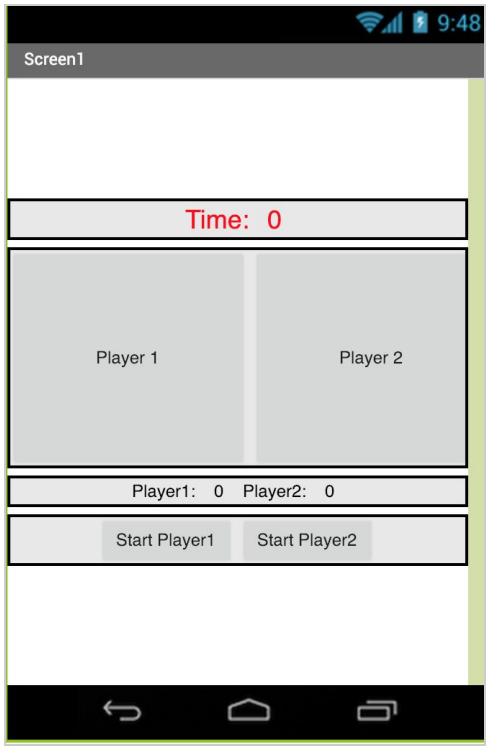
At the end of this lesson, students should be able to:

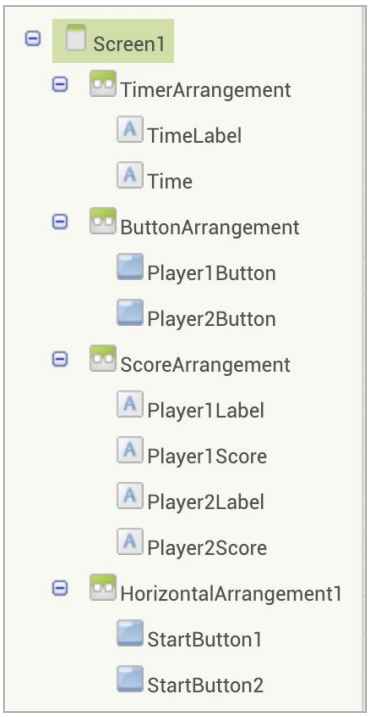
1. Use Layout components to organize elements in an app.
2. Code a simple two button game app.

Lesson Outline

Introduction to Unit (10 minutes)

1. Demonstrate the Two-button Game App (TwoButtonGame.aia).
2. Remind students about the use of Layout components to have more control over arrangement of other UI components.





Note above that there are 4 arrangements, all Horizontal, and within those are a combination of buttons and Labels. Organizing these into Layout Arrangements allows you to place components next to each other instead of the default vertical alignment. You can also hide or show an entire arrangement with the Visible property to hide components that are not currently being used.

3. Explain that in part 1, students will make a simple app where two players compete by clicking a button on the same device.

Coding (30 minutes)

1. Students work in pairs to complete *Student Guide: Part 1*. Alternatively, they may follow the Youtube video (https://youtu.be/xhbYVOh9SMA). Explain that the goal is to create a simple two button game that works on one device. The user must click the left and right buttons as many times as possible. Or two people can play it on the same device, each player pressing one of the two buttons to make it competitive.

2. If students complete Part 1, they can start to work on Part 2, where they add a Clock component to add a 10 second timer to the game. The Youtube video for Part 2 is here. (https://youtu.be/IRqdrigBlXw)

Wrap-up (5 minutes)

1. Review Layout components.
2. Ask how students they this could make work with a countdown timer. Some students may have used a Clock component in earlier units involving games.

Appendix 2
Teacher's Guide: Lesson 2

Learning Objectives

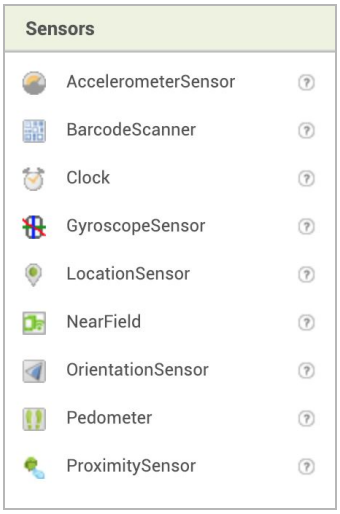
At the end of this lesson, students should be able to:

1. Use a Clock component to include a countdown timer in a game app.
2. Develop an app incrementally and iteratively, by adding new features to an app.

