TWO-BUTTON GAME: CHALLENGE

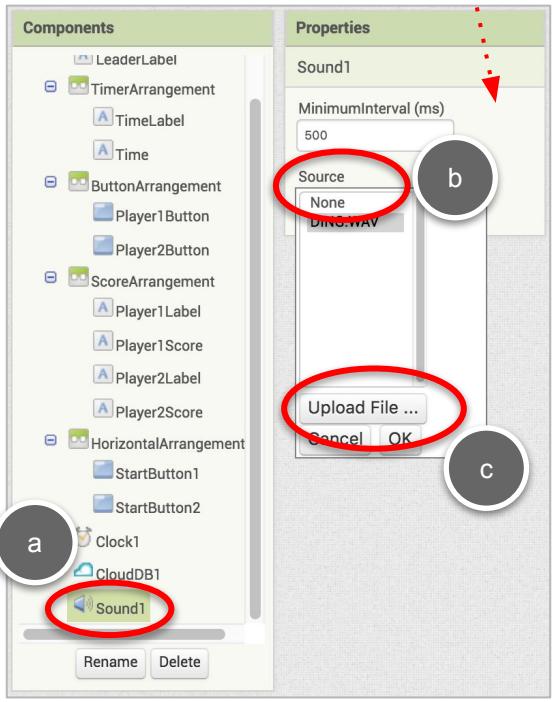
ADD SOUND

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

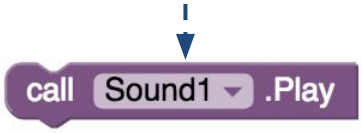
1. Add a Sound (or Player) component. Remember that it's non-visible so you won't "see" it on the Viewer.

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 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?

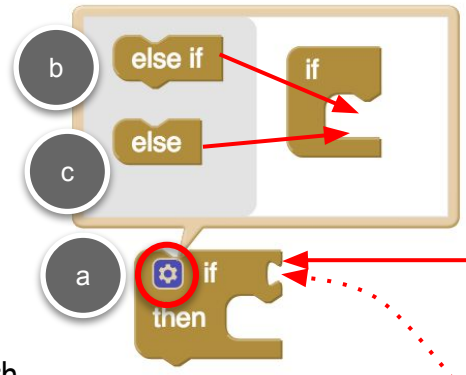
1. Add a **Label** somewhere in your user interface in the Designer. Name it appropriately.

2. Add an **if-then** block to test who is leading whenever a score is updated. Depending on who is leading, set your Label appropriately.

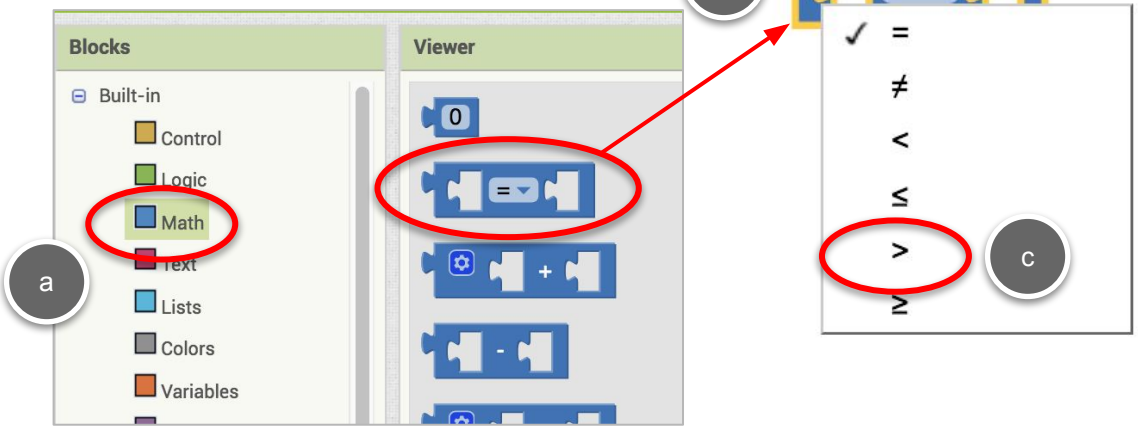
You need to cover 3 possible conditions:

- ❑ Player 1 leads
- ❑ Player 2 leads
- ❑ It's a tie.

