

# TWO-BUTTON GAME: CHALLENGE

Here are some  
HINTS to help you do  
some of the  
challenges

## ADD SOUND

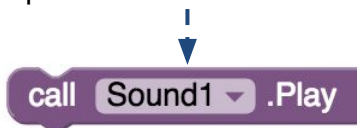
You can use a Sound component from the Media drawer. The Sound component works just like the Player component, but is usually used for short sound files.

1 Add a Sound (or Player) component. Remember that it's non-visible so you won't "see" it on the Viewer.

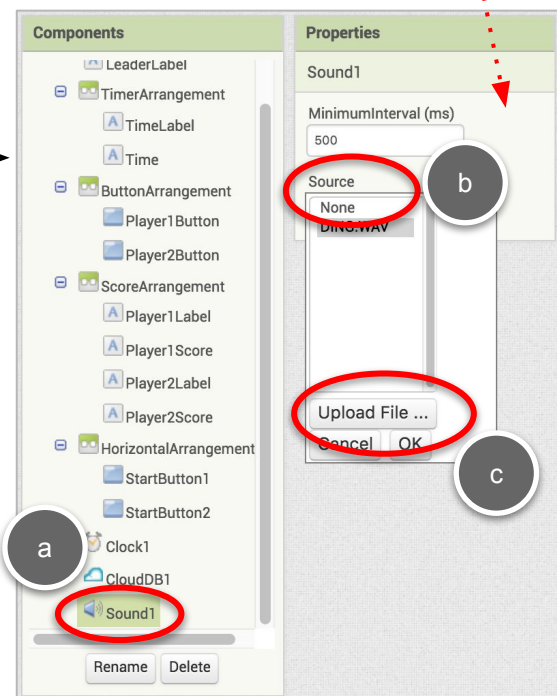
2 Find a sound, like this [DING.WAV](#) and download it to your computer.

3 Upload your sound file and set it as the component's Source file. ----->

4 In the Blocks Editor, add Sound1.Play where you want the sound to play.  
Perhaps when a user clicks their button?



5 You can use more than one Sound component if you want different sounds to play for each Player.

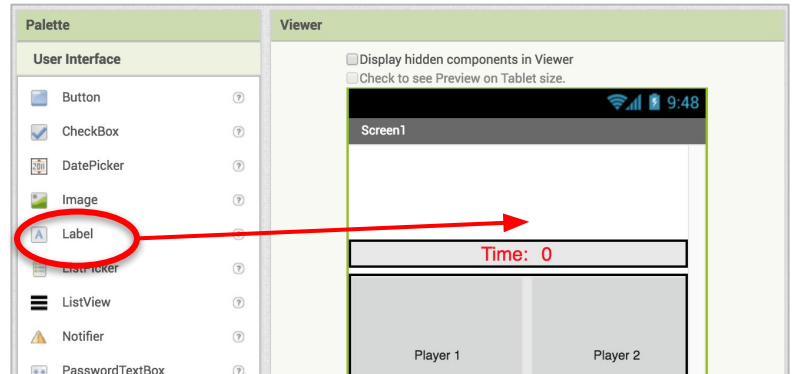


## DISPLAY LEADERBOARD

How about showing the players who is in the lead while the game is being played?  
And also displaying who the winner is?

1 Add a **Label** somewhere in your user interface in the Designer. Name it appropriately.

2 Use an **if-then** block to test who is leading. Depending on who is leading, set your Label appropriately.

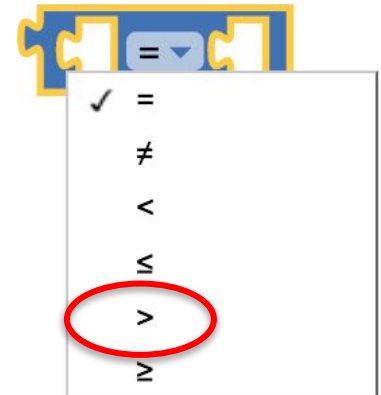


You need to cover 3 possible conditions:

- Player 1 leads
- Player 2 leads
- it's a tie.



ies.



“ Player 1 is in the lead! ”

“ Player 2 is in the lead! ”

“ It's a tie! ”

3 You can use the same **if-else-if-else** block for the end of the game. You can Duplicate the entire block and just change the text blocks slightly. ----->

“ Player 1 wins! ”

“ Player 2 wins! ”

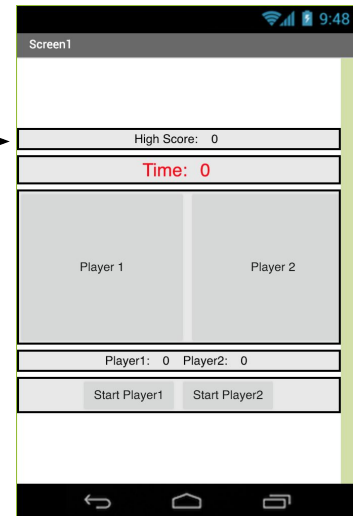
“ It's a tie! ”