Lesson Outline

Introduction to Clock Timer (10 minutes)

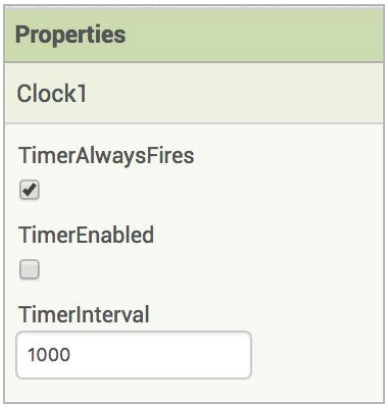
1. Demonstrate the Clock component.
    a. Component is in the Sensors drawer. It's a non-visible component.



b. Properties for setting how often Timer fires:

TimerEnabled lets you enable or disable (start/stop) the Clock.

TimerInterval specifies how often the Timer goes off (triggers the Clock.Timer event). It measures in milliseconds, so 1000 is 1 second.



b. **Clock.Timer** block event. This event is triggered based on the **TimerInterval**.

For example, if **TimerInterval** is 1000, this event is triggered every second.

3. Explain to students that they will add the timer feature to their app with the Clock component.

Coding (30 minutes)

1. Students work in pairs to complete *Student Guide: Part 2*, which includes a 10 second timer for playing the app. Alternatively, they can follow the Youtube video for Part 2 (https://youtu.be/IRqdrigBlXw).

2. Once student pairs complete part 2 and show the teacher an app that includes a working timer, they can start on *Student Guide: Part 3*, which incorporates CloudDB to allow for play across two devices. The Youtube video for Part 3 is here (https://youtu.be/Jm2lMLCYVbs).

Wrap-up (5 minutes)

1. Review Clock component. Make sure students understand the properties and how adjusting them affects how often the Clock.Timer event fires.

2. Ask how students think this could work over multiple devices.

Appendix 3
Teacher's Guide: Lesson 3

Learning Objectives

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

1. Use CloudDB to synchronize data in the same app between devices.
2. Develop an app incrementally and iteratively, by adding new features to an app.
3. Work collaboratively to create a multiplayer game.
4. Create and download apk files to enable testing on multiple devices.

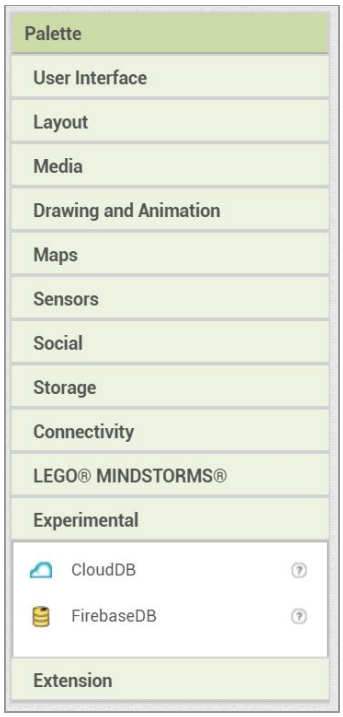
Lesson Outline

Introduction to CloudDB (15 minutes)

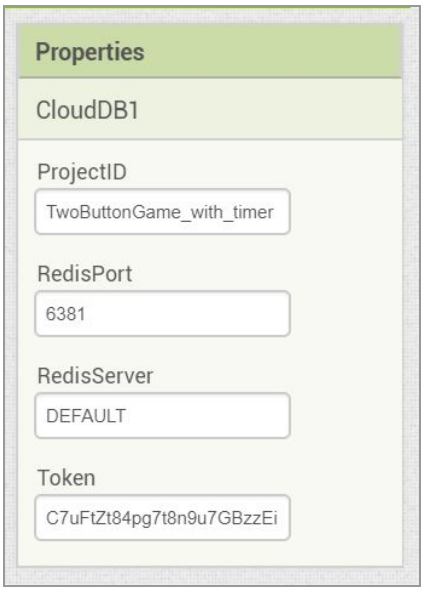
1. Demonstrate the CloudDB version of the Two-Button Game to show how CloudDB works to update scores on multiple tablets. Have 2 students compete.
2. Explain that we're going to use a new component called CloudDB, which is like TinyDB, but it stores data in the cloud, so people can share data between devices. Explain the similarities and differences between variables, TinyDB, and CloudDB.
3. Do the unplugged activity to explain how CloudDB stores and retrieves data in an app.

Teacher's Resource: About CloudDB

CloudDB is under the Experimental drawer in App Inventor.



The Properties panel looks like this:



For most apps, nothing in this window needs to be changed. The **Token** property is automatically filled out with a unique signature when the component is added.

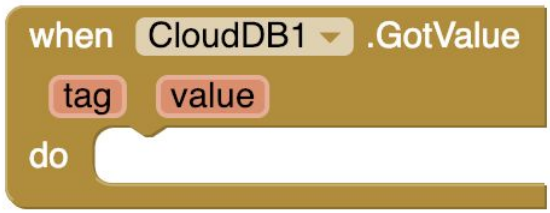
While CloudDB allows apps to share data, the sharing is limited to instances of the same app. That means that if different students build their own apps, even if it's the same code and the same name, the apps cannot share data. Data is shared among everyone using the SAME app. Therefore, student pairs will test together using the same app.