3. Add **else if** and **else** to your **if** block.

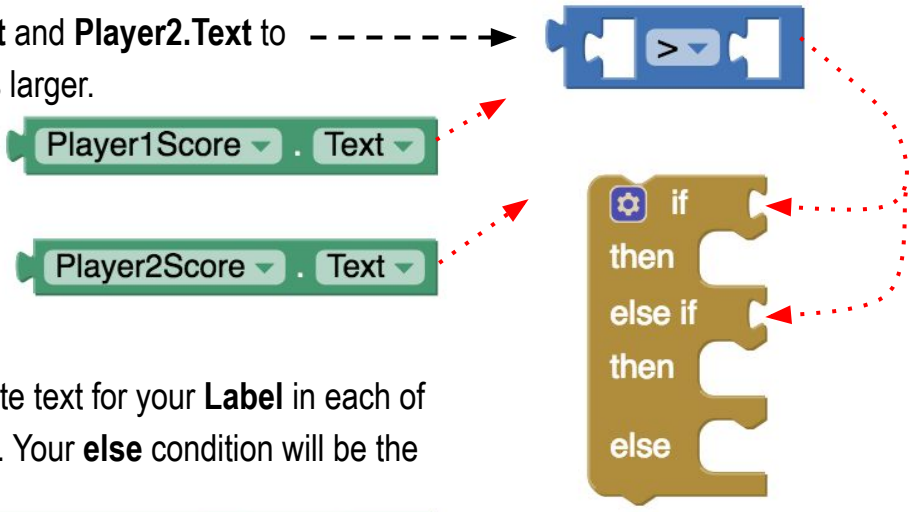


4. To test is something is greater than, drag out an equals block from the Math drawer and change it to a greater than (>) block.

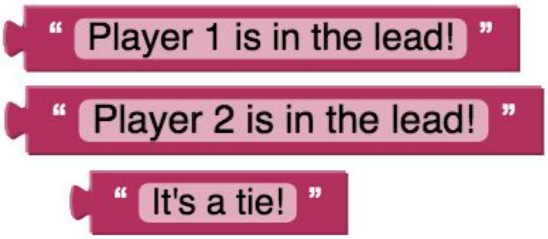


DISPLAY LEADERBOARD (continued)

5 Use **Player1.Text** and **Player2.Text** to compare which is larger.



6 Set the appropriate text for your **Label** in each of your 3 conditions. Your **else** condition will be the tie.



7 This entire **if-else-if-else** block needs to be snapped in … where?

- Whenever the score gets updated.
  Do you remember where that happens?

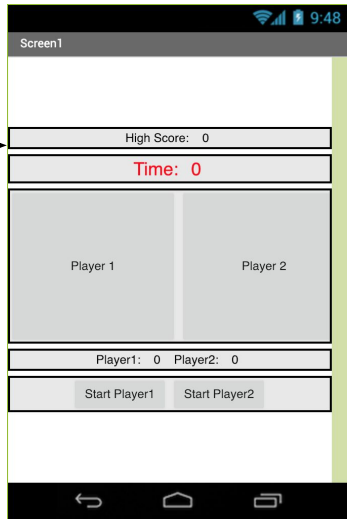
8 You can use the same **if-else-if-else** block for the end of the game. You can Duplicate the entire block and just change the text blocks slightly.

