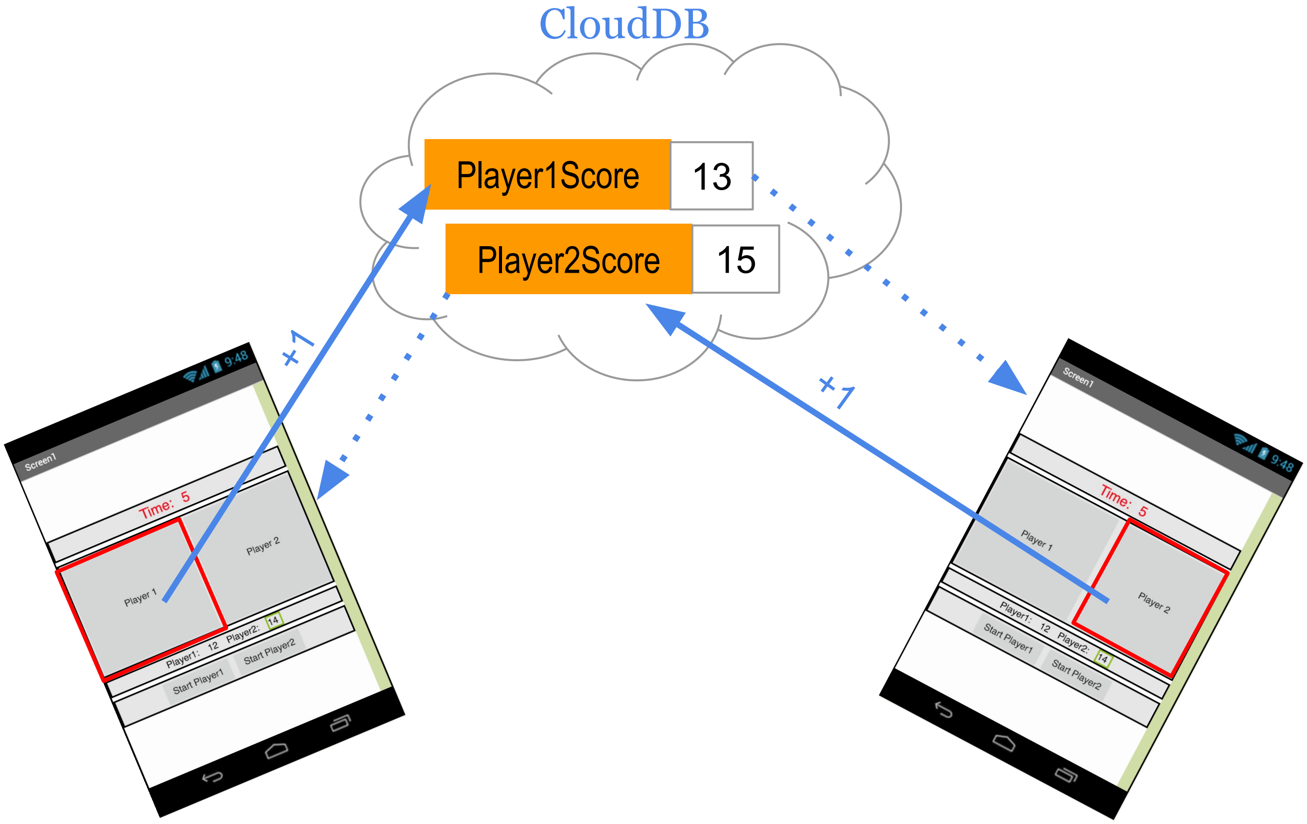
TWO-BUTTON GAME: PART 3

In this lesson, you will be adding functionality to make this a two player game over two devices.

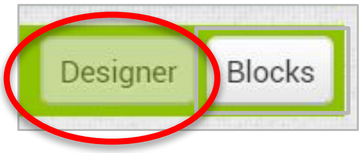
HOW CLOUDDB WORKS

You will use CloudDB in the app to update each player's score on the other player's tablet by adding code to do the following. When a player clicks on their button, they'll store their new score in CloudDB. The DataChanged event will be triggered every time that happens, and that will signal the other player to update the score on their tablet.



START HERE

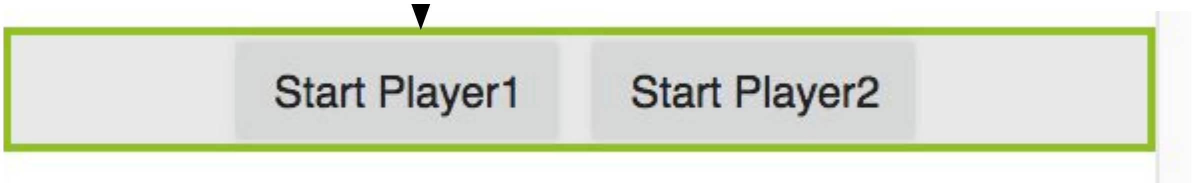
1 Open your TwoButtonGame app, and switch to the Designer.