CloudDB is similar to TinyDB in that both store data; one stores it in the cloud for multiple users, and one stores it on the device privately. Both have a StoreValue operation. However, where TinyDB's GetValue is instant, that is not the case with CloudDB. A GetValue in CloudDB requests the data from the cloud, and the value is returned in a GotValue event. Also, whenever anyone else changes data, that generates a DataChanged event.

Store a value:

Ask to get a value:

Triggers GotValue when received:



All devices running the app get notified if anyone updates values in CloudDB:

The chart below outlines the differences between variables, TinyDB, and CloudDB. This chart can be used to demonstrate the differences for students as well.

	Variable	**TinyDB**	**CloudDB**
Storing a new value	initialize global x set global x to	TinyDB.StoreValue	CloudDB.StoreValue
Getting a value	get global x	TinyDB.GetValue returns value immediately	CloudDB.GetValue requests value which is returned via CloudDB.GotValue event
Data Changed (Not by current user)	Not possible	Not possible	CloudDB.DataChanged event
Location of Data	app memory	Device (phone/tablet)	In the cloud (not local)
Accessibility of Data	available when app is running. Erased when app closes	Device user only	Anybody running the same app

Unplugged Activity Details

This simple unplugged activity below mimics a group chat, such as within Facebook Messenger.

Prep

1 student - CloudDB

1 student - “Internet” (message passing)

5 students - chat client users

1 student - Sender

1 student - Late joiner

Post-it notes

Order of Events

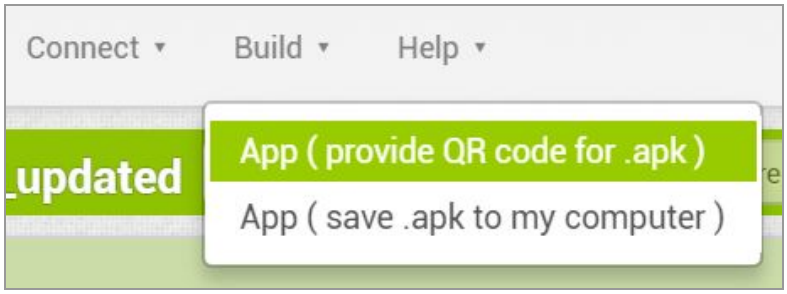
1. Sender sends a message (written on a post-it note) for the group chat. The Internet carries that message to the database. For all messages in this exercise, students can come up with their own messages to share.
    a. This is a **StoreValue** call.
2. Database makes a copy of this message and sends it (via Internet) to all chat client users, so that all other chat client users get a copy of the new message (so everyone gets a post-it note with the sender’s message on it)
    a. This is a **DataChanged** event.
3. Late joiner comes in, and asks for the chat history so he can catch up. Internet takes his request to the database.
    a. This is a **GetValue** call.
4. Database sends back chat history with internet to the late joiner.
    a. This is a **GotValue** event.
5. Several messages should be sent back and forth amongst the chat clients so that students understand how the process of storing and receiving data works.

Coding the App (25 minutes)

Once student pairs complete part 2 and show the teacher a working app, they can start on *Student Guide: Part 3*, which incorporates CloudDB to allow for play across two devices. Alternatively, they can follow the Youtube video for Part 3: (https://youtu.be/Jm2lMLCYVbs).

NOTE: Student pairs should test their games by downloading the apk to two tablets and playing against each other.

a. Instead of connecting to the AI2 Companion, students go to the Build menu, and choose "App (provide QR code for .apk)



b. When the QR code appears, each student should scan the QR code on their tablet, and follow the directions to install the apk file.

c. Once the app is installed, each student should open the app and play the game against each other. One student acts as Player1, and the other acts as Player2.

Wrap-Up (5 minutes)

1. Review CloudDB and storing and sharing data in the cloud.
2. Generate a discussion with students. How could the CloudDB component be use in other apps/games?

Appendix 4
Teacher's Guide: Lesson 4

Learning Objectives

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

1. Use CloudDB to synchronize data in the same app between devices.
2. Work collaboratively to create a multiplayer game.
3. Create and download apk files to enable testing on multiple devices.

Lesson Outline

Introduction to Lesson (5 minutes)

Students may either continue with *Student Guide: Part 3* or continue on to the Challenge to enhance their apps. Make sure students are comfortable testing using apk's instead of using the MIT AI2 Companion.

Coding (30 minutes)

Students continue working on the Two Button Game. If they complete a working multi-device app, they can try some of the challenges.

Wrap-up (10 minutes)

1. Review Clock and CloudDB components.
2. Ask students to share a new feature they have added to their game.
3. Ask students how they could use CloudDB in other apps they might build.