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

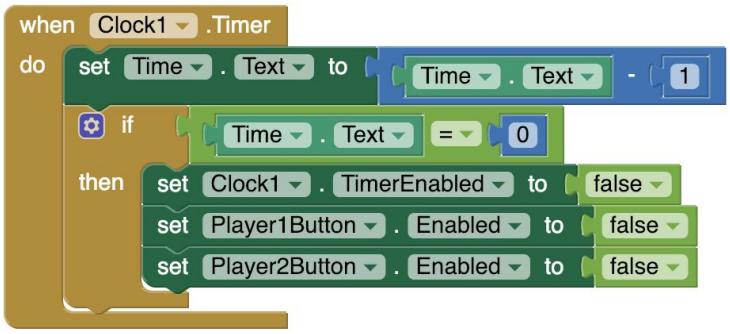
DISPLAY HIGH SCORE

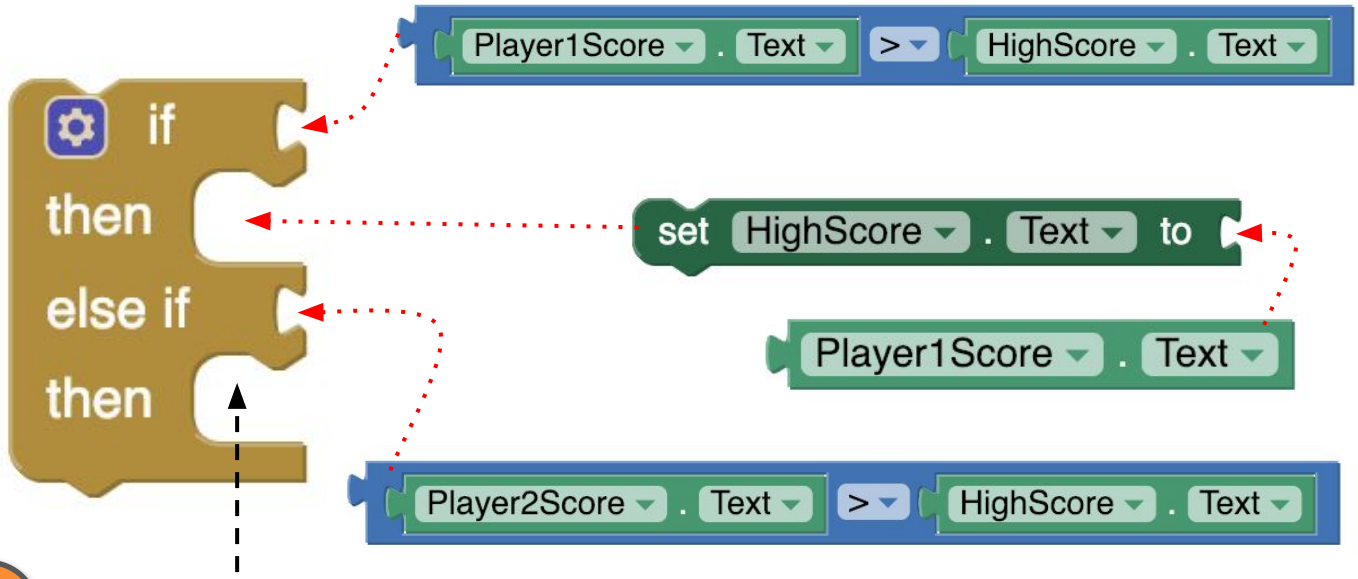
Since you're competing against friends over devices, it would be fun to display the high score of all time.

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*.