4 Don't forget to test out your new features! Listen for your sounds and look for the updates in your new Label.

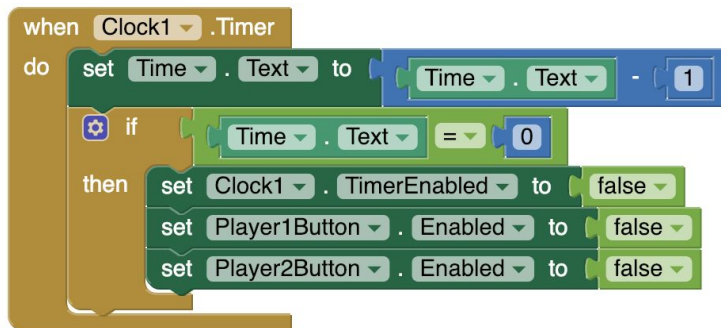
## DISPLAY HIGH SCORE

Since you're competing against friends over devices, it would be fun to display the high score of all time.

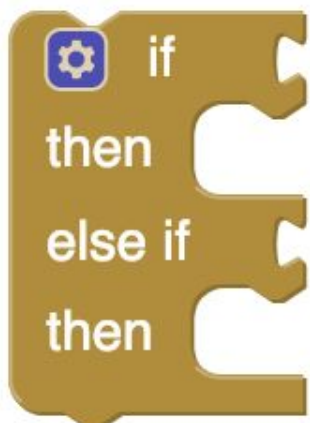
- 1 Add a **HorizontalArrangement** and two **Labels** for high score somewhere in your user interface in the Designer. Name the components appropriately.



- 2 In the **Blocks Editor**, when the game is over, check if either Player 1 or Player 2's score is greater than the high score. Note that you can store the high score in your HighScore label's *Text*. Then store the new high score in CloudDB.



HighScore . Text



set HighScore . Text to

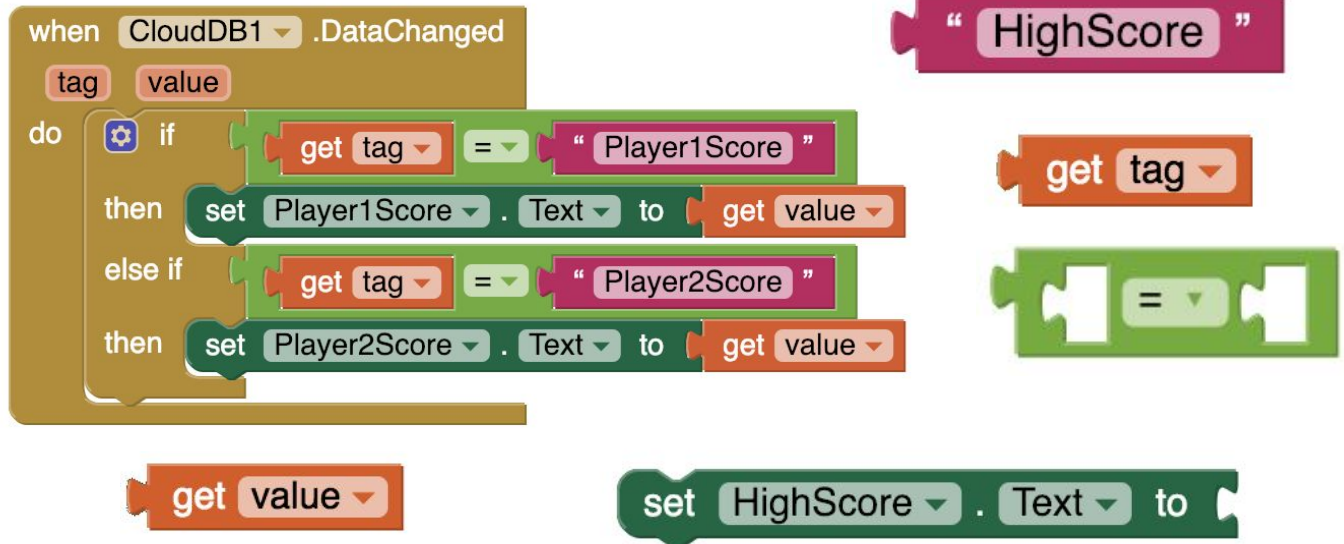
Player1Score . Text

Player2Score . Text

"HighScore"

DISPLAY HIGH SCORE (continued)

- 3 Add to the **CloudDB1.DataChanged** event to test for the new tag. Use the blocks below.



- 4 When the app starts, ask CloudDB for the current high score. Remember that when you use **CloudDB1.GetValue**, it triggers a **CloudDB1.GotValue** event when it returns the tag/value. Use these blocks.