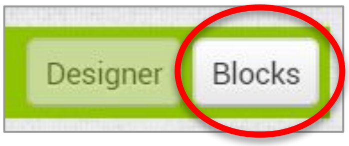
2 Add the components in the table below. **StartButton2** should be added to **ButtonArrangement**, to the right of **StartButton1**.

Drawer	Component	Name	Property	Setting
User Interface	Button	StartButton2	*Text*	"**Start Player2**"
Experimental	CloudDB	CloudDB1		

3 Change the *Text* property for **StartButton1** to "**Start Player1**".



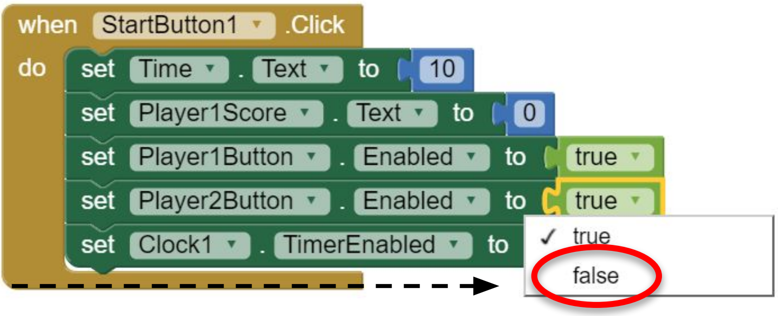
4 Switch back to the Blocks Editor.



UPDATING BUTTON.CLICK EVENTS

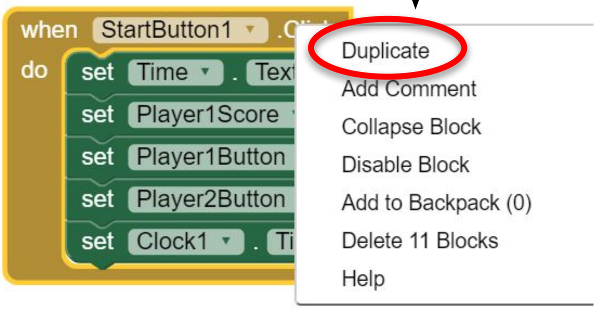
The first thing to do is update the Start buttons. **StartButton1** should be mostly correct already.

5 **Player2Button** should **not** be enabled for Player 1. Change **Player2Button.Enabled** to **false** by clicking on **true** and selecting **false**.



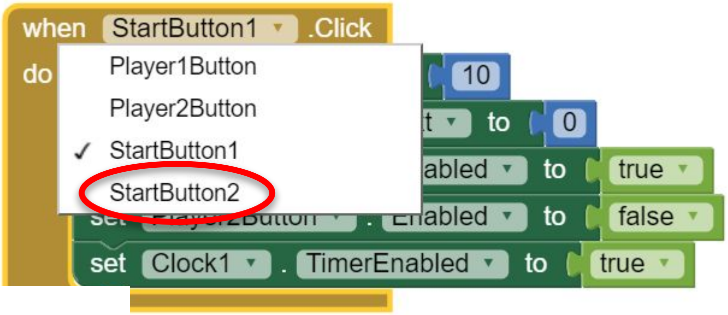
```
when StartButton1 .Click
do  set Time . Text to 10
    set Player1Score . Text to 0
    set Player1Button . Enabled to true
    set Player2Button . Enabled to true
    set Clock1 . TimerEnabled to
```

6 Rather than rewrite all the code for **StartButton2**, you can *Duplicate* **StartButton1.Click** and change it. Right click on **StartButton1.Click**, and select **Duplicate**.



