

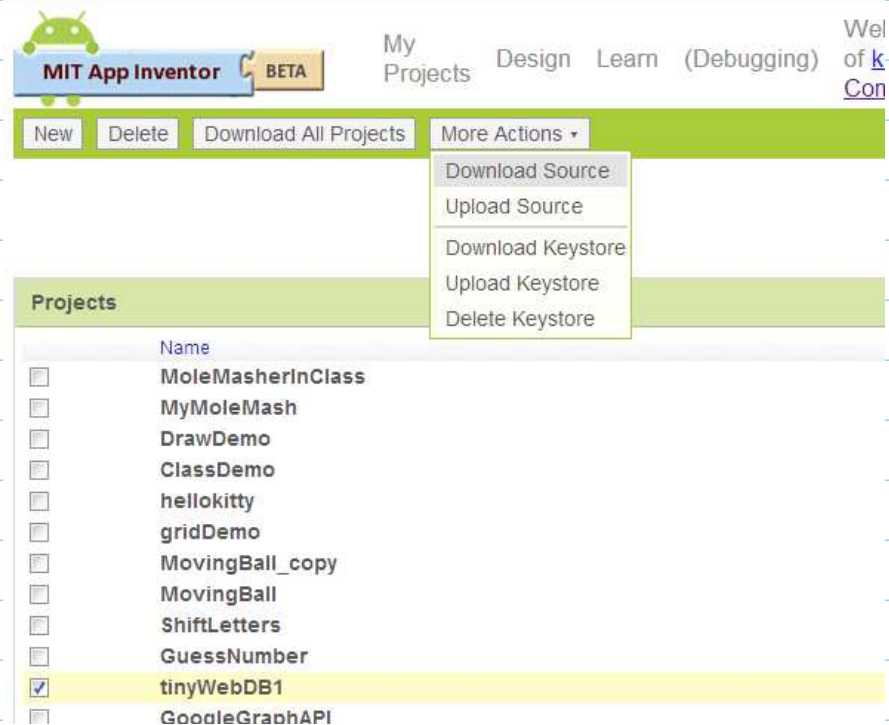
# 11-30 Class Notes CS 100

Friday, November 30, 2012  
10:00 AM

## Announcements:

Due **next** Thursday 12/6: some kind of game or contest built using AppInventor that includes storing high score using TinyWebDB.

To upload someone else's AppInventor file, they must have first downloaded it (see below) as a zip file. Use the same menu to *upload* it.



Due **next** Thursday 12/6: some kind of game or contest built using AppInventor that includes storing high score using TinyWebDB.

