Projects

Canine quiz

Learn how to use App Inventor to make a quiz app

Step 1 Introduction

These cards are going to help you learn to build apps for Android phones and tablets by introducing you to App Inventor.

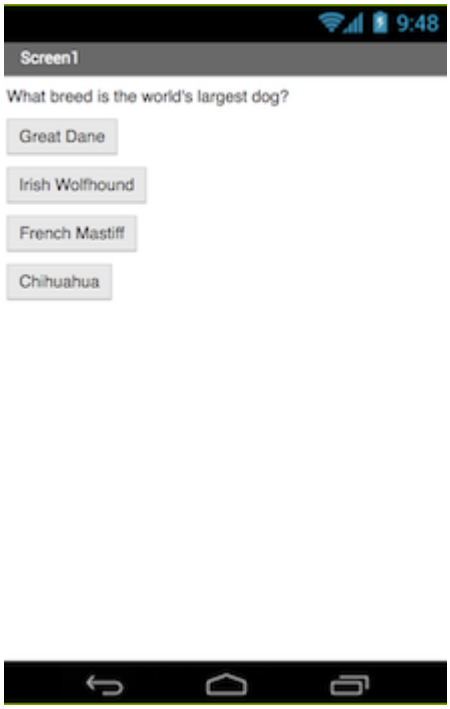
You're going to get the chance to:

- Build a quiz that keeps score as the player goes through it
- Use any questions and answers you like, maybe about your favourite band or sport, or a book, movie, or TV show you love.

I'm going to do my quiz on dogs, because I like dogs a lot. If you want to follow along with me, you're welcome to. If dogs are not your thing, then wherever I have a dog–themed question, you can come up with your own question, the right answer, and some answers that sound like they might be right! If you're going to make your own quiz, have a think about that now and maybe brainstorm some ideas with the other Ninjas in your Dojo about what kind of questions you should have.

What you will make

You'll end up with something that looks like this, and you can play around with colours and adding images to brighten it up if you like:



What you will learn

- How to use App Inventor to make Android apps
- Adding components to the screen and changing their properties
- Adding screens and writing code to move between them
- Using buttons to trigger an action
- Storing information between screens

What you will need

Hardware

- A computer capable of accessing App Inventor
- An internet connection

Optional:

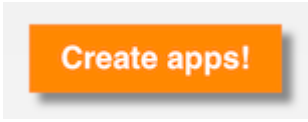
- An Android phone or tablet

Once you've learned these techniques, you can apply them to create more than just a quiz! These same coding tools can be used to make a complete interactive story, or to build a calculator or a whole bunch of other cool apps.

Step 2 Get set up

Before you can start coding your quiz app, you're going to need an account on the App Inventor website.

- Go to dojo.soy/appinv-start (http://dojo.soy/appinv-start) and then click on the Create apps! button in the top right-hand corner of the screen.



The website will ask you to sign in with a Google account. If you don't have one, you'll need to create one or use your parent's/guardian's account.

Once you have signed in with a Google account, App Inventor will show you options to either set up an Android device (a phone or tablet) or install an emulator (a program that acts like an Android device) on your computer.

Which option should I choose?

You may need a mentor at your Dojo to help you with this step!

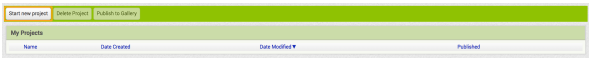
If you do not have a device available (your own or one from your Dojo), and the emulator isn't on your computer already, then you need to install it.

Otherwise, you can set up your own device or a Dojo device, or if the Dojo already has devices set up, move on to coding.

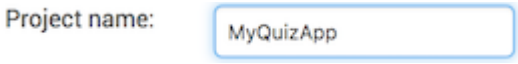
If you need to do either kind of setup, click on the right link on the App Inventor website and follow the instructions there. Once you've finished setting up, come back to these Sushi Cards.

Right! Now you're all set to go, it's time to create your first Android app.

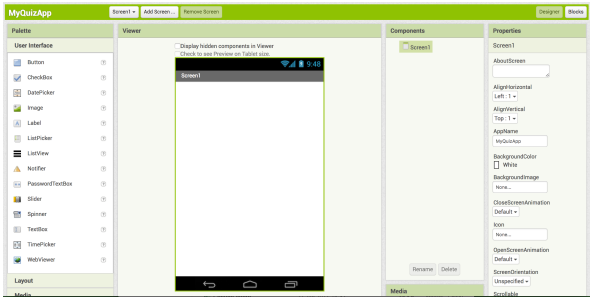
- Go back to App Inventor in your browser and click on the Start new project button in the top left-hand corner of the screen.



- Call your project `MyQuizApp` and click OK.



You'll see a screen like this one, which means you're ready to get coding!