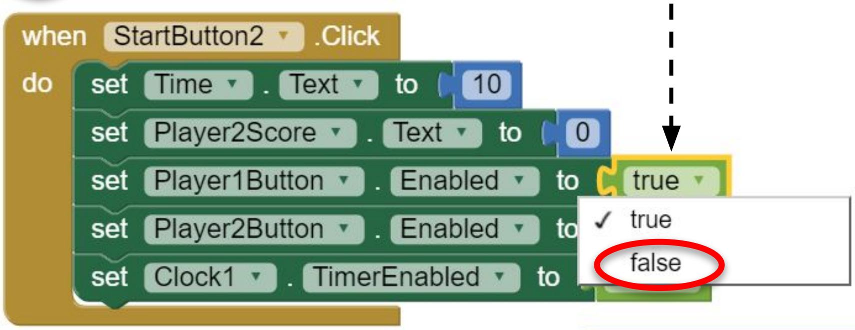
```
when StartButton1 .Cl
do  set Time . Text
    set Player1Score
    set Player1Button
    set Player2Button
    set Clock1 . Ti
```

7 Click the drop down menu of **StartButton1** and change it to **StartButton2**.



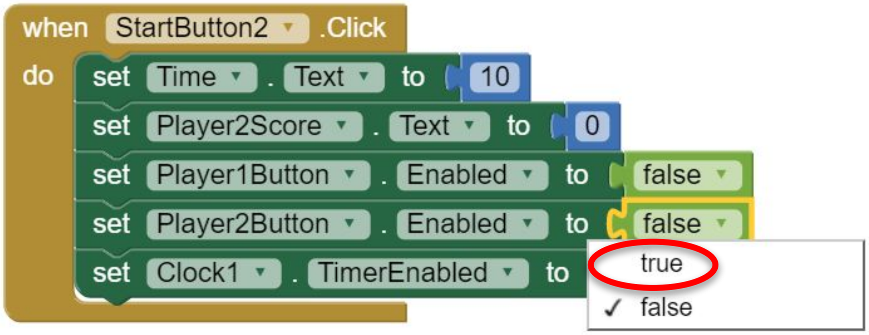
```
when StartButton1 .Click
do                              10
                           t to 0
                       abled to true
                       Enabled to false
    set Clock1 . TimerEnabled to true
```

8 Change **set Player1.Enabled** to **false**.



```
when StartButton2 .Click
do  set Time . Text to 10
    set Player2Score . Text to 0
    set Player1Button . Enabled to true
    set Player2Button . Enabled to
    set Clock1 . TimerEnabled to
```

9 And change **set Player2.Enabled** to **true**.

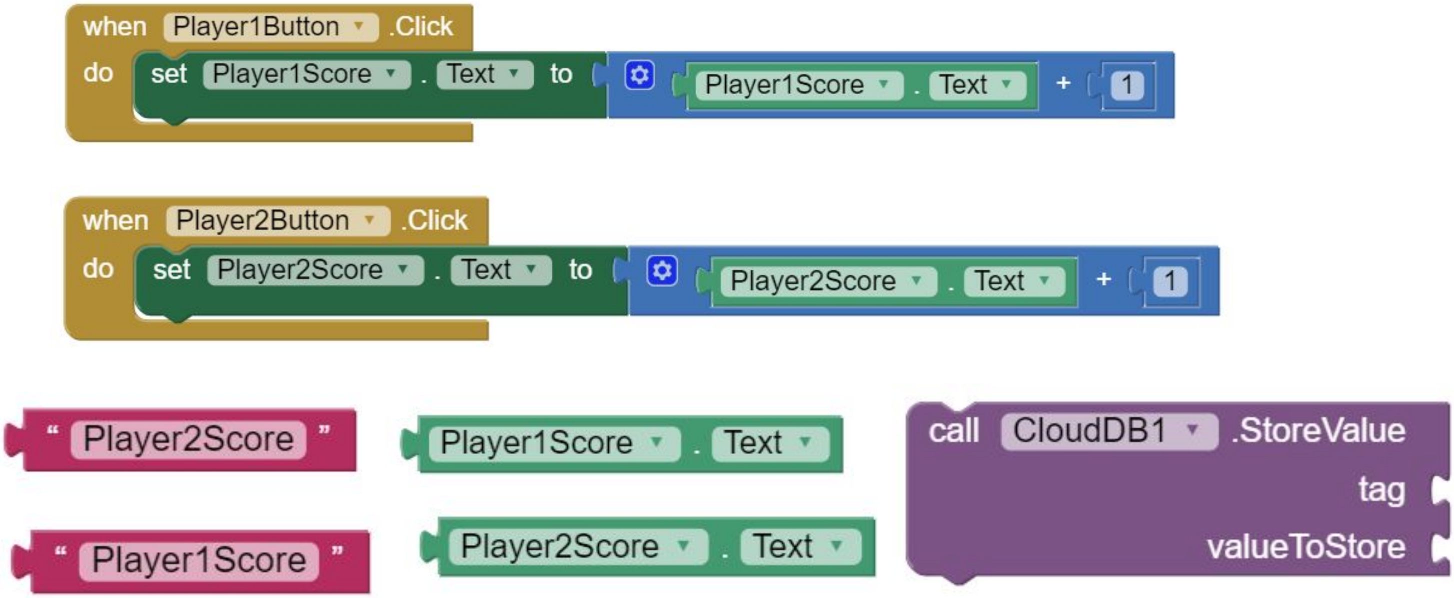


```
when StartButton2 .Click
do  set Time . Text to 10
    set Player2Score . Text to 0
    set Player1Button . Enabled to false
    set Player2Button . Enabled to false
    set Clock1 . TimerEnabled to
```

CLOUDDB STORE VALUE

Every time a player scores a point, save the new score in CloudDB.