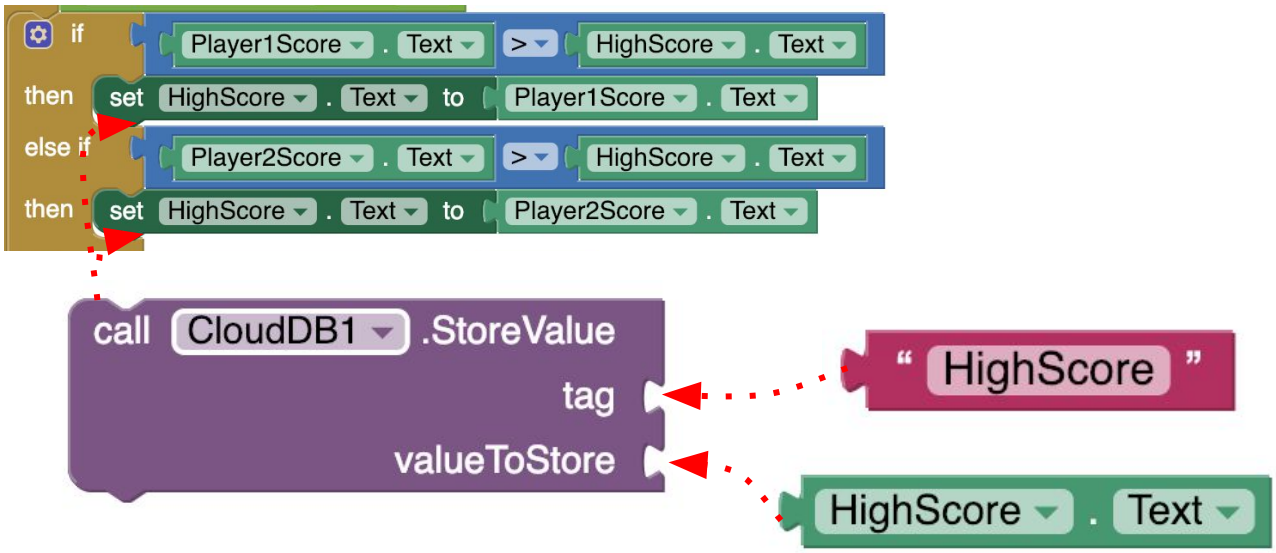




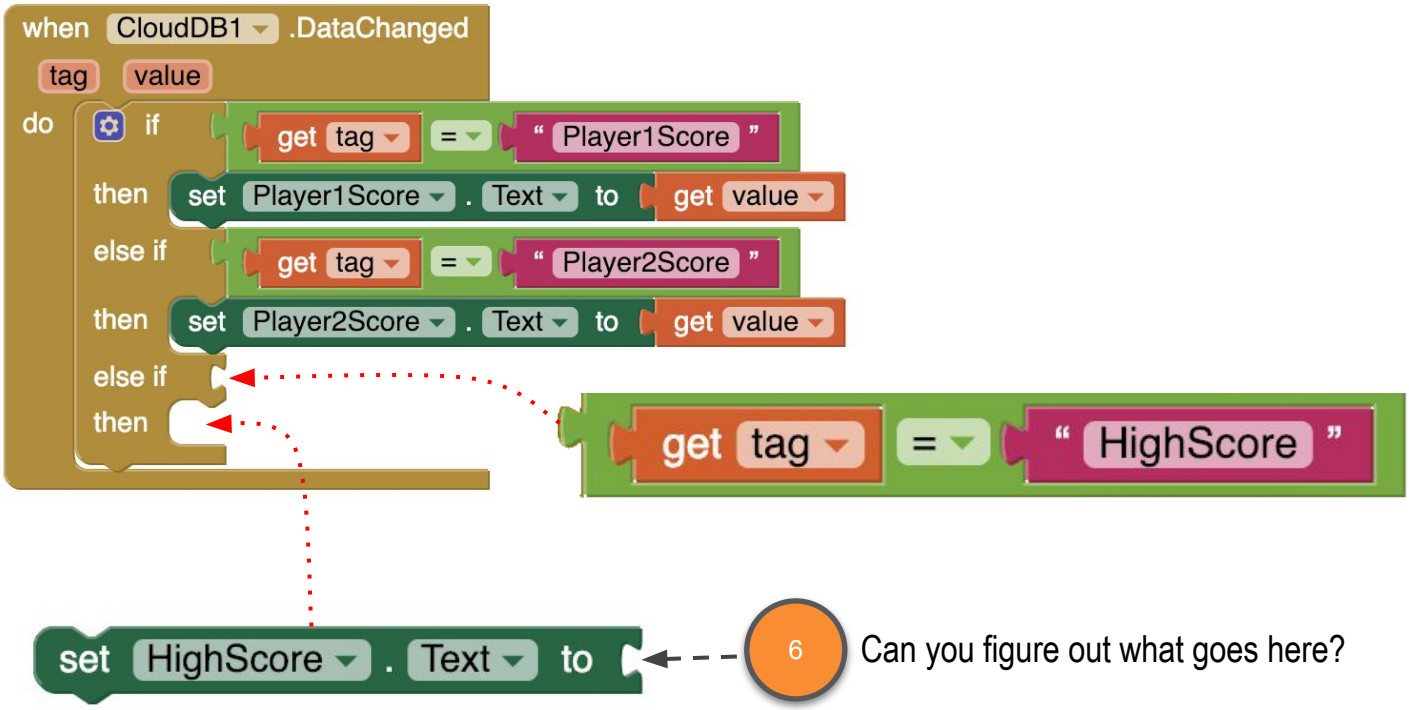
3 What goes here?

DISPLAY HIGH SCORE (continued)

4 Then store the new high score in CloudDB.



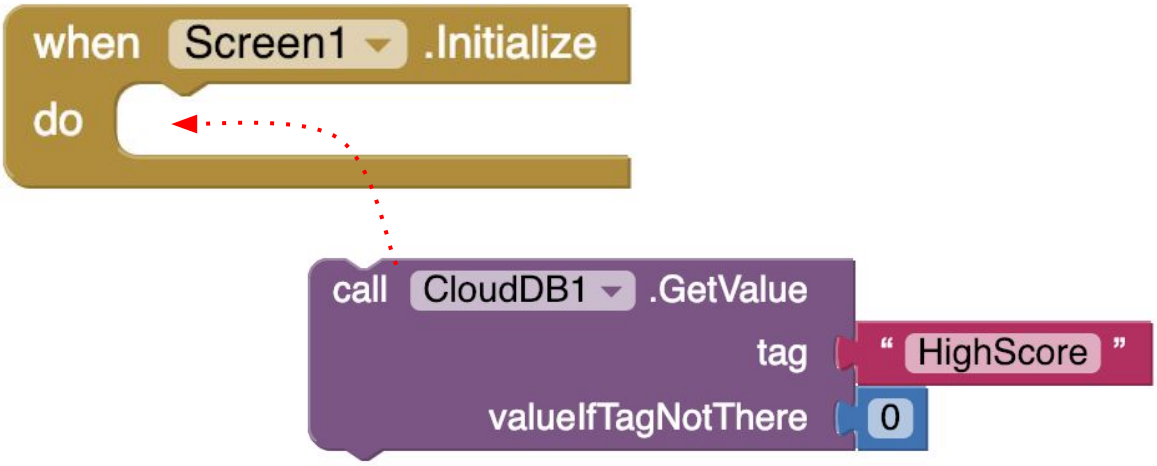
5 Add another else-if in the **CloudDB1.DataChanged** event to test for the new tag.



6 Can you figure out what goes here?

DISPLAY HIGH SCORE (continued)

7 Each time the app starts, you need to explicitly ask CloudDB for the current high score.



8 Everytime you call **CloudDB1.GetValue**, the information is returned in the **CloudDB1.GotValue** event. This works just like **DataChanged**. Test if you've got the right tag. If so, set high score to the value.