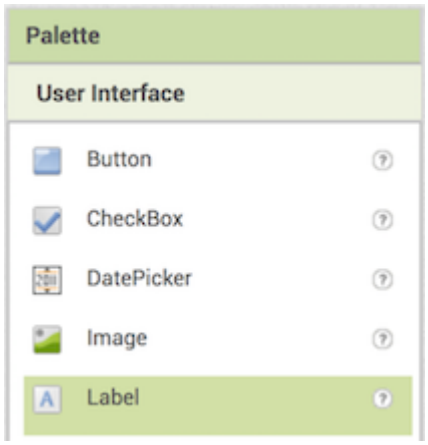
You can see that the App Inventor Designer view is broken into four key sections:

- Palette, from which you pick the components you will use to build your app
- Viewer, where you can see the app you are working on, and rearrange and select components
- Components, where you can see a list of the components in your app, and their relationships to each other
- Properties, where you can see and change the properties of the component you have selected at the moment

There are other buttons and even another view, but these four sections are what you'll be using right now.

Step 3 Create a question

- First, drag a Label component into the viewer to use for your question.



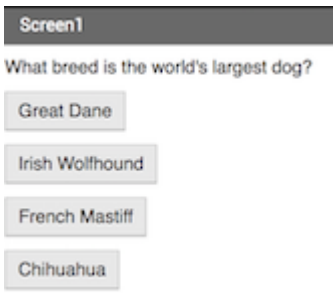
- To make this Label have the question you want to ask in it, first select it by clicking on it either in the Viewer or the Components section.
- Now, in the Properties section, look for Text (you may have to scroll down).

- Change the text in the Text box to your question. I'm going to go with "What breed is the world's largest dog?", but you can pick any question you like. Maybe ask what your favourite band's first song was, or for the score in the last World Cup final!

Of course, what good is a question if the player doesn't have a chance to answer? Now it's time for you to add a few!

- Drag four Buttons from the Palette into the Viewer, then select each of them and change the Text in the Properties section so that one of them is the right answer and the other three are wrong answers. Be as tricky or as funny as you like with the anwers!

What you end up with should look a little like this:

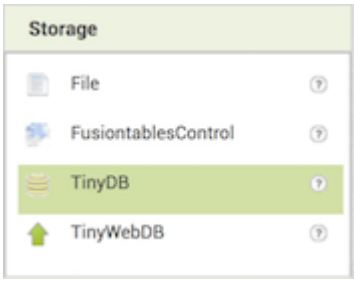


The largest dog in the world is a Great Dane, by the way! Check him out: dojo.soy/big-dog [(http://dojo.soy/big-dog)](http://dojo.soy/big-dog)

Step 4 Keep score

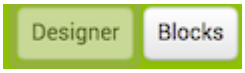
Android apps are made of Screens. You've created your first question on one Screen, and you're going to want to add more. You'll put the questions on new screens, but you need a way of keeping score between those screens. On this card, you'll be adding a TinyDB database to keep the score in, and some code to mark the right answer!

- First, in the Palette section in Storage, find the TinyDB component and drag it on to the Viewer. You won't see anything new there, but TinyDB1 should appear in the Components section.



Now it's time to start putting together the code that will power your quiz!

- In the top right-hand corner of the screen, click on the Blocks button to access the Blocks view.



Just like the previous Designer screen, this Blocks screen has sections:
Blocks, where you pick code blocks
Viewer, where you drag your code blocks to assemble them

There are lots of kinds of code blocks — you're just going to need a few of them for now.

- In the Blocks section, click on whichever button matches the right answer to your question. For me, it was Button1. Grab the `when Button1.Click do` block and drag it onto the viewer.

- Now click on TinyDB1 and choose the `call TinyDB1.StoreValue` block. Drag it so it's inside the previous block, and then go to the Built-in blocks and attach the pieces from Math and Text to make it look like this:

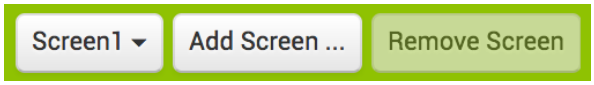
- Finally, update the Text value to `score` and the number to `1`, like this:

What you've done is to store the value `1` (as in one point for a right answer) under the label `score` in the TinyDB database. You can pull it back out and change it on later screens. This way, you can keep score throughout the app, no matter how many questions you add!

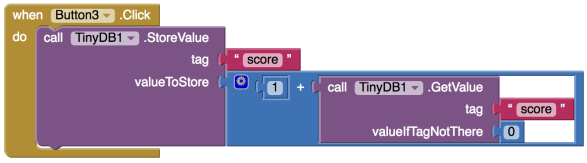
Step 5 More questions

Now you've got your first question, and you're giving the player a point for the right answer.