10 Update **Player1Button.Click** and **Player2Button.Click** to include a **CloudDB1.StoreValue** block. Make sure you include the corresponding tag and value for the given Player.



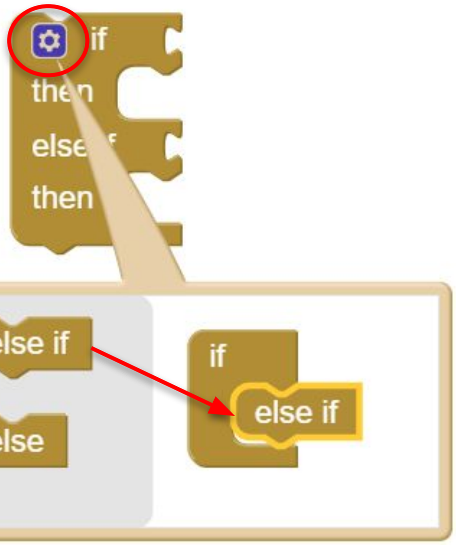
```
when Player1Button .Click
do  set Player1Score . Text to  Player1Score . Text + 1

when Player2Button .Click
do  set Player2Score . Text to  Player2Score . Text + 1
```

```
" Player2Score "        Player1Score . Text      call CloudDB1 .StoreValue
                                                                       tag
" Player1Score "        Player2Score . Text                   valueToStore
```

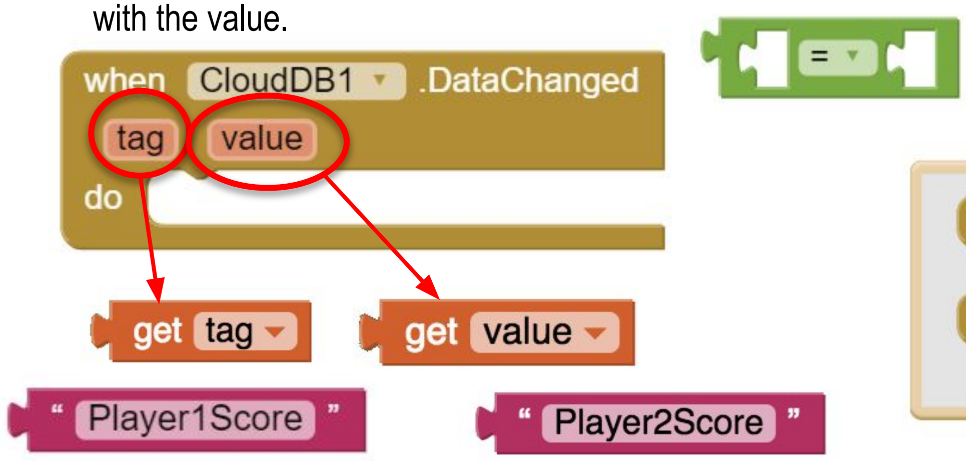
Every time a player's score is updated in CloudDB, that generates a **CloudDB.DataChanged** event that gets triggered on every mobile device running the app.

11 Code the CloudDB1.DataChanged event block. Test which tag has changed, and set the corresponding Player Score with the value.



```
if
then
else if
then
```

```
else if        if
else           else if
```

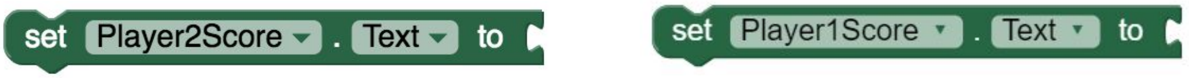


```
when CloudDB1 .DataChanged
  tag  value
do

[ ] = [ ]

get tag        get value

" Player1Score "        " Player2Score "
```