## Questions?

### Last Time:

The "Half-Right" example in AppInventor

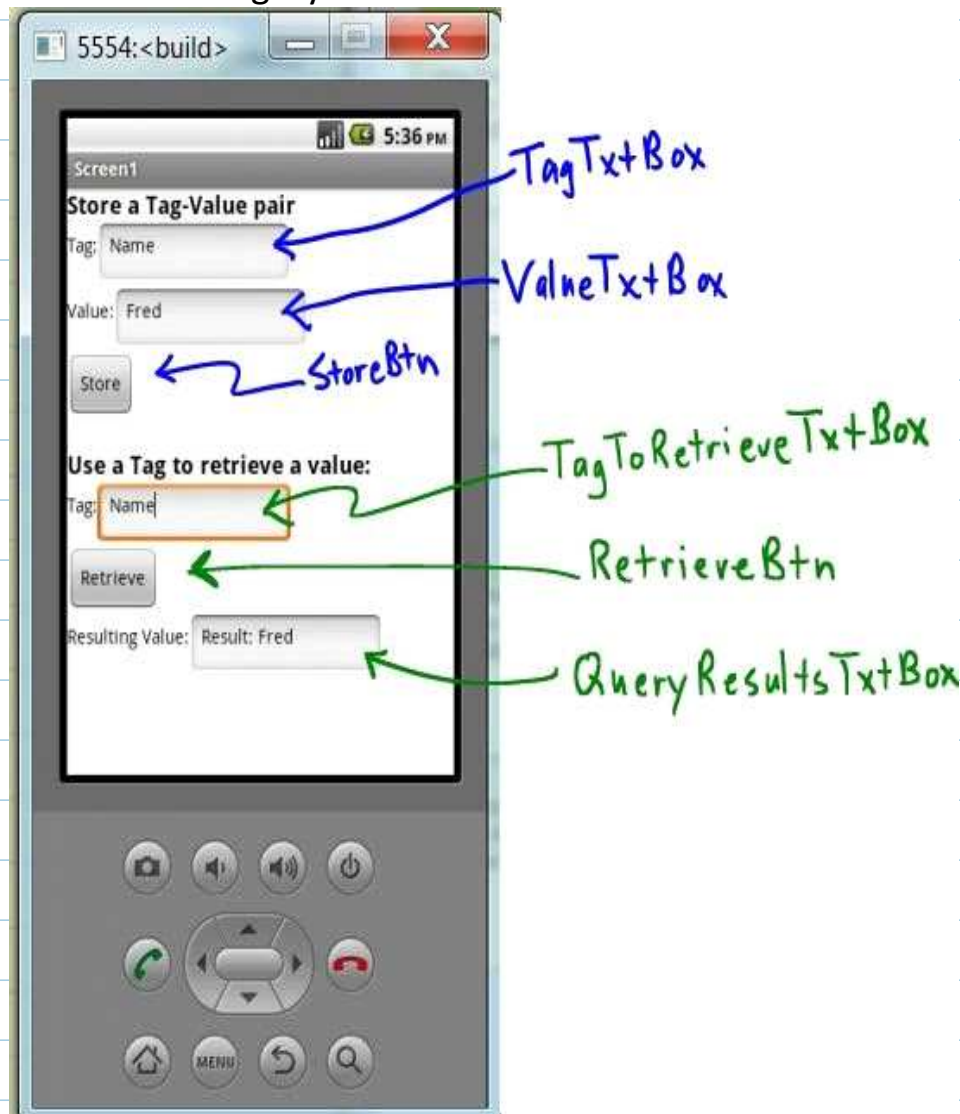
### Today:

Making our own TinyWebDB example in AppInventor

## Making our own TinyWebDB example in AppInventor

Create an interface that looks like what is shown below. Don't worry too much about formatting, just get the components on the screen so you can use them.

You will also need to select a non-visible component for the TinyWebDB, which is in the "Other" category.