- To add another question, you'll need to switch back to Designer view and click on the Add Screen… button in the top menu bar.



- Name your new Screen whatever you like. I decided to be boring and leave it as Screen2 for now!
- Your new Screen will be blank. Add a Label with your next question and four Buttons with answers, just like on the last Screen. Drag a TinyDB on too, so you can get that score value you stored! I've asked what animals dogs are most closely related to (it's wolves!), but you pick whatever works for the quiz you're writing.
- Add some code in the Blocks view that looks like the following. Remember to use the right Button for your correct answer (mine was on Button3):



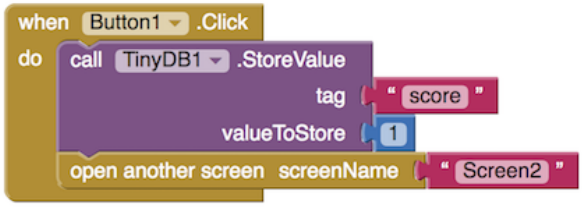
Why is the code different this time?

The code to change the score has to be a little cleverer here, since you've first got to get the value of `score` before you can add `1` to it and store it.

And since there would only be a value there if the player got the right answer on the last Screen, you need to set a default value that can be used if the player got the answer wrong.

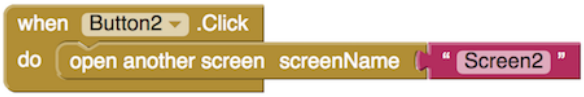
Great! But how do players get from Screen1 to Screen2? You need to go back to Screen1 and give them a way!

- To switch screens, click on the Screen2 button and pick Screen1 from the drop-down menu.
- Now, from the Built-in blocks, take the `open another screen screenName` Control block and a Text block, and add them below the score code, like this (if you've changed your screen name, you'll need to use that where I've used 'Screen2'):



Of course, this code only works if Button1 is clicked.

- You need to add a simpler block for all the other buttons (the wrong answers), like this:



By creating more screens, and adding these same kinds of blocks that point to the next screen each time, you can create an endless number of questions, and keep score throughout!

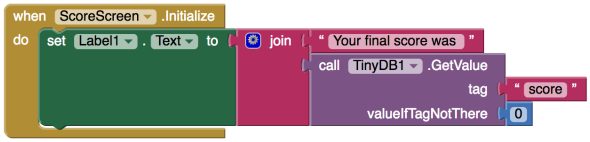
- Go make one or two more question screens following the steps on this card.

Step 6 Finish your app

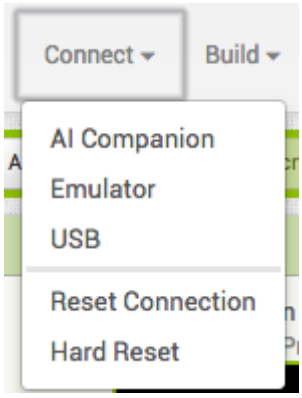
Now that you're finished adding questions and connecting screens, you need a way to tell the player how they did!

- Create one more screen called ScoreScreen with nothing but a Label on it. You also need to drag the TinyDB on so you can access the score.

- Now switch to the Blocks view and use blocks from the ScoreScreen, Label1, Text, TinyDB1, and Math sections to put this together:

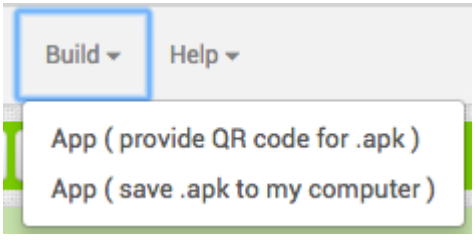


- You should now be able to go to the Connect menu and choose the emulator or other connected device that you set up for testing your apps (you might need a mentor's help here!) and see how your code works right now.



How can I install my app on a phone or tablet?

- To build a working app that you can install on your Android device, use one of the options from the Build menu: you can either download the app installer directly, or get a QR code that you can scan to download the app to your device.



That's it! You've got a quiz that will keep score across all its pages, no matter how many you add, and will tell the player how they did at the end. Nice work! Check out the next card for a few ideas on what else you can do with what you know now!

Step 7 What next?

Now you know how to build this quiz, what else can you do with this code? On this card, I'm going to give you a few ideas, but you can always come up with something cooler.

- What about an interactive story? You can use the same blocks you use to move from one screen to the next and move around screens based on users' decisions. You can create a story where, for example, you can decide to turn left or right at a fork in the road. If you map your idea out on paper first, you can create a complex branching story with loads of different characters, secrets, and endings!
- You could build an app that gives information on different screens connected with buttons — it can be about anything you're interested in! You can even try adding an Image component to the Screens to upload pictures too. In this way, I built another dog–themed app, with pictures and information about different breeds!