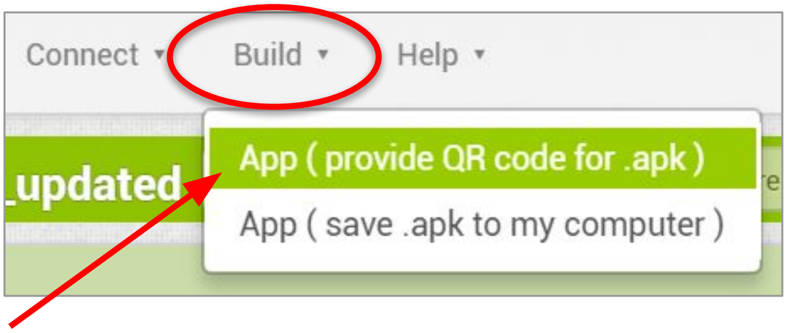


```
set Player2Score . Text to        set Player1Score . Text to
```

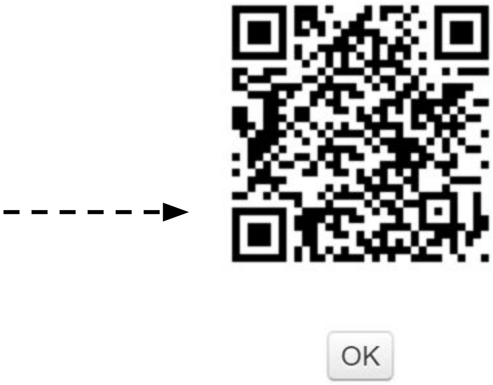
TESTING

Time to test! Because this game app sends scores to the cloud, you will need to test using two devices. You can only test with one AI2 Companion, so you'll make your project an apk file and install the app on two separate tablets.

12 Go to the "Build" menu at the top of the screen, and click on "App (provide QR code for .apk)". This will start the process of building the app so that it can be installed on any tablet.


Connect ▾
Build ▾
Help ▾
_updated
App (provide QR code for .apk)
App (save .apk to my computer)

13 A QR code will appear in a pop-up window once the app is built. When it appears, both you and your partner should scan it using your tablets. Follow the prompts to download and install the .apk on your tablets.



Note: this barcode is only valid for 2 hours. See the FAQ for info on how to share your app with others.

14 Play the game against your partner on your own tablets. Does it work correctly? Do the scores update correctly?

Choose Ways to Extend Your App

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

Add sounds! One for Player1 and one for Player2!

Add a label to display who is currently leading

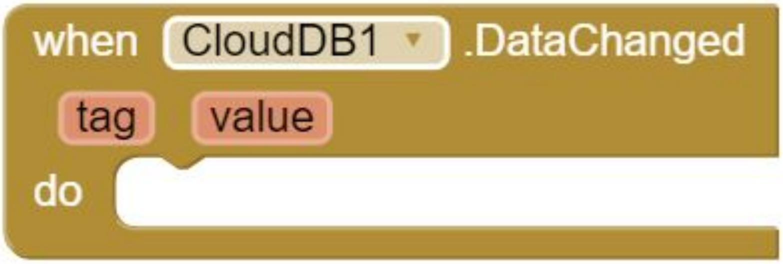
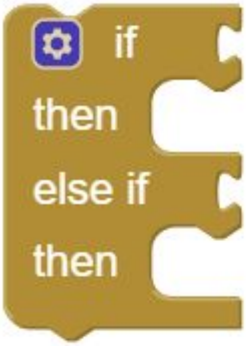
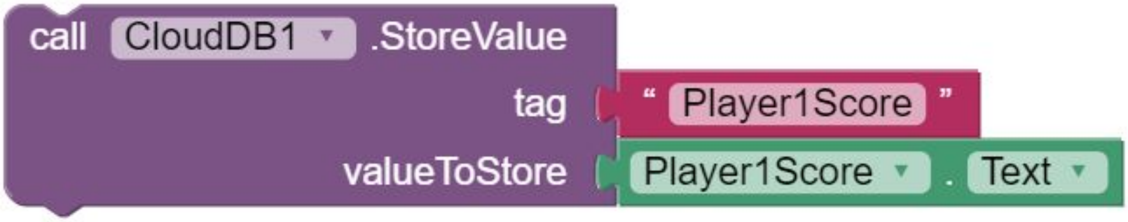
Add a High Score that is saved in CloudDB

What other ideas do you have?

TWO-BUTTON GAME: PART 3

COMPUTATIONAL THINKING CONCEPTS

The following are the Computational Thinking Concepts learned in this lesson.

Two-Button Game
1. Events:  when CloudDB1 ▾ .DataChanged tag value do
2. Conditionals:  if then else if then
3. Data Manipulation:  call CloudDB1 ▾ .StoreValue tag " Player1Score " valueToStore Player1Score ▾ . Text ▾