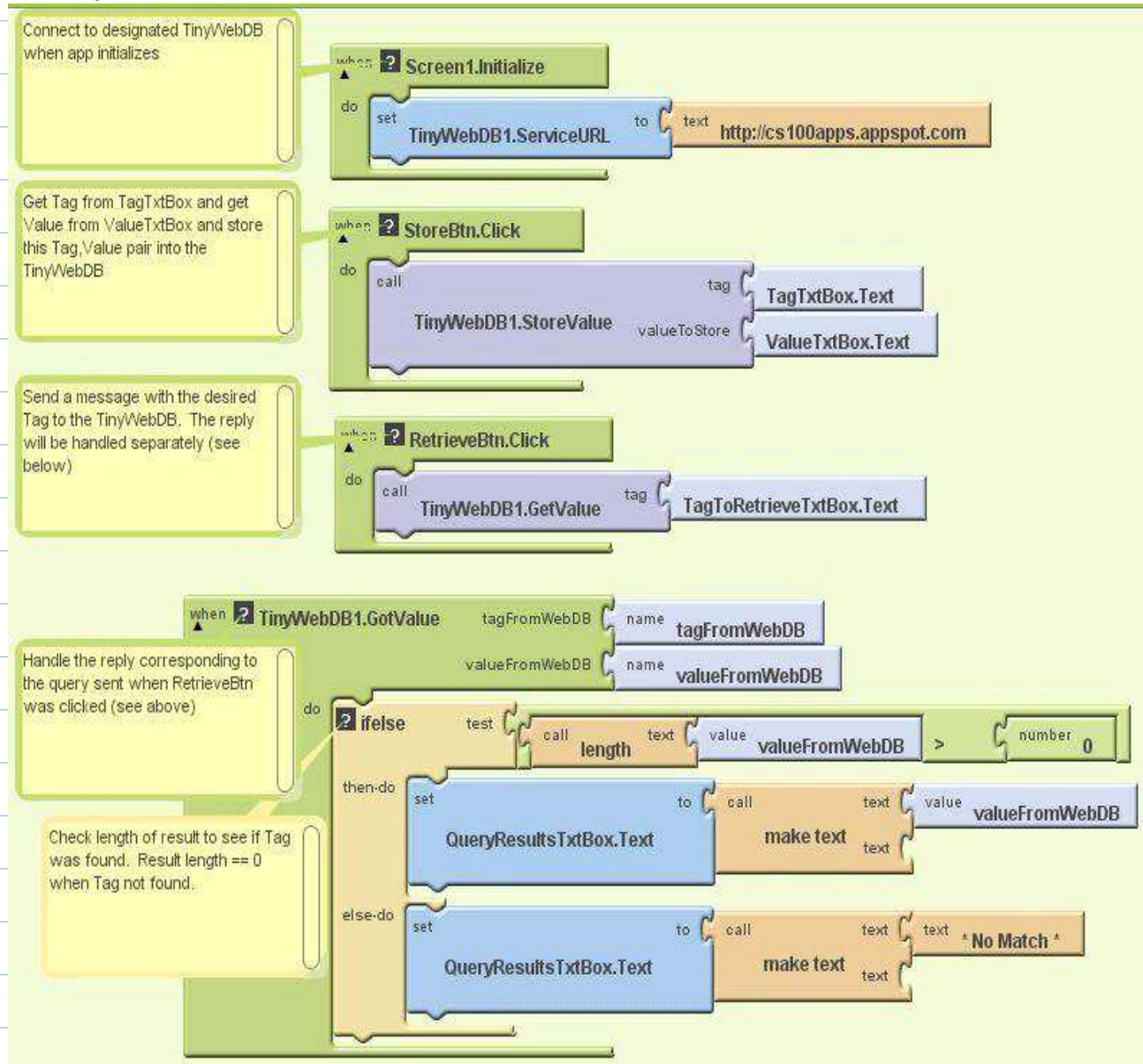
Then using the blocks editor corresponding to the above interface create the following actions:

- Screen1.Initialize should define which TinyWebDB we will be using. Set this to **cs100apps.appspot.com**
- StoreBtn.Click should call TinyWebDB1.StoreValue
- RetrieveBtn.Click should call TinyWebDB1.GetValue, which uses the text from the TagToRetrieveTxtBox. Note that this sends the request, but the reply from

the TinyWebDB is handled separately.

- d. TinyWebDB1.GotValue handles the reply. We check the length of the reply, which should be  $> 0$  if the Tag matched some existing text entry.

The blocks to do this should look like what is shown below, but you will not necessarily have all the added comments.



Once this is working, choose a partner or group with a different computer and make sure you can send messages back and forth to each other. Your agreement with each other on what Tags to use is called your *protocol*.

Questions:

- o Would this work if you were in different places and not both in the same room? How could you be sure that the person on the other end received your

message?

- How could you have your application notify you when a message has come in? For instance, how might you write an App that allows you to play Tic-Tac-Toe against someone else?
- What happens if some other group uses the same Tags that you do?

With another group now design a *protocol* so that anyone in the class could send messages to anyone else in the class. Implement your protocol. What are the issues that arise?

Further work:

Since anyone can see all the posted messages, how might you encode your messages so that even if someone else sees them they can't figure out what they mean?

### **Putting it all together**

Implement the Half-Right game, where instead of using a local variable for the sum and counter, we store and retrieve these values from TinyWebDB. This way the whole class can play the